POLITECNICO DI TORINO

Master's Degree in Computer Engineering

Master's Degree Thesis

Automating the Testing of Battery Modules for Electric Vehicles

Supervisors

Prof. Claudio PASSERONE

Ing. Fabrizio IMPINNA

Candidate

Giulia GOLZIO

October 2025

Summary

The aim of this thesis is to streamline and accelerate a critical yet repetitive laboratory task: testing the correct functionality of battery modules for electric vehicles by simulating realistic operating conditions, including temperature, humidity, acceleration, braking, and steady-speed phases.

To achieve this, a custom electronic board and a dedicated PC application are developed to manage all machinery and sensors involved in the test from a centralized control interface. The system enables synchronized communication with devices such as chillers, climatic chambers, cyclers, and battery management systems via protocols including CAN and Modbus TCP.

Although the physical setup remains complex and the time required for its preparation cannot be significantly reduced, the implemented solution enables a more efficient execution of tests, both in terms of time and operational flexibility. Operators can run multiple test sequences in succession, monitor key parameters in real time, and interrupt the procedure either gradually or immediately in case of anomalies. At the end of each test, a detailed report is generated, summarizing the conditions and outcomes.

The thesis demonstrates that the proposed system effectively simplifies and accelerates the validation process of newly assembled battery modules and packs. Future improvements may include enhancing the graphical interface for better usability and extending the system's capabilities to support additional types of DUTs, such as inverters, as well as integrating further laboratory instrumentation.

Acknowledgements

Prima di inoltrarci in questo progetto di tesi, vorrei porgere i miei ringraziamenti a chi mi ha accompagnato in questo cammino.

Innanzitutto vorrei ringraziare i miei relatori per il loro costante sostegno. In particolare ringrazio il professore Claudio Passerone per la sua genuina cordialità e per gli infiniti consigli che mi hanno aiutato a scrivere questa tesi. Ringrazio l'ingegner Fabrizio Impinna e tutti i suoi colleghi per avermi accolta nel loro laboratorio e per avermi spinto a trovare da sola soluzioni alternative, mai lasciandomi senza sostegno.

Un caro ringraziamento va ai miei genitori, alle mie sorelle e a mio fratello. In questi anni non hanno mai smesso di sostenermi e credere in me, e da sempre mi insegnano cosa voglia dire famiglia. Al mio ragazzo e ormai compagno di vita, Michael, ringrazio tutto. Dall'immenso amore col quale mi supporta (e sopporta) ogni giorno, fino ai piccoli gesti che hanno reso questi anni i più belli della mia vita.

Sicuramente non sarei mai riuscita a concludere questo percorso di magistrale senza i miei compagni di corso, e ora tra i miei più cari amici. Ringrazio Giulia e Giacomo per il loro supporto nei momenti più difficili, e assieme a loro Davide, Angela e Federico per le fantastiche serate giochi passate assieme. Tra i miei compagni conosciuti nei primi anni di triennale ringrazio in particolare Filippo: abbiamo affrontato assieme lezioni ed esami dell'università, gioie e fatiche della vita. Ringrazio Arianna per il suo sostegno, la sua schiettezza e soprattutto la sua amicizia. Ringrazio Sara, Giorgio, Ioana, Luciano, Giada e Daniele per le risate e le serate insieme: grazie di avermi accolta.

Un caro ringraziamento va ai miei amici Mauro, Rebecca, Manuel, Riccardo e Francesca. In questi anni abbiamo condiviso tanto e insieme siamo cresciuti e cambiati. Ringrazio le mie amiche ormai storiche Sara, Bea e Serena: quando ci incontriamo sembra sempre sia passato solo un giorno dall'ultima volta che ci siamo viste.

Infine i miei ringraziamenti vanno a quelle persone che in questi anni, per un motivo o per l'altro, sono entrate ed uscite dalla mia vita. Anche con loro ho riso, pianto e sono cresciuta.

Buona lettura.

Table of Contents

List of Tables

List of Figures

Acronyms

CAN
: control area network

DUT
: device under test

GUI
: graphical user interface

HMI
: human machine interface

HVPS
: high-voltage power supply

LVPS
: low-voltage power supply

PSS
: power supply system

RTU
: remote terminal unit

TCP
: transmission control protocol

UDP
: user datagram protocol

UI

user interface

"Lo stupore è la chiave"
Barbascura X, Il genio non esiste

Chapter 1

Introduction

The rapid diffusion of various types of electric vehicles, such as cars, scooters, bikes, and kick scooters, is driving significant growth in the field of batteries. The increasing demand for low-environmental-impact vehicles, whether fully electric or hybrid, is reshaping the automotive sector. As a result, companies engaged in the design and production of batteries are required to develop efficient and flexible battery packs and modules. Furthermore, accelerating the processes of design, test, and production has become essential in order to remain competitive in the global market. In this context, the present research focuses on the development of a tool designed to conduct battery testing in a more flexible, practical, and efficient manner.

A crucial aspect of battery testing lies in replicating conditions as close as possible to those experienced when the battery is integrated into a vehicle. To achieve this, a wide range of equipment—including power supplies, safety chambers, cooling systems, and climatic chambers—must be employed. Traditionally, operators are required to configure and operate each of these systems individually, resulting in considerable effort and time consumption. The objective of this project is therefore to design a control system capable of managing all equipment through a single interface. This system not only centralizes control but also enables the execution of different categories of battery tests. The framework is inherently flexible, allowing multiple devices to be tested using the same interface. The implementation relies on a dedicated software application and an ad hoc electronic control board developed by the hosting company.

The research methodology is structured into three main phases. First, the individual components of the system and their specific roles within the testing procedures are thoroughly analyzed, in order to clearly define their functions and constraints. Second, the communication protocols required to ensure interoperability between the machines, the application, and the control board are investigated. This step involves both hardware development—such as the design of custom communication cables—and software implementation, including the integration and adaptation of the necessary communication protocols. Finally, each of the defined tasks is validated through targeted testing. The methodology follows a modular approach, whereby the three phases are applied independently to each machine. Upon completion, an integrated validation test is conducted on a battery module designed and assembled by the hosting company.

1.1 Research Motivation

The objective of this research is to enhance the efficiency and quality of laboratory activities related to battery module testing. The work introduces three main advancements.

Firstly, the proposed system enables operators to significantly reduce the number of manual operations typically performed through the machines' human–machine interfaces (HMIs). By means of a unified PC-based interface, operators can rapidly configure machine parameters and operating values. In addition, the automated and sequential transmission of data eliminates the need to generate and manually transfer configuration files to each individual machine via USB devices. Furthermore, the simultaneous and coordinated start-up of all laboratory equipment is ensured, as the process is software-controlled rather than manually initiated.

Secondly, the system provides greater flexibility. Prior to initiating a test, operators can select the equipment to be employed, define its operating parameters, and upload a test profile containing the complete set of values to be distributed across the selected machines.

Finally, this research enables new categories of experiments that were previously not possible in the laboratory. For instance, it allows specific testing procedures to be performed on battery modules, simulating the function of the battery in a vehicle on the road. This will be discussed in detail later in the thesis.

Overall, the motivations underlying this research are multifaceted, with the overarching objective of improving laboratory processes and simplifying the work of operators.

1.2 Related Works

This section provides a brief overview of previous studies related to the automation of battery module testing.

As early as 2003, Phillip E. Pascoe and Adnan H. Anbuky published a paper describing a typical battery test cycle, involving sequential charging and discharging phases. Their work also introduced techniques that apply elevated temperatures and operational stressors to accelerate battery ageing, a process known as accelerated thermal ageing. In their article, titled *Automated Battery Test System* [1], the authors identify key battery parameters to be monitored and controlled through a specialized cyclic testing protocol. The proposed solution is tailored for valve-regulated lead-acid batteries, aiming to characterize their behavior under stress. A notable feature of their automation framework is its versatility: the system is designed to accommodate batteries of varying configurations, support multiple data acquisition methods, and execute tests in diverse operational modes.

A decade later, in 2013, Yi Peng Wan and Wei Yao presented their work titled *Study on Mechanical Automation with Improvement of Battery Test System for Electric Vehicle* [2]. This study outlines the development of a testing platform for electric vehicle battery packs. The authors employ a PC-based software simulator to emulate real-world battery conditions or to perform controlled charge/discharge cycles. Test profiles are communicated via CAN bus and RS485 interfaces, and configuration data can be uploaded prior to testing through an Excel-based interface, enhancing usability and flexibility.

More recently, in 2022, Silvia Colnago, Marco Faifer, Emil Petkovski, and Luigi Piegari presented their research at the IEEE International Conference, focusing on battery health estimation and remaining useful life prediction. Their primary objective was to develop a cost-effective and automated system for battery ageing. The experimental setup involved eight lithium-ion cells tested under varying states of charge and voltage thresholds. However, the system was designed to be scalable and adaptable to batteries with different cell counts. Their contribution, titled *Automated Test Equipment for Battery Characterization* [3], also highlights the capability for users to program and execute multiple test cycles in succession, thereby enhancing the platform's flexibility and applicability.

In this same year (2025), Shahab Khalghani published his thesis on Theseus, an online repository providing access to academic works from Finnish Universities of Applied Sciences. In his study, titled *Streamlining Electrical Battery Testing Through Automation: a method to automate the testing of automotive batteries employing LabVIEW, TestStand, and Arduino* [4], Khalghani presents a fully automated test bench specifically designed for selected electric vehicle battery packs. The system integrates multiple software platforms to eliminate the need for manual intervention during the testing process. The operator's role is limited to verifying electrical connections and identifying the battery pack under test via its serial number; all subsequent operations—including data acquisition and report generation—are handled autonomously.

The testing procedure encompasses a comprehensive set of measurements and diagnostics: it monitors cell temperature and voltage, ambient temperature and humidity, and performs equipotential, insulation, and dielectric tests. This approach not only enhances repeatability and reliability but also significantly reduces human error and operational overhead, aligning with the broader trend toward intelligent and autonomous battery testing solutions.

1.3 Thesis Outline

The thesis is developed as follows:

- Chapter 2 provides an overview of the system used, focusing on the various machines and instruments available.
- Chapter 3 illustrates the architecture designed for the thesis, both at hardware and software level.
- Chapter 4 explains how a test is performed using the architecture described in the previous chapter, and outlines the test executed.
- Chapter 5 analyzes the results obtained, evaluates the thesis work carried out, and highlights both strengths and weaknesses.
- Chapter 6 draws the conclusions of the thesis and proposes possible future developments to extend or improve it.

Chapter 2

System and Components Overview

The system's high-level organization is depicted in this chapter, and each element is given detailed attention.

2.1 System Organization

The system architecture is structured around two main components: the Power Supply System (PSS) board and the computer-based application. These two elements maintain continuous communication before, during, and after the execution of a test. Commands, data streams, verification signals, and warnings are constantly exchanged through this channel. Control and management responsibilities are distributed between the PSS board and the application: components and information that require real-time handling on the order of milliseconds are managed directly by the PSS board, whereas elements that only require responsiveness on the order of seconds are supervised by the application. Initially, the system architecture was designed to be fully managed by the PSS board, with the application interfacing exclusively with the board. However, it was later observed that certain devices featured incompatible communication ports relative to those available on the board. This prompted a design decision: either to expand the board's interface capabilities by adding the required ports, or to reroute the communication of these specific devices through an alternative channel. Given that the devices in question—the chiller and the climatic chamber—exhibit slow state transitions, typically on the scale of tens of seconds, the latter approach was deemed more efficient. Consequently, these components were directly interfaced with the application, bypassing the board, without compromising system performance or reliability.

The secondary components within the system architecture are designed to interface exclusively with one—and only one—of the two primary components. These secondary units are not interconnected, do not exchange data among themselves, and operate independently, unaware of the existence of other components. Their operational scope is confined to configuring received parameters, executing assigned commands, and transmitting feedback derived from internal data acquisition processes. Allowing a secondary component to communicate simultaneously with both the PSS board and the application would introduce redundancy and increase the risk of communication bottlenecks. Therefore, a streamlined and non-redundant configuration was adopted to ensure clarity, efficiency, and robustness in the system's communication flow.

2.1.1 System Diagram

Figure 2.1 presents a comprehensive schematic of the system architecture. The operator's workstation hosts a Qt-based application, which interfaces with both the chiller unit and the electronic board. The control board, in turn, communicates with the cyclator, which may consist of either a low-voltage or high-voltage power supply. The configuration depicted in the figure illustrates a scenario in which the DUT is positioned within the test cage. The locking mechanism of the cage is monitored and verified by the PSS board. In alternative setups where a climatic chamber replaces the test cage (not depicted in figure), the chamber establishes direct communication with the Qt application. To accommodate this network topology, an Ethernet switch is employed to manage communication among the three devices (board, application and chamber), ensuring reliable data exchange and coordinated control across the system.

The DUT is powered via the selected cyclator. Along the positive power line, a contactor—denoted in orange as `C`—is inserted, while a shunt—also marked in orange as `S`—is placed on the negative line. When the high-voltage power supply is employed, an additional contactor (indicated in green as `C`) is integrated to enhance the safety of the DUT.

A closed-loop coolant circuit connects the DUT to the chiller. Temperature sensors (labeled in orange as `T1` and `T2`) and pressure sensors (marked in orange as `P1` and `P2`) are installed at the inlet and outlet of the DUT. Additionally, a flow sensor (indicated in orange as `F`) is present within the circuit.

All sensors, contactors, and the shunt transmit their respective data directly to the PSS board, as represented by the dark orange lines in the schematic.

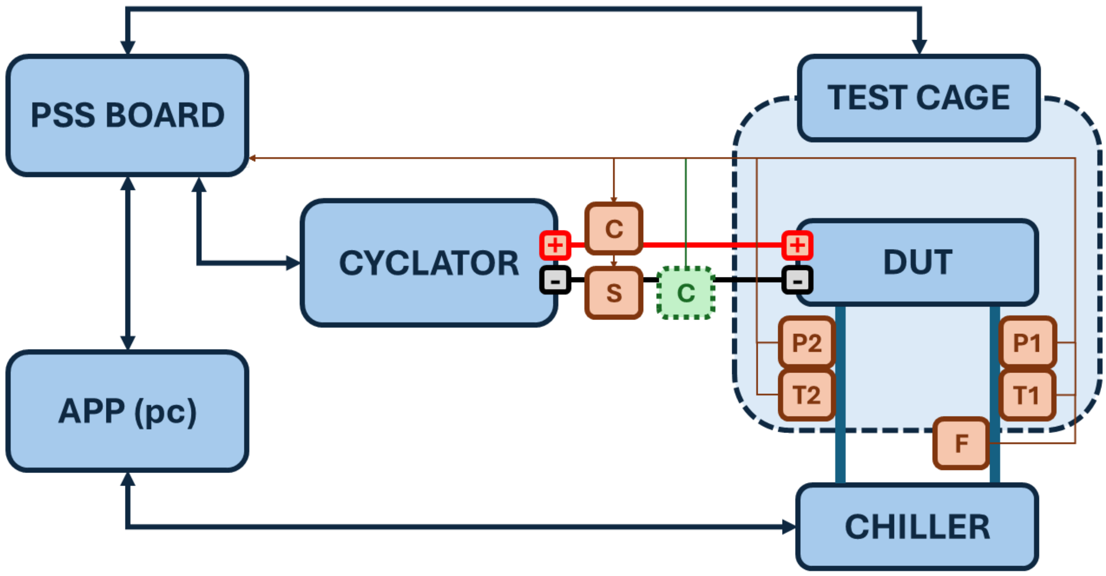

Figure 2.1: Overall system diagram

2.2 Device Under Test

The Device Under Test (DUT) represents the central element of the experimental framework. It is the component subjected to controlled stimuli in order to evaluate its functional behavior, performance characteristics, and response to specific operating conditions. The DUT's interaction with the test environment provides the data necessary for analysis and validation, making it the focal point of the entire testing procedure.

This thesis primarily focuses on testing battery modules and battery packs, which are fundamental components in electric vehicles. A battery module is composed of multiple battery cells connected in series and/or parallel to achieve higher voltage and capacity levels. It typically includes interconnecting elements, a BMS, and a protective casing. The BMS plays a critical role in monitoring cell parameters, balancing charge levels, and ensuring safe operation under various conditions.

A battery pack comprises multiple battery modules arranged in a defined configuration, supplemented by external subsystems including an advanced BMS, thermal management units, and mechanical enclosures designed to ensure safety, durability, and environmental protection. The pack-level integration introduces additional complexity in terms of electrical, thermal, and mechanical interfaces, which must be carefully managed during testing.

At the most granular level, a battery cell constitutes the basic unit of energy storage. Its intrinsic properties—such as nominal voltage, capacity, internal resistance, and cycle life—directly influence the overall performance and reliability of the battery system. The aggregation of cells into modules, and subsequently into packs, allows for scalable energy solutions tailored to specific application requirements.

Although the DUT in this project is limited to battery modules and packs, it is worth noting that other types of DUTs may be considered in similar testing environments. For instance, a pair of high-voltage inverters connected to electric motors could be alternative DUTs. Such configurations, however, fall outside the scope of this thesis and are not addressed in detail.

2.3 PSS Board

The Power Supply System (PSS) board [5] is a custom electronic board developed by the host company, designed to offer a wide range of input/output capabilities—both physical and digital—as well as communication and power distribution functionalities [6]. At its core, the board integrates a TMS570 microcontroller from Texas Instruments [7], which serves as the main processing unit. Both the PSS board and the TMS570 module are shown in Figure 2.2.

Initially, the board was conceived to supply power to various peripheral components requiring electrical input. Examples include flow sensors, pressure sensors, and shunt resistors. This original power distribution role is the origin of its name. However, the company later extended its functionality to serve as the central control unit for the entire test system.

In its expanded role, the PSS board is responsible for acquiring feedback from connected devices—including sensors, contactors, and shunts—and for transmitting parameters, commands, and control values to the relevant components in a time-synchronized manner. To support these tasks, the board is equipped with multiple communication interfaces, primarily consisting of Molex connectors. These connectors, while compact and widely adopted in industrial applications, present certain limitations in terms of ergonomics and ease of use—particularly in scenarios requiring frequent reconfiguration. As the number of crimped wires increases, the connectors tend to become progressively more difficult to detach, potentially impacting the efficiency of test setup and teardown procedures.

Communication with the board is established via the Ethernet port located on the TMS570 microcontroller. This port not only enables real-time data exchange with the desktop application but also serves as the interface for uploading firmware and configuration code to the board.

The control board requires a dedicated 12V power supply to ensure proper operation. This voltage can be provided by any power source capable of delivering a stable 12V output, as long as it meets the safety specifications for the system.

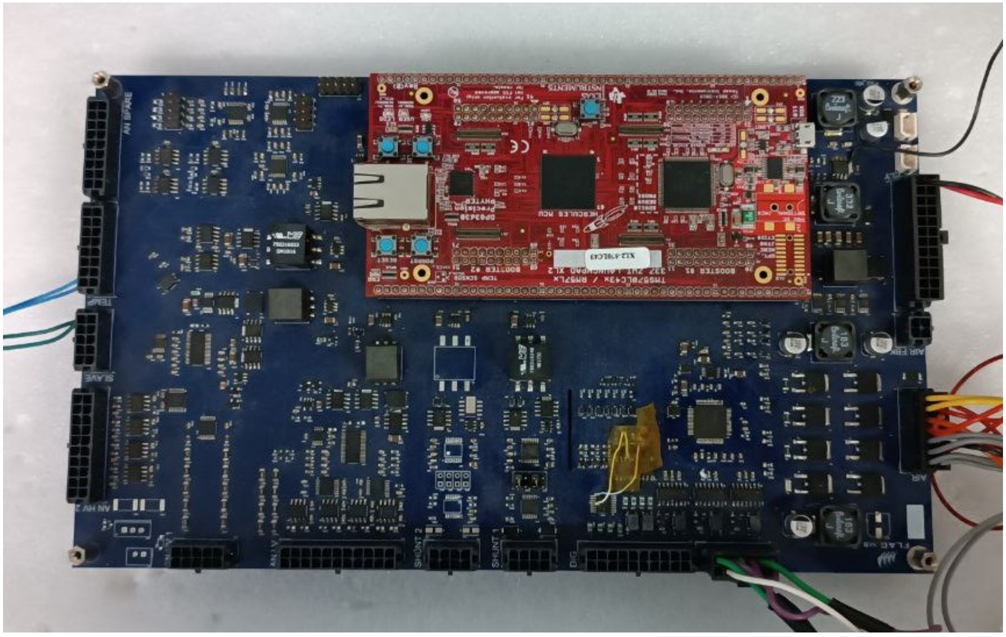

Figure 2.2: PSS board (in blue), with TMS570 onboard (in red)

2.4 Application

The software application has been specifically designed to serve as an intuitive and robust graphical interface for operators engaged in system-level testing procedures. Its primary objective is to facilitate the configuration, control, and monitoring of complex test environments, ensuring both operational safety and procedural repeatability. The application is developed using the Qt framework, a widely adopted cross-platform development toolkit that supports the creation of high-performance graphical user interfaces. Qt leverages a modular architecture based on widgets, which act as the fundamental building blocks for interface design. These widgets enable dynamic layout management, event-driven interaction, and seamless integration with underlying control logic.

Through commands from the GUI, the application acquires all necessary input data required for test initialization and execution. This includes the configuration of machine-specific parameters such as actuator limits, sensor calibration values, and operational thresholds. Additionally, the GUI allows the definition of data transmission profiles, specifying sampling rates, communication protocols, and logging formats. Control commands—such as `start`, `stop`, and `panic`—an emergency halt—are also managed through the interface, providing operators with direct access to critical system functions.

Communication with the PSS board is established via a shared and standardized protocol, which supports bidirectional data exchange. This protocol enables the transmission of control signals, real-time status updates, and diagnostic information between the application and the embedded hardware.

In addition to core control functionalities, the application incorporates logic for managing the chiller subsystem. The chiller can be operated in either pressure-regulated or flow-regulated mode, depending on the requirements of the test scenario. Target values for temperature, pressure, or flow rate are defined via the GUI, and the application orchestrates the activation and deactivation of the cooling fluid circulation loop. Internal feedback from the chiller—such as thermal readings, pump status, and fault codes—is continuously monitored and logged. This ensures that the thermal and pressure conditions remain within specified tolerances and that any deviations are promptly addressed.

A similar control strategy is applied to the climatic chamber, which supports two operational modes: temperature-only regulation and combined temperature-humidity regulation. The application enables precise configuration of environmental parameters, including setpoints, ramp rates, and dwell times. Real-time monitoring of chamber conditions is performed through integrated sensors, and the application ensures that the environmental profile adheres to the predefined test protocol. This capability is essential for conducting thermal cycling and humidity stress evaluations.

Another critical function of the application is the systematic logging of operational messages, control events, and measurement data throughout the test execution. The logging subsystem is designed to capture high-resolution time series data, annotated with metadata such as timestamp and source module. At the conclusion of the test, the collected data is compiled into a comprehensive final report. This report includes configuration summaries and a chronological log of system events. Such documentation is indispensable for post-test analysis, validation of experimental outcomes, and traceability in engineering development workflows.

2.5 Battery Management System

A Battery Management System (BMS) serves as the electronic "brain" of a battery pack or module. It is responsible for monitoring and regulating key operational parameters such as charge and discharge cycles, cell temperature, and voltage balancing across individual cells. The primary objectives of the BMS are to ensure safety, optimize performance, and extend the overall lifespan of the battery system.

The BMS continuously detects and mitigates conditions such as overcharging, deep discharging, and thermal extremes. It actively redistributes charge among cells to maintain voltage uniformity and prevent degradation due to imbalance. In addition, the BMS provides real-time data on the battery's state of charge, state of health, and other diagnostic metrics essential for system-level control and predictive maintenance.

The use of a BMS becomes essential when dealing with multi-cell battery configurations, as is the case in modular battery systems for electric vehicles and energy storage applications. Without such a supervisory system, the risk of cell failure, thermal runaway, and performance loss increases significantly.

To operate, the BMS requires a stable and programmable power source. In the thesis setup, a Rigol DP800 power supply [8] is employed to energize the BMS. This device allows precise control over voltage and current parameters, ensuring safe startup and reliable operation. The Rigol DP800 is shown in Figure 2.3.

Communication between the BMS and external systems is primarily handled via the Controller Area Network (CAN) protocol. CAN is a robust, real-time communication standard widely used in automotive and industrial applications. Messages transmitted over CAN are structured into frames that include an identifier, a data length code (DLC), and up to 8 bytes of payload data. Each identifier corresponds to a specific type of message—such as voltage readings, temperature alerts, or fault conditions—allowing receivers to filter and process only the relevant information.

In the thesis configuration, the BMS periodically broadcasts diagnostic data and status updates to the PSS board.

Figure 2.3: Rigol DP800

2.6 Test Cage

The test cage serves as a critical safety enclosure within the laboratory environment, designed to mitigate the risks associated with high-voltage and high-current experimental procedures. Structurally, it consists of a transparent protective chamber equipped with a sliding panel that functions as an access door, as can be seen in Figure 2.4. This configuration allows for the safe insertion and removal of test components, boards, power modules, contactors, and other laboratory instrumentation.

The use of the test cage becomes mandatory when the electrical parameters involved exceed safety thresholds—specifically, voltages greater than 50 V or currents above 10 mA. Under such conditions, the risk of dangerous electrical discharge increases significantly, posing a serious hazard to human operators and sensitive laboratory equipment. These discharges, which may result from component failure, wiring errors, or transient faults, can be both dangerous and potentially fatal for humans. Moreover, they may induce secondary damage to adjacent systems through electromagnetic interference or thermal effects.

To address these risks, the test cage acts as a containment barrier, effectively shielding the surrounding area from unintended discharges. Its transparent design ensures visual accessibility for monitoring purposes, while its physical integrity provides robust protection against arc flash events and other fault-induced phenomena.

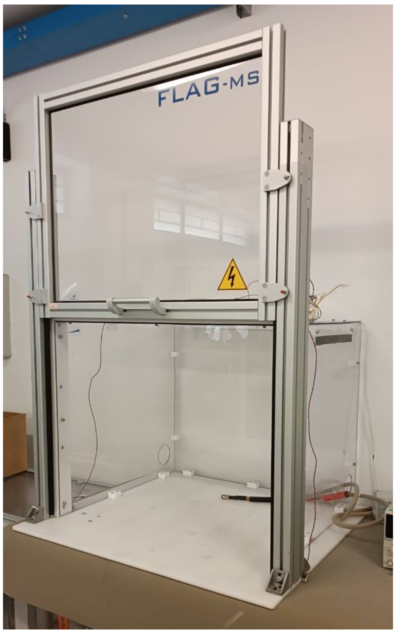

Figure 2.4: Test cage

2.7 Climatic Chamber

The host company owns a climatic chamber manufactured by MSL, model EC300CA [9], featuring a standard configuration with an internal volume of 300 dm^3 and a temperature range from −45 °C to 180 °C. The climatic chamber is a machine within which it is possible to create environments that have a defined temperature and humidity. As it is visible in Figure 2.5, a port is available to insert the element being tested; a hole on the side allows to pass the necessary connections from the DUT outwards. The HMI allows the operator to choose a temperature value in centigrade and an humidity value in percentage (relative humidity), with a cent accuracy for both. After the command `start` is given, the heat and humidity inside the climatic chamber starts to change. Reaching the requested values can take some minutes, depending on the interval between the starting and the final values. Due to this, the climatic chamber communicates with the Qt application.

The human–machine interface of the climatic chamber provides the capability to create or load predefined “recipes” via a USB device. These recipes consist of temperature and humidity profiles defined as a function of time. Once a recipe is initiated, the climatic chamber sequentially executes the programmed instructions: it reaches the specified setpoints, maintains them for the designated duration, and subsequently proceeds to the next programmed values. This process continues iteratively until the recipe is completed. Recipes can either be created directly through the HMI or prepared externally as files and then uploaded to the system via USB. The HMI is visible in Figure 2.6.

For the purposes of this research, however, it is essential that all the equipments involved in the test operate in a synchronized manner. To achieve this, it is more efficient to maintain a common external profile and transmit the required data to each machine in a timed sequence. Pre-loading complete recipes onto individual devices could introduce synchronization issues across the system and limit the ability to safely terminate a test in advance. In particular, it would hinder the capacity to drive the equipment to safe operating values for the device under test in the event of an interruption. For these reasons, the internal recipe management functionality of the climatic chamber is not employed in this work.

2.8 Chiller

The host company owns a chiller manufactured by LNEYA, model KRY552T [10]. The chiller unit allows for the implementation of a closed-loop liquid-based thermal management system, designed to regulate the temperature of components under test—most notably battery modules, requiring cooling—during experimental procedures. The system operates by circulating a thermally conditioned fluid

Figure 2.5: Climatic chamber

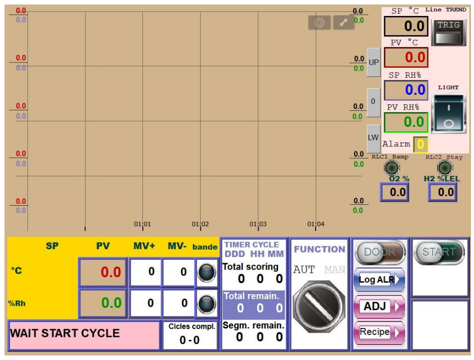

Figure 2.6: Climatic chamber's HMI

through a network of hydraulic pipes. Initially, the fluid is stored in an internal tank, where it is cooled to a predefined setpoint. Once the desired temperature is reached, the fluid is pumped outward through the circuit.

The chilled fluid travels through insulated hydraulic lines until it reaches the battery module, where it flows internally across heat-exchange surfaces. During this phase, thermal energy is absorbed from the module, thereby reducing its operating temperature. After completing the cooling cycle, the now-heated fluid exits the module and returns to the chiller via a separate return line. Upon re-entry, the heated fluid is directed back into the tank, where it undergoes another cooling cycle, thus completing the loop. This continuous circulation ensures stable thermal conditions and prevents overheating during high-power or long-duration tests. The Figure 2.7 shows the schematic of the chiller.

It is important to note that the chiller is not limited to cooling operations. The machinery is capable of applying elevated temperatures to the circulating fluid, effectively functioning as a liquid heating system when required. However, the scope of this thesis is restricted to its cooling capabilities, and heating functionalities will not be addressed in detail. Furthermore, the temperature control system of the chiller can operate in two distinct modes: linear and non-linear conversion curve.

Operational parameters of the chiller can be configured via its integrated HMI, which allows the user to define target temperature values in either degrees Celsius or Fahrenheit. Additionally, the flow characteristics of the circulating fluid can be specified in terms of pressure (expressed in bar) or volumetric flow rate (expressed in liters per minute). This dual-mode control—pressure-regulated or flow-regulated—provides flexibility in adapting the thermal management strategy to different experimental setups and component geometries.

Once the system is activated, the chiller rapidly adjusts its internal control mechanisms to meet the specified temperature and flow requirements, typically achieving steady-state conditions within a few minutes. To enable automated control and remote configuration, the chiller is interfaced with the Qt-based application described in previous section. Communication between the two systems is bidirectional: the application transmits configuration commands, including setpoints and control mode selection, while the chiller responds with real-time feedback containing both actual and requested values. This integration ensures precise thermal regulation and facilitates seamless coordination with other subsystems involved in the test procedure.

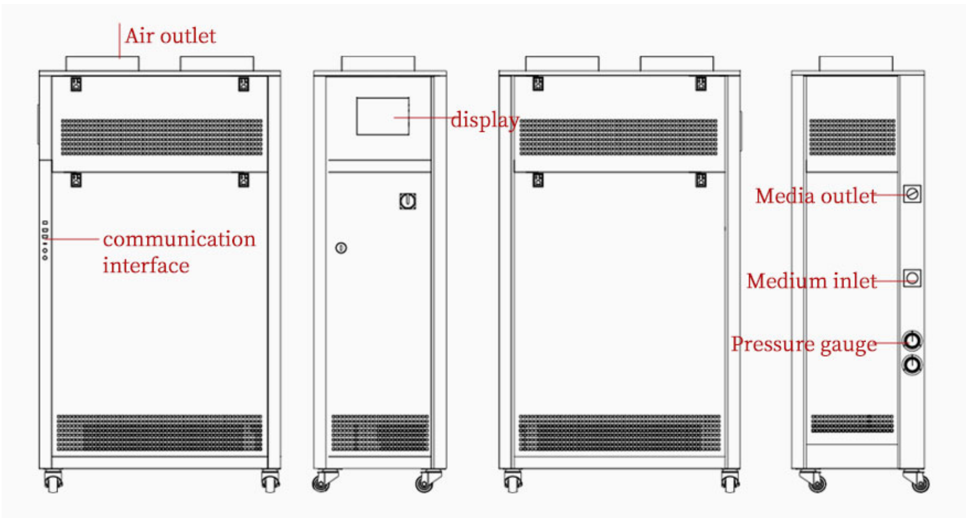

Figure 2.7: Chiller schematic

2.9 Cyclators

This thesis employs two cyclators belonging to the EA-PSB9000 series, manufactured by EA Elektro-Automatik [11]. These devices are programmable bidirectional DC power supplies that integrate two functionalities within a single unit: a power source and an electronic load with energy recovery capabilities. This dual-mode operation enables both the delivery and absorption of electrical energy, making them highly versatile for battery testing applications.

Each cyclator can be fully configured and operated either via the integrated front-panel touchscreen interface or remotely through digital communication protocols. For the scope of this research, the devices are controlled using the CAN interface, which ensures reliable and real-time data exchange with the system's control board. An EA-PSB9000 is depicted in Figure 2.8.

Cyclators are primarily used to simulate the charging and discharging cycles of battery modules under controlled laboratory conditions. This simulation is essential for replicating the dynamic behavior of batteries in electric vehicles, where energy is typically discharged during acceleration and recovered during braking. When operating as an electronic load or sink, the cyclator draws current from the battery, emulating energy consumption and discharge. Conversely, when functioning as a power source, it supplies current to the battery, simulating regenerative braking and recharging phases.

Two specific models are utilized in this study:

- Low-Voltage Power Supply (LVPS): model PSB9060-360, capable of delivering high currents up to 360 A at voltages up to 60 V. This unit is particularly suited for testing low-voltage, high-current battery modules.

- High-Voltage Power Supply (HVPS): model PSB9750-60, designed to operate at voltages up to 750 V with a maximum current output of 60 A. This configuration is ideal for high-voltage battery systems commonly found in modern electric vehicles architectures.

Both devices support precise control over voltage, current, and power parameters, in both sourcing and sinking modes. This bidirectional capability allows for seamless transitions between charging and discharging operations, which is critical for simulating real-world battery usage scenarios.

Rapid modulation of output parameters is essential not only for the accuracy of the test procedures but also for ensuring the safety and integrity of the battery modules under examination. To achieve this, the cyclators communicate with the PSS board at a frequency of approximately 60 Hz. This high-frequency exchange enables the cyclators to receive target values for voltage, current, and power, and to transmit real-time feedback on the actual values being delivered or absorbed. Such responsiveness is vital for maintaining operational safety and preventing damage to the battery systems during testing, expecially in case of problems.

Figure 2.8: PSB9000

2.10 Other Components

In addition to the primary components, several auxiliary elements are integrated into the system to ensure proper operation and safety.

Firstly, communication cables provide the physical connections between the main subsystems previously described, enabling reliable data and command transmission. If necessary, adapters and connectors are included with them.

Pressure, flow, and temperature sensors are employed to monitor the circulation of the cooling fluid supplied by the chiller, thereby providing additional feedback on thermal management performance. In particular, the pressure difference between the sensor positioned on the inlet pipe of the DUT and the one placed on the outlet pipe reflects the internal pressure exerted within the DUT. This parameter must be carefully monitored in order to preserve the integrity of the battery module and to prevent potential damage. A Facet temperature sensor [12] and a Sensata Technologies pressure sensor [13], used in this project, are visible in the Figure 2.9. Furthermore, supplementary NTC-type temperature sensors are positioned both inside and outside the DUT to measure its thermal conditions; these measurements are transmitted either to BMS or to the PSS board, depending on the configuration.

Another critical set of components is represented by the contactors, which serve as protective devices for the DUT. Contactors function as switches that open or close the circuit linking a cycler to the powered DUT. They are indispensable in critical conditions where current, voltage, or power—whether delivered or absorbed—reaches levels that could endanger the battery module. In such scenarios, the circuit must be interrupted rapidly and safely to prevent damage. When high voltage or current values are involved, the contactors are typically housed within the test cage to ensure proper containment and operator protection. For this thesis project LEV200H4ANA contactors [14] of TE Connectivity are used. TE Connectivity is a global industrial technology company that specializes in designing and manufacturing high-performance electrical and electronic connectors, sensors, and solutions. One of its contactors is shown in Figure 2.10.

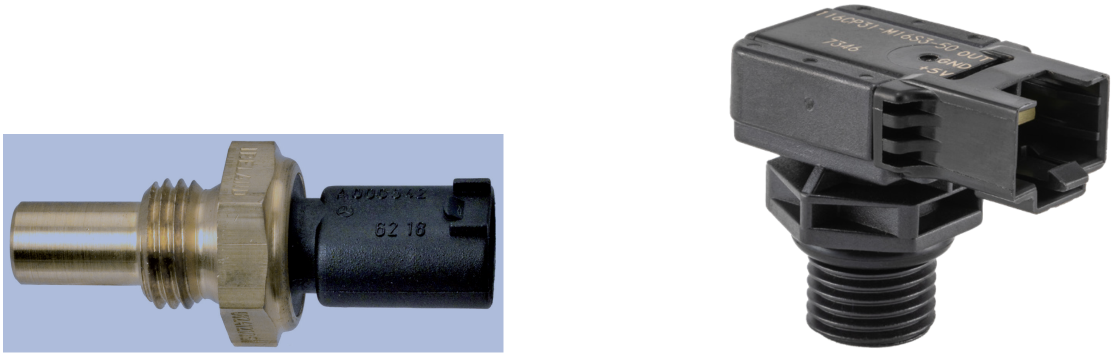

Figure 2.9: Facet temperature sensor (on the left) and Sensata Technologies pressure sensor (on the right)

Figure 2.10: Contactor LEV200H4ANA

Moreover in order to maintain proper control of the cyclators, shunt resistors are employed. These components act as voltage sensors, as they are characterized by a fixed resistance. By measuring the current flowing through the circuit, it is possible for them to determine the corresponding voltage with good accuracy. In the system, a shunt resistor is inserted at each connection between the cyclator and the DUT, thereby providing additional feedback. This configuration is necessary since the readings displayed by the cyclators are not always reliable to the hundredth.

Chapter 3

Architectural Design

This chapter outlines the system architecture and the rationale behind its design. The first section provides a detailed description of the hardware architecture, with particular emphasis on the board connectors and the physical cabling that interlinks the various components of the system.

The second section focuses on the system's software architecture. It includes an in-depth analysis of the Qt-based application running on the operator's PC, the firmware architecture of the board, and the communication protocols employed to exchange data across the physical connections linking the PSS board, the PC, and the associated machinery.

3.1 Hardware Architecture

For the purposes of this thesis project, a custom-designed hardware architecture was implemented.

The most critical element of the system is the proprietary PSS board developed by the company. Although the board was already available at the outset of the project, it underwent some modifications to meet the specific requirements of the experimental setup. Previously unused ports were activated for the first time, necessitating the creation of new electrical connections. Additionally, certain existing ports were reconfigured through manual interventions, including cable soldering and crimping operations.

Furthermore, custom communication cables were fabricated to match the physical interfaces and communication protocols supported by the various machines involved. These cables were essential for establishing reliable connections between the equipment and the two primary system components.

3.1.1 PSS Board

The PSS board, measuring approximately 24×14 cm, integrates a wide array of ports and onboard circuitry. Certain functionalities, such as Ethernet connectivity, are provided by the embedded TMS570 microcontroller, which occupies a significant portion of the board's surface.

The majority of the ports are implemented using Molex connectors, available in various configurations depending on the number of pins. For instance, a 10-pin Molex connector with 3 mm spacing between terminals is illustrated in Figure 3.1.

Below is a detailed list of the interfaces available on the PSS board:

- A 24-pin Molex connector dedicated to the power supply, delivering 12V to the board.
- A 14-pin Molex connector used to interface with three contactors. The male counterpart was manually crimped with 10 wires to ensure proper power delivery to the logic circuitry responsible for contactor control, as well as to establish the channels for transmitting open/close commands and receiving voltage feedback signals.
- A 12-pin Molex connector for CAN communication. The connector was wired to support two simultaneous CAN connections, each requiring three wires—two for the differential pair and one for ground. Consequently, not all connector slots were utilized.
- A 16-pin Molex connector intended for NTC temperature sensors. Since the corresponding female connector was unavailable, the two wires required for each NTC sensor were directly soldered onto the board's pins from the underside—all the Molex connectors are mounted on the top side of the board.
- Two 8-pin Molex connectors for interfacing with shunt sensors.
- A 20-pin Molex connector designated for digital signals, both input and output.
- Some additional Molex connectors reserved as spares, to accommodate future expansions or hardware upgrades.

All connectors are distributed along three of the four edges of the PCB, as can be seen in Figure 3.2. The fourth edge remains unused due to the spatial occupation of the TMS570 microcontroller.

For interfacing with the BMS, one of the CAN connections is utilized, while the second CAN channel is typically connected to the cyclator employed during testing. Communication with the host PC is established via the ETH port integrated into the TMS570.

Figure 3.1: Molex connectors 10 ports; male (on the left) and female (on the right)

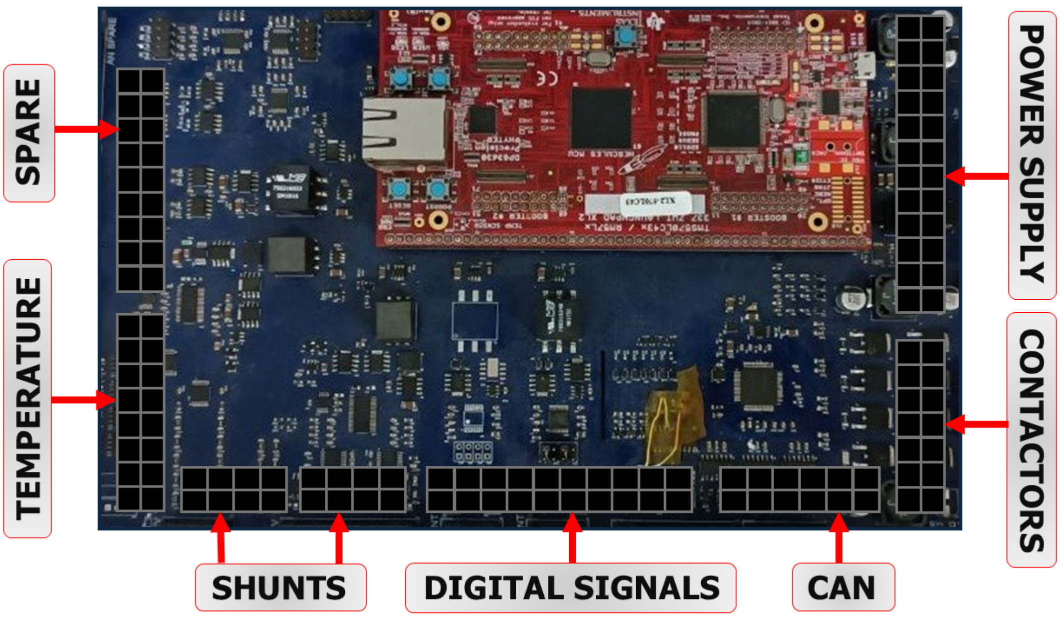

Figure 3.2: Schema of molex connectors over PSS board

3.1.2 Communication Cables

Regarding the physical interconnections among the various system components, multiple cables and ports are employed.

Firstly, the computer running the application is connected to the control board via an Ethernet cable. Since the computer lacks a native Ethernet port, a USB-to-Ethernet adapter is used to establish the connection, as the board is equipped with an Ethernet interface. The adapter is depicted in Figure 3.3. The same type of connection is used to interface with the climatic chamber.

Figure 3.3: USB - ETH adapter

The chiller is connected to the computer using a twisted pair cable with two signal wires (designated A and B) and a ground wire. The communication protocol adopted is the RS-485 serial interface standard (EIA/TIA-485), which supports multi-point communication over long distances and offers high noise immunity due to its differential signal transmission on the A and B lines. This configuration was chosen not only due to its compatibility with both the chiller and the computer, but also for its superior noise immunity. An essential feature given the cable length of several meters and the critical need for accurate acquisition of temperature, flow rate, and pressure data. A DP9 connector is used between the cable and the chiller. The three wires were selected and soldered as follows:

- A green wire for the non-inverting signal (line A), connected to pin 7.
- A yellow wire for the inverting signal (line B), connected to pin 8.
- A white wire for grounding, connected to the connector's ground pin.

The opposite end of the cable is soldered to a USB-to-RS485 adapter, which interfaces with the computer running the application. The adapter model is SH-U10B, featuring a CP2102 main chip (SILICON LABS) and a MAX13487 transceiver (MAXIM). It is depicted in Figure 3.4.

Regarding communication with the cyclers, both devices are equipped with a CAN interface. In this setup, two separate cable segments are used. The first cable is directly connected to the CAN port on the cycler. The opposite end of this cable terminates in a female DP9 connector. The second CAN cable features a male DP9 connector on one end, which mates with the female connector of the first cable. The other end of the second cable is connected to a dedicated CAN port on the PSS board. The DP9 connector is depicted in Figure 3.5.

Communication between the BMS and the PSS board is also handled via the CAN protocol. This ensures a robust and time-deterministic exchange of data between the battery management system and the supervisory control unit.

Figure 3.4: USB to RS485 adapter

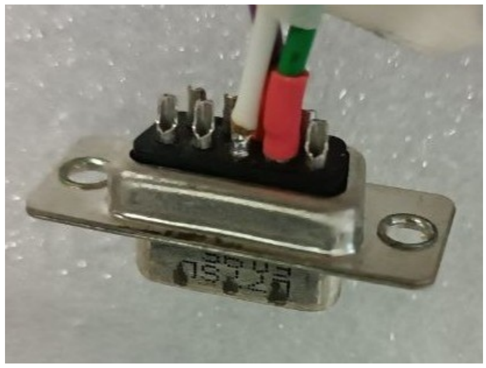

Figure 3.5: DP9 CAN connector

3.2 Software Architecture

The software architecture implemented in this thesis is composed of two interdependent components, each fulfilling a specific role within the overall system. This modular design ensures a clear separation of responsibilities between high-level user interaction and low-level hardware control, thereby enhancing maintainability, scalability, and robustness.

The first component consists of a desktop application, developed in `C++` using the `Qt` framework. This application is executed on the host computer and serves as the primary interface for the user. It provides functionalities for system configuration, real-time monitoring, data visualization, and logging. Through the graphical user interface, the operator can initiate test procedures, adjust control parameters, and observe system behavior in real time. The application also manages communication with the embedded system via a dedicated protocol, ensuring synchronization and data integrity.

The second component is the embedded firmware, written in `C`, which runs on the control board. This firmware is responsible for executing time-critical operations, including signal acquisition, actuator control, and communication with peripheral devices such as sensors, power supplies, and safety modules. It interprets commands received from the host application and translates them into low-level instructions suitable for hardware execution. Additionally, the firmware implements safety checks and feedback mechanisms to ensure reliable operation under varying test conditions.

Both components are designed to operate in a coordinated manner, with a well-defined communication protocol facilitating data exchange and command execution.

The following sections provide a detailed description of each component, including their internal structure, functional modules, and interaction mechanisms.

3.2.1 Qt Application

The software application provides a graphical user interface designed to support the operator during battery testing procedures. The development environment selected for this purpose is `Qt Creator`, due to its capability to facilitate the creation of dialog windows, each associated with a header file and an executable source file. These windows are defined using the `.ui` format, which, while not a proprietary file type, stands for *user interface* and refers to the collection of visual and interactive elements through which a user interacts with a digital system.

Each interface window is managed independently through its three constituent components: the header file, the source code file, and the UI definition file. These components interact in a manner analogous to traditional `.h` and `.cpp` file pairs in `C++`. Moreover, each window can be invoked and reused by other components within the application, promoting modularity and code reusability.

There are two primary approaches to developing a UI in Qt:

- Programmatic Approach: The interface is constructed entirely through code within the source file. This method offers maximum flexibility and control over widget behavior and layout management but lacks immediate visual feedback, as the interface can only be previewed upon execution.
- Visual Design Approach: This method leverages the integrated `Qt Designer` tool, which supports a *What You See Is What You Get* (WYSIWYG) paradigm. Developers can drag and drop widgets (e.g., buttons, text boxes) onto a form, arrange them using layout managers, and connect them to backend functions without writing code manually. This approach significantly accelerates the interface design process by providing real-time visual feedback.

All interfaces for this thesis were developed using the visual design approach.

The version of Qt Creator employed for the development is **14.0.2**.

Project Organization

The entire project is structured across multiple source files, each responsible for a specific functionality within the system. This modular organization allows the separation of concerns, where each module encapsulates a distinct aspect of the application's logic or interface. In many cases, a module is associated with a dialog window; however, this is not always the case. As a result, both file pairs (e.g., header and source) and triplets (header, source, and UI) are present within the codebase.

Each module is instantiated and utilized by the main project file, which creates the corresponding object and invokes its methods as needed. When a dialog window is associated with the module, the main file may also provide a method to display it. In such cases, the module is responsible for managing and storing the relevant data, including user actions and requests, ensuring that this information can be saved and applied appropriately during the application's execution.

The main file of the project is `main.cpp`. Its sole responsibility is to initialize and execute the application, primarily by instantiating the `MainWindow` object. Consequently, the core logic and user interaction mechanisms are encapsulated within the triplet of files associated with the `MainWindow` class. This window represents the initial interface presented to the operator upon launching the executable.

Through the use of push buttons and drop-down menus, the operator can access additional dialog windows, hereafter referred to as *secondary interfaces*. When interacting with a secondary dialog, the application records the user's actions. These interactions may include selecting checkboxes, radio buttons, or push buttons; entering values into spin boxes or text into line editors; and choosing options from combo boxes. Additionally, users can load external files and visualize their contents using table widgets.

All of these components are classified as *widgets*, which are graphical elements provided by the GUI framework. Widgets facilitate intuitive and efficient communication between the user and the application, enabling the collection of essential input data. In this context, such data may include machine configuration settings, operational parameters, and selected equipment.

3.2.2 Embedded Code

For this project, the previously developed design for the PSS electronic board was extended and enhanced. The original implementation of the company already incorporated a wide range of functionalities, including port management and handling of both physical and analog inputs and outputs. Additional features such as signal conditioning and error detection mechanisms were also integrated to improve system flexibility and reliability.

Since the embedded code developed for the electronic board is intended for the host company, its implementation adheres to the corporate coding guidelines [15]. These guidelines incorporate the MISRA standards, which are a set of coding rules designed to ensure the safety, reliability, and maintainability of software in electronic systems, particularly in embedded applications.

The primary modifications focused on the communication protocol, which was completely redesigned to introduce a new structure for message handling and control logic. Each communication packet consists of 64 bytes. The first byte serves as the message identifier (ID), while the remaining bytes may contain either payload data or supplementary metadata related to the transmission process.

Message types are defined using enumerations that are consistently implemented across both the embedded firmware and the Qt-based application interface, ensuring seamless interoperability and standardized message parsing. From the electronic board's perspective, the message IDs used for communication are as follows in Figure 3.6.

```
enum received_datagram_type
{
    GUI_ETH_CNTRL = 0x10,
    GUI_TEST_INFO,
    GUI_TEST_LIMITS,
    GUI_TEST_BEHAVIOR,
    GUI_INSTRUMENTATION,
    GUI_PRFL,
    GUI_MOBASS_PRFL,
    GUI_CMD_TEST_START,
    GUI_CMD_TEST_STOP,
    GUI_CMD_TEST_PAUSE,
    GUI_CMD_TEST_CONTINUE,
    GUI_CMD_TEST_PANIC,
    GUI_CMDS_DUT,
    BUFF_RESET,
    BL_UPLOAD_PSB,
    /* debug */
    GUI_SETPOINT_LVPS,
    GUI_SETPOINT_HVPS,
    GUI_FDBK_SENSOR_MEAS,
    EN_MC33816,
    GUI_AIR_1,
    GUI_AIR_2,
    GUI_AIR_4,
    GUI_VIN_CSV,
    GUI_VIN_VDC,
    GUI_VIN_VCELL
};
```

```
enum send_datagram_type
{
    FDBK_PS = 0x10,
    FDBK_DUT,
    FDBK_SH,
    FDBK_AIRS,
    FDBK_SENSOR_FLOW,
    FDBK_SENSOR_PRESSURE,
    FDBK_SENSOR_NTC,
    FDBK_CYCLTRS,
    ALARM,
    WARNING,
    BUFF_FREE,
    BUFF_RCV_OK
};
```

Figure 3.6: PSS board ID messages

In addition to modifying the existing codebase, new functionalities were implemented. At the initial stage of communication, the two endpoints are required to exchange a substantial amount of data related to the test setup and its operating environment. Moreover commands that must be written, transmitted, received, and parsed with minimal latency, as well as configuration parameters and runtime data that may need to be stored locally.

During the first iteration of the communication protocol, the board retrieves and stores the sender's IP address. This IP management is therefore dynamic for this phase. In contrast, the communication ports are statically defined within the source code of both the embedded system and the Qt-based application.

Following the establishment of the connection, the board receives the test configuration parameters, which include voltage, current, temperature, and optionally humidity thresholds; the expected behavior of the equipment in response to interrupts such as pause, stop, or panic conditions; and the instrumentation to be activated and managed throughout the test.

A critical aspect of the setup is the transmission of the test profile. The profile defines the target values that each involved machine must assume at specific time intervals. The first column specifies the duration for which the values in the corresponding row must be maintained, while the subsequent columns contain the actual setpoints. These values are stored in separate vectors, one for each column of the profile. For instance, a test scenario where a battery pack is controlled solely via current within a test cage would require a profile with two columns: one for time and one for current values.

Given that profiles can be large in size, a dedicated transmission mechanism was implemented to prevent excessive memory usage or data overwriting on the board. Configuration parameters such as thresholds, behaviors, and instrumentation settings are transmitted in bulk and stored locally. If their size exceeds the packet limit, they are segmented into multiple packets in a structured manner. In contrast, the profile data is transmitted using fixed-length buffers—for example, two buffers of 100 elements each. Once the configuration is complete, the first 200 values of a profile column are sent and stored sequentially in `buff1` and `buff2`. As the test begins, the board starts transmitting values from the first buffer to the machines. Upon exhausting the first buffer, the board signals the application that the buffer is free, prompting the transmission of the next 100 values. Meanwhile, the board continues operating using the second buffer, which will be overwritten with new incoming data once depleted. This procedure is repeated for each column of the profile, resulting in 2 columns $\times$ 2 buffers $=$ 4 vectors of 100 elements in the given example.

This mechanism ensures that the board always retains only the subset of data required for imminent operations, thereby optimizing memory usage while safeguarding against data starvation. The number and size of the buffers are determined primarily by the rate of change of the profile values and the overall length of the test sequence.

3.2.3 Communication Protocols

Several communication protocols are employed in this thesis project, due to the heterogeneous and numerous devices involved.

Firstly, the application running on the operator's computer communicates with the electronic control board via a UDP over Ethernet protocol. The User Datagram Protocol operates at the transport layer of the TCP/IP model and is used over Ethernet to transmit data rapidly without requiring a prior connection establishment with the recipient. As a connectionless and lightweight protocol, UDP is well-suited for time-sensitive applications such as video streaming, online gaming, and DNS queries, where transmission speed takes precedence over guaranteed delivery of individual data packets. In the context of this work, rapid data exchange is prioritized over communication reliability.

The climatic chamber interfaces with the application using the Modbus TCP protocol over Ethernet. Modbus TCP Ethernet (similar to Modbus RTU) is a standardized version of the Modbus protocol that leverages Ethernet networks for industrial communication, encapsulating Modbus data within TCP/IP packets for transmission over standard networks. Operating on a client-server model, it enables interoperability among devices from different manufacturers across local area networks and the Internet, offering enhanced flexibility, scalability, and speed compared to serial communication modes. This makes it particularly suitable for advanced automation and control systems. For each machine that provides such a communication protocol, the manufacturing company supplies customers with a dedicated `.xlsx` file listing all the addresses required to execute specific commands or to read particular values. Although the file is confidential, a sample of selected rows is presented in Table 3.1.

Read/Write	Address	Name	Type	Remark
READ	5450	Actual temperature	float	udm: °C
R/W	15513.01	Clima mode selector	bool	
R/W	15513.08	Open door button	bool	
WRITE	15518	Set humidity	float	-10.0 to +98.0 %rH

Table 3.1: Climatic chamber register map for remote control and monitoring

The chiller communicates with the application using the Modbus RTU protocol over RS485. Modbus Remote Terminal Unit is an industrial communication standard based on a master/slave architecture for data exchange between PLCs, computers, and other electronic devices over serial lines (RS-232 or RS-485). It employs binary encoding and a cyclic redundancy check error-checking mechanism to ensure efficient and reliable data transmission, making it ideal for industrial automation and building management systems. In this setup, the chiller functions as the slave device, while the application acts as the master. Also in this case the manufacturing company supplied the host company with a dedicated `.xlsx` file listing all the addresses required to read particular values or to execute commands. Although the file is confidential, a sample of selected rows is presented in Table 3.2.

Read/Write	**Address**	**Name**	**Type**	**Remark**
READ	40001	Liquid supply pressure	INT	acc: 0.01 bar
READ	40015	Forwarding of signals	INT	0 or 1
WRITE	40054	Remote start	INT	1: open, 0: close
WRITE	40055	Remote stop	INT	1: open, 0: close
WRITE	40058	Remote mode	INT	1: press, 0: flow

Table 3.2: Chiller register map for remote control and monitoring

The control board communicates with the two cyclers using the CAN 2.0B protocol over a CAN bus. This protocol is a serial communication standard for fieldbus systems, developed by Bosch for automotive electronics and industrial automation. It is designed for real-time data transmission among various electronic control units and sensors. Its key feature is the use of a 29-bit identifier in the message format, distinguishing it from CAN 2.0A, which uses an 11-bit identifier. CAN 2.0B is recognized for its efficiency, robust error detection and recovery mechanisms, and multi-master communication capabilities.

The same protocol is generally used for communication between the control board and the device under test, as well as between the board and the shunt modules.

Chapter 4

Implementation

This chapter addresses the implementation phase of the project. The initial section outlines the hardware setup required prior to commencing the software application activities. Test procedures may be conducted by positioning the DUT either within the designated test cage or inside the climatic chamber. The utilization of the chiller is optional; the selection of the cyclator is typically based on its voltage and current limitations. In scenarios requiring high-voltage testing, the high-voltage power supply is preferred, as the alternative unit lacks the capability to deliver the necessary output. Furthermore, if the device under test is equipped with a battery management system, an additional power supply unit will be required to support its operation.

The second section explains how the test is conducted and managed. Once the physical setup has been completed, the software configuration can begin. Possibly, communication settings are selected as needed for the climatic chamber, cyclers, and chiller. Communication with the involved components—previously powered on during the hardware setup—is initialized and verified. Through the application's graphical user interface the test data and the execution profile are loaded. Once all systems are ready and the preliminary checks have been successfully passed, the test can be started. During execution, the equipment reaches the assigned values of current, voltage, power, temperature, or humidity, maintaining them for the specified duration. If no anomalies occur, the test proceeds until completion. In this case, the feedback data collected during the process are saved in log files, and a new test can be initiated via the GUI. If an alarm is triggered during the test, predefined values are sent to the equipment to stop the procedure as quickly and safely as possible. It is essential to prevent damage to both the DUT and the laboratory instrumentation. The operator may also decide to interrupt the test manually by issuing a *pause*, *stop*, or *panic* command. Upon receiving a *stop* command, the system behaves as it would in the case of an alarm arose. Regarding the other two commands, the behavior is different and will be described later.

The third and final section of this chapter is dedicated to the presentation of a series of experimental tests conducted in a controlled laboratory environment. These tests were designed to validate the functionality and integration of the software system components developed throughout the thesis. Specifically, two of the tests focused on verifying the communication and control capabilities of external subsystems: the chiller unit and the climatic chamber. The tests confirmed the ability of the software application and the embedded control system to establish stable communication links, issue control commands, and receive feedback data in real time. The third test involved a battery module prototype developed by the host company. This module required functional assessment prior to its deployment in further experimental campaigns. The test provided a valuable opportunity to apply the software interface and the PSS board in a realistic operational scenario. Through this practical application, the centralized management system was shown to offer significant advantages in terms of usability and coordination. Moreover, the testing phase facilitated the identification of several areas for potential improvement.

4.1 Physical Setup

Before initiating the actual test procedure, several preliminary steps are required. Regarding the preparation of the working environment, it is essential to arrange the laboratory equipment—particularly the portable components—on the bench in the most ergonomic and accessible manner. For instance, the PSS board must be connected to the operator's PC, the cyclers, and the DUT, and therefore should occupy a central physical position.

As for equipment with fixed or non-adjustable placement—such as the chiller, climatic chamber, and test cage—special attention must be paid during the creation of custom cabling, taking into account the distance between these devices and the bench, as well as the rest of the instrumentation to be used.

At this stage, physical connections between the various system components can be established using the cables and adapters described in 3.1.2, and all necessary instrumentation is powered on.

The numerous sensors and additional elements described in 2.10 must be positioned accordingly. The sensors monitoring flow, temperature, and pressure of the chiller are already integrated into the hydraulic structure developed by the company to connect the DUT to the chiller; therefore, they only need to be connected to the electronic board. Supplementary NTCs may be added at points of interest—on the DUT, along specific cables, or on the board—and connected to the available ports on the PSS board.

Regarding the contactors, these are placed between the cycler and the DUT, managing the direct connection between the two; they may be positioned inside the test cage. The cables used for switching operations are routed directly from the contactor itself. Feedback signals, on the other hand, are acquired via external cables terminated with ring connectors, which are affixed to bolts protruding a few centimeters from the upper side of the contactor.

The use of contactors introduces additional complexity to the physical setup, as specific precautions must be taken prior to their closure. The voltage present at both terminals of the contactor must be matched with a precision of approximately 0.02 V. To achieve this, the voltage of the battery module—or, in general, of the DUT—must be read via its BMS, and the output voltage of the cycler must then be manually adjusted to match it. However, the cables connecting the cycler to the contactor are not ideal and exhibit a certain voltage drop. Consequently, the voltage set on the cycler must always be slightly higher than the value read by the BMS on the DUT. For accurate calibration—which is essential—the voltage must be measured directly at the contactor on the cycler side, for example using a multimeter, and verified to be virtually identical, down to the hundredth of a volt, to the value reported by the BMS. This procedure must be performed manually during the test setup phase, as it ensures proper voltage alignment prior to contactor closure.

The contactors themselves provide a voltage feedback signal that is read by the board and transmitted to the application and it can be used instead of the multimeter during the closure operation. To read such data, the use of shunts is also foreseen; these are placed along the connection between the DUT and the cycler to provide an additional and more precise feedback of the instantaneous active voltage.

4.2 Test Management

The preparation and execution of each test procedure are entirely managed by the operator through the main window of the GUI. As illustrated in Figure 4.1, the interface provides a comprehensive set of options immediately upon launching the executable, enabling intuitive and structured interaction with the system.

The top navigation bar includes several drop-down menus, each corresponding to a specific functional domain:

- Settings: grants access to the communication parameters of external subsystems, specifically the chiller unit and the climatic chamber. These settings allow the operator to configure serial or network protocols settings.
- Configuration: provides access to both system-level configuration and test-specific setup, including the initialization of specific battery module experiments. These options are also accessible via dedicated push buttons located in the lower section of the interface.

- Diagnostic: opens diagnostic pages for the chiller, climatic chamber, and the electronic board. Within the PSS diagnostic page, the operator can also verify the operational status of the cyclators, ensuring proper functionality before initiating a test.
- Feedbacks: displays real-time data during test execution, segmented into three dedicated pages. These include feedback from the chiller, the climatic chamber, and all components interfaced with the electronic control board—such as cyclators, sensors, and connectors.

In addition to the menu bar, the GUI features several push buttons that guide the operator through the test preparation workflow. Located in the upper-left section of the window, these buttons allow the user to sequentially:

1. Configure the general test parameters.
2. Set up the battery module test experimental conditions, if applicable.
3. Validate the configuration to ensure all required components are correctly initialized.

Once this sequence is completed successfully, the `START` button becomes enabled, allowing the test to begin.

During test execution, additional buttons appear on the right-hand side of the interface, providing the operator with the ability to generate interrupts if necessary. These interrupts can be used to pause, continue, or terminate the test in response to unexpected conditions or manual intervention.

In the lower-left section of the GUI, diagnostic buttons are available for inspecting specific subsystems. Throughout the experimental phase of this thesis, these buttons were frequently modified—added, removed, or reconfigured—to accommodate evolving architectural requirements and to validate the implementation of new functionalities. This iterative refinement process contributed to the robustness and adaptability of the software platform.

4.2.1 Setup Connections

For managing communication with the chiller and the climatic chamber, valid configuration parameters are set according to the specific versions of the two devices currently available in the company's laboratory. Should these devices be replaced or the project be deployed in different contexts, two graphical interfaces are provided to the user, allowing for the modification of connection settings.

Specifically, as shown in Figure 4.2, the following parameters can be configured for the chiller connection:

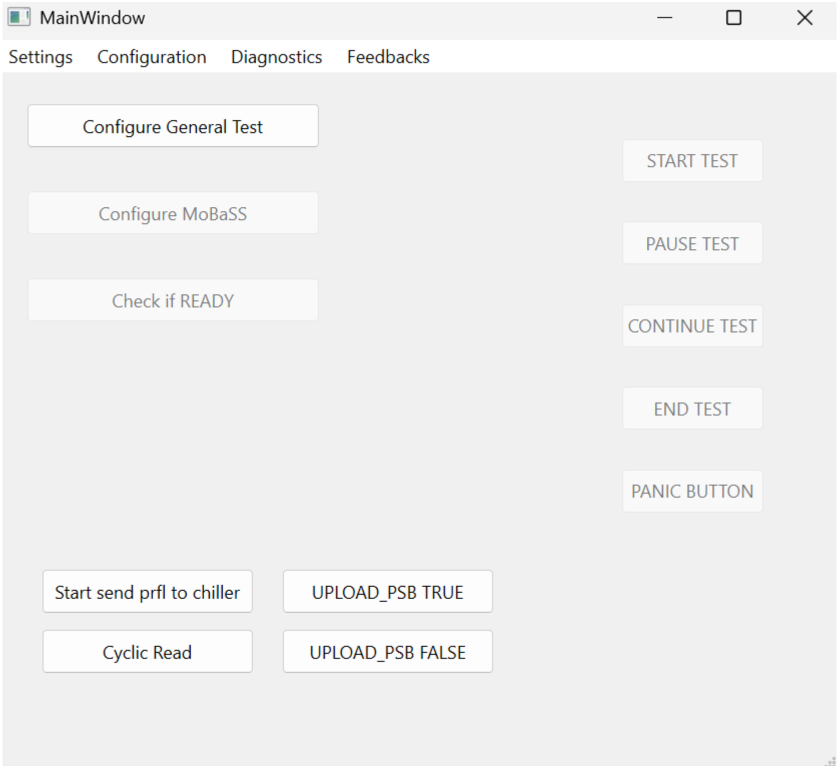

Figure 4.1: Main window of the GUI

- the port name, currently `COM3`.
- The number of data bits contained in a single packet.
- The server address; the value `1` is used in this project, as shown in the figure, because there are only two participants to the RTU serial connection over RS485.
- The type of the parity bit, that is the additional bit used for error detection during data transmission or storage; it can be `even`, `odd`, `space`, `mark` or not present.
- The baud rate, a very important setting in a communication protocol. It is the transmission rate of the messages and both the transmitter and receiver must be set at the same baud rate to ensure proper synchronization and reliable data exchange.
- The number of stop bits. They are signals that indicate the end of a character or block of data and provide a time interval for the receiver to process the received data.

- The response timeout, that is the waiting time for confirmation of correct receipt of the message.
- The maximum number of packets to send if no response is received, called number of retries.

For the climatic chamber connection, less settings can be chosen, as shown in Figure 4.3. This is due to the fact that the connection is ETH. In particular, the following parameters can be configured for the climatic chamber connection:

- the URL for the connection, containing the IP address and the port.
- The server address. Also in this case `1` is used.
- The response timeout.
- The number of retries.

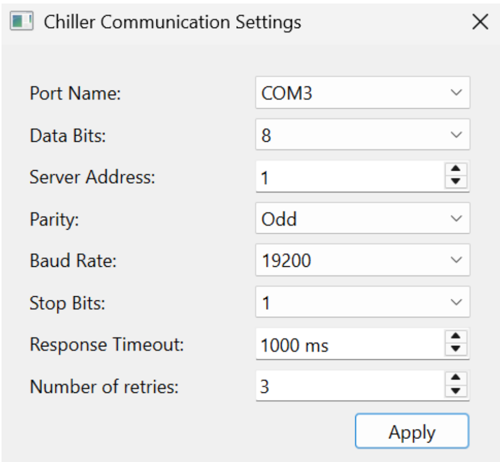

Figure 4.2: Chiller communication settings window

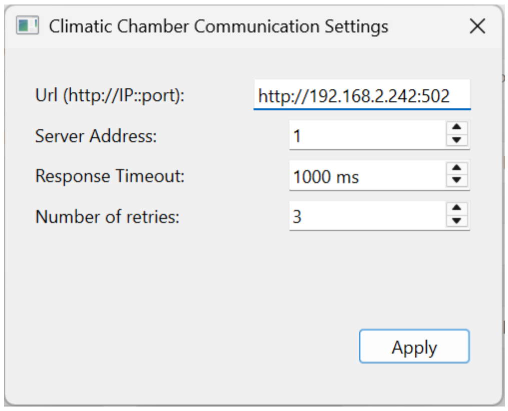

Figure 4.3: Climatic chamber communication settings window

4.2.2 Test Configuration

Once the physical configuration of the system and the communication interfaces has been completed, the next essential step is to configure the test procedure. This involves opening the dedicated interface—accessible in the *MainWindow*, either via the push button *Configure Test* or through the drop-down menu *Configuration*—and filling in all required fields.

A key part of this configuration process is specifying which laboratory instruments will be employed during the test. If the chiller is selected, an additional field will appear prompting the user to define the operating mode—either `pressure` or `flow`. Similarly, if the climatic chamber is selected, the interface will request the operating mode—`thermo` or `clima`—as well as the optional selection of any preloaded recipe available on the device. It is important to note that the selection of the test cage automatically excludes the use of the climatic chamber, and vice versa, due to mutual exclusivity in their operational contexts.

Subsequently, the user must upload the relevant files, namely the test configuration file and the execution profile. These files must be saved in `.csv` format, with a specific naming convention and stored within the directory path defined by the software. By pressing the `Load` button, the files are read line by line, parsed into structured data, and displayed within the multi-tabbed table located in the lower-right section of the GUI. This immediate visual feedback allows the user to verify the correctness of the loaded data; if any discrepancies are identified, the files can be modified and reloaded using the same button.

An example of a ready configuration is shown in Figure 4.4.

Once the configuration is complete, the user proceeds by applying the changes (using the `Apply` push button bottom-right). The configuration window closes, returning to the `MainWindow`, where additional information becomes visible—specifically, the operational limits to which the test will be subjected. A representative example of this post-configuration view is provided in Figure 4.5.

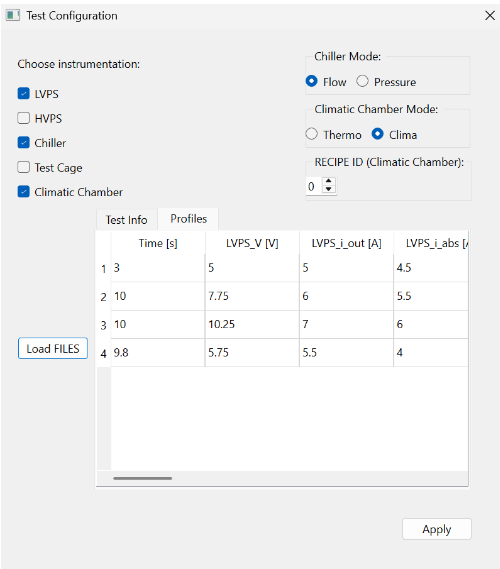

Figure 4.4: Test configuration window

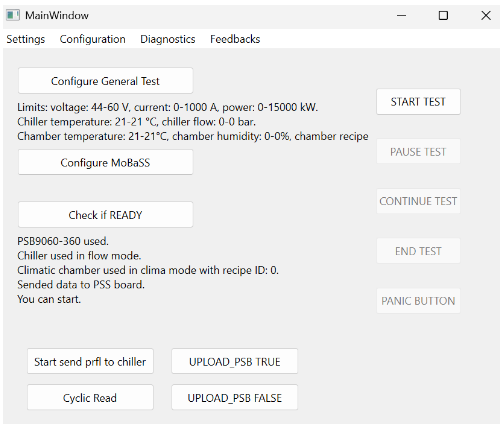

Figure 4.5: Main window of the GUI, after test configuration

4.2.3 Test Ongoing Management

Once the configuration has been selected, the relevant data loaded, and all preliminary checks successfully completed, the test procedure can be initiated. To ensure operational safety and consistency, the `START` button remains disabled until all required steps have been correctly executed. This mechanism prevents accidental or premature activation of the test sequence.

Upon initiation, a status message appears on the `MainWindow`, clearly indicating that the test is in progress. This message remains visible throughout the duration of the test and will only change upon successful completion or if an interrupt signal is received, which triggers an almost immediate and controlled interruption of the procedure.

During execution, the operator has access to secondary feedback windows that display real-time data acquired from the various sensors and peripheral devices. These data include key parameters such as voltage, current, temperature, and flow rate, depending on the instrumentation involved. Additionally, graphical representations of the data are available through a dedicated interface previously developed by the company, allowing for intuitive monitoring and analysis.

While the test is running, certain actions can be performed via specific push buttons integrated into the GUI. These buttons generate interrupt signals that are processed by the application and transmitted to the electronic control board. Both the software and the board then propagate the instructions to the connected subsystems, guiding them in executing the appropriate response protocols.

As a result, the operator assumes a supervisory role during the test, primarily observing the behavior of the DUT through sensor feedback and the BMS. Although the system is designed to operate autonomously, the operator retains the ability to intervene if necessary—either to halt the test prematurely or to respond to unexpected conditions—ensuring both flexibility and safety throughout the procedure.

Data Acquisition

During the execution of the test, a substantial amount of data is continuously acquired from various sources distributed throughout the system. These include sensors positioned across the setup, contactors, shunt resistors, and the laboratory equipment itself. The electronic control board collects all incoming values, performs real-time analysis to detect potential anomalies, and immediately transmits the processed data to the application running on the operator's computer.

Each measured value is evaluated against predefined acceptable ranges. If a value slightly exceeds either the upper or lower threshold, a warning is generated to alert the operator. Should the value break a second, more critical boundary—entering a zone significantly outside the permitted range—an alarm is triggered, and the system automatically transitions into a *stop* state to ensure safety and prevent damage.

Once the data have been analyzed, the board does not retain them in memory. Instead, it forwards them directly to the application, which temporarily stores the received values in dedicated data structures. These values are then displayed in real time within the feedback windows of the GUI and simultaneously written to log files in `.csv` format.

The same procedure is applied to data received directly from the chiller and the climatic chamber, provided these devices are used during the test. Their operational parameters are monitored and processed in the same manner as other system components.

Real-time visualization focuses exclusively on the parameters relevant to the operator's supervision tasks, enabling prompt identification of anomalies that may not be automatically flagged by the software. Upon completion of the test, all acquired data—regardless of their immediate relevance—are available in the `.csv` log files for post-analysis and documentation purposes.

Interrupt Commands

The commands that the operator may issue during test execution are as follows:

- **Stop**: This command requests the premature termination of the test. In this case, the two main components—PSS board and Qt application—refer to the behavioral instructions defined during the configuration phase, as described in Section 4.2.2. The transmission of profile values is suspended. It is not permitted to abruptly set voltage and current to zero, nor to request sudden changes in temperature or humidity. Instead, a short stop profile is initiated, lasting a few seconds, which gradually yet promptly adjusts the output parameters of the equipment to ensure a safe shutdown.

- **Panic**: This command represents a more critical intervention than the previous one. Initially, a similar mechanism was considered, involving more rapid changes in operating conditions. However, to emulate a true emergency stop button, a more immediate response was required. Consequently, the *panic* interrupt was designed to open the contactors instantly, without first reducing the current. While this approach may damage the contactors—since they are rated for a limited number of such abrupt interruptions—it effectively prevents harmful electrical stress from reaching the DUT, thereby prioritizing its protection.

- **Pause**: This interrupt was originally conceived to temporarily suspend the test, maintaining the current operating values indefinitely. These values would only be modified upon receiving a *continue* command or another interrupt. Although theoretically sound, this approach proved to be unsafe in practice. For instance, leaving a battery module under high voltage or current for extended periods could lead to significant overheating and potential damage. For this reason, the *pause* interrupt was removed from the project and has never been used with connected DUTs.

4.2.4 Test Termination

Upon transmission of the final row of the test profile and the expiration of the associated execution time, the test procedure is formally concluded. At this point, all data acquisition and monitoring processes are halted, and the system transitions into a post-test state. The log files generated during the test are properly closed and stored in the default directory specified by the software configuration. These files remain accessible for subsequent analysis and documentation purposes.

Any auxiliary subsystems, such as the chiller unit and the climatic chamber, are automatically deactivated. However, they remain powered on in standby mode, ready to be reinitialized for future test cycles. The same behavior applies to the **cyclators**, which gradually reduce their output parameters—voltage, current, and power—to predefined default values. This controlled ramp-down ensures that the battery module remains in a safe and stable condition, avoiding abrupt transitions that could compromise its integrity.

Specifically, in order to safely open the contactors, it is imperative that the voltage supplied by the cyclator matches the instantaneous voltage of the battery module, as reported by the BMS. This precaution mitigates the risk of interrupting the electrical circuit under a voltage differential, which could potentially damage the DUT. Furthermore, to avoid degradation of the contactors, before to open them the current must remain below a specified threshold, typically indicated in the datasheet of the respective contactor model. In laboratory, a conservative upper limit of 5 A is imposed to ensure operational safety and component longevity.

On the graphical user interface, a confirmation message is displayed: *The test ended correctly.* This visual feedback informs the operator of the successful completion of the procedure. Additionally, the interface presents the option to initiate a new test cycle. The `Start` push button is once again disabled, indicating that a new configuration process must be completed before the system can resume operation. This design choice significantly facilitates the execution of multiple consecutive test cycles.

4.3 Performed Tests

In order to validate the software implementation developed in this thesis, a series of practical tests were conducted using the custom-designed hardware system. These preliminary trials were essential for verifying the correct operation of each subsystem individually, prior to their integration into the complete architecture.

The tests were performed sequentially, targeting specific components such as the PSS board, the cyclators, the chiller unit, and the climatic chamber. Each module was subjected to controlled stimuli to assess its responsiveness, stability, and communication capabilities. These tests were executed in the absence of a DUT, focusing solely on the instrumentation's behavior and its ability to process commands and return feedback accurately.

This modular validation approach enabled the identification and resolution of potential issues at an early stage, ensuring that each subsystem was fully functional and ready for concurrent operation within the integrated test environment. The successful completion of these trials confirmed the reliability of the communication protocols and the readiness of the system for full-scale testing.

The final validation step consisted of a comprehensive test involving a real DUT: a battery module specifically designed and manufactured by the host company. This test aimed to simulate realistic on-road conditions, including dynamic charge and discharge cycles governed by predefined timing and frequency profiles. The battery was subjected to varying electrical loads to emulate acceleration and regenerative braking scenarios, thereby replicating its expected behavior in an electric vehicle context.

4.3.1 PSS Board Communication Validation

A fundamental step in the development and verification process was the validation of the custom communication protocol designed to facilitate data exchange between the desktop application and the embedded control board. The initial focus was placed on ensuring the continuous transmission of control messages, which serve to monitor and maintain a stable and active communication link. This persistent exchange forms the backbone of the system's reliability, enabling real-time responsiveness and fault detection.

Subsequent tests were conducted to verify the correct reception and execution of commands issued by the operator, from the GUI to the control board. Particular attention was given to the transmission of initial configuration data structures, including system parameters, stop profiles, and test-specific settings. Interrupt commands were also tested to ensure they could be reliably triggered and processed under various conditions.

During active test phases, the protocol was further validated through the exchange of dynamic data, such as feedback from sensors, cyclators, and connector states.

Remote control of the cyclators was also a key aspect of the validation process. The system was tested for its ability to set target values for current, voltage, and power—both in sourcing and sinking modes—as well as to activate and deactivate the cyclator outputs. These operations were essential for verifying the correct communication with the two PSB9000.

PSS Board Diagnostic Interface

To facilitate these tests, several push buttons available on the main window of the GUI were utilized. However, the majority of the validation was performed using a dedicated diagnostic interface specifically designed for the PSS board and its associated cyclators, as shown in Figure 4.6. This diagnostic window provided granular control and monitoring capabilities, allowing for precise verification of each subsystem.

One particular challenge encountered during communication testing was the accurate translation of data before and after transmission. To address this, the GUI includes functionality for sending predefined values—such as zeroes or sequential numbers—to verify correct encoding and decoding across the communication channel. The interface also allows the operator to issue interrupt commands, control the opening and closing of all three connectors, and verify that these commands are executed only when the relevant control pins are active.

Midway through the diagnostic interface, a dedicated section is available for cyclator-specific testing. Here, the operator can assign target values and verify their correct application by observing the digital displays on the cyclators themselves. This ensures that the transmitted data is correctly interpreted and applied by the hardware.

Finally, the lower portion of the diagnostic window includes controls for selecting the mode by which values are transmitted to the cyclators. Three modes are supported:

- `CSV` mode: values are read from a configuration profile in `.csv` format, loaded during test setup.
- `VDC` mode: values are manually assigned via a company interface, for direct control.
- `VCELL` mode: values are dynamically adapted based on real-time battery feedback, allowing for responsive control.

This flexible architecture enables comprehensive testing and fine-grained control, ensuring that all components of the system connected with the PSS board, and the board itself, operate reliably and cohesively.

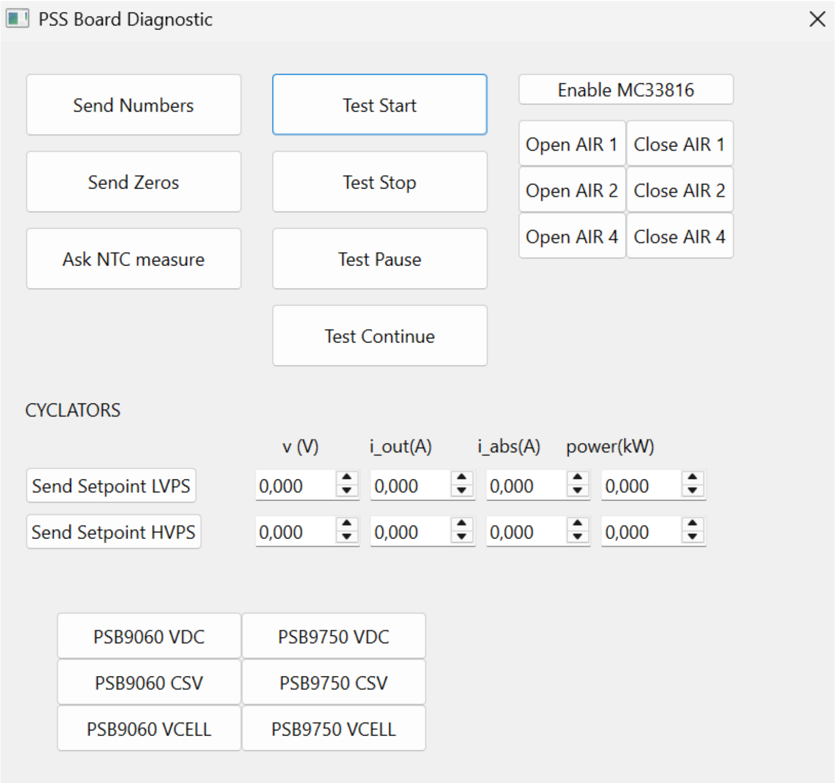

Figure 4.6: PSS board diagnostic window

4.3.2 Chiller Communication Validation

The verification of communication and control between the chiller unit and the desktop application proved to be relatively straightforward compared to other subsystems. The primary objective was to ensure that the RTU protocol over RS485 was correctly established upon request and maintained a stable connection throughout the interaction.

Initial tests confirmed that the chiller could be reliably started and stopped via software commands issued from the GUI. Furthermore, the system successfully transmitted configuration parameters and operational modes to the chiller, which responded appropriately by adjusting its internal settings. These operations validated the correct implementation of command encoding and transmission logic. The correct circulation of the cooling fluid was at the same time assessed. This was performed by routing the fluid through an empty battery module casing and monitoring its thermal behavior. The temperature of the coolant was evaluated both via dedicated sensors integrated into the hydraulic recirculation system and through direct tactile inspection—by placing a hand on the tube carrying the fluid.

An additional feature of the communication protocol is the ability to issue a `Read`

`Data` request to the chiller. Upon receiving this command, the chiller responds by transmitting a comprehensive set of internal parameters and status information. This functionality is essential for real-time monitoring and diagnostic purposes.

All of these interactions are facilitated through the dedicated chiller diagnostic interface, shown in Figure 4.7. This window provides a visual workspace where diagnostic data are displayed, including the commands sent to the chiller and the corresponding responses. In particular, when the `Show Data` function is activated, the retrieved data are rendered within the interface, allowing the operator to inspect the current state of the chiller in detail.

4.3.3 Climatic Chamber Communication Validation

A verification procedure analogous to that performed for the chiller was conducted to assess the communication and control capabilities between the climatic chamber and the Qt-based desktop application. In this case, the connection is established via the Modbus TCP protocol over Ethernet.

Once the physical connection is made using a standard Ethernet cable, the operator can manage the communication through the dedicated diagnostic interface, shown in Figure 4.8. This interface allows the user to initiate and terminate the connection with the chamber, start and stop its operation, and configure key parameters such as temperature and humidity setpoints.

Similar to the chiller, the climatic chamber supports internal data retrieval via a `Read All` command. Upon request, the chamber transmits a structured set of internal variables, including current operating conditions, sensor readings, and system status indicators. These data are then displayed within the diagnostic interface, providing the operator with real-time feedback and facilitating system monitoring and validation.

4.3.4 Battery Module under Test

Following the completion of the thesis project implementation—which included both a custom-designed hardware architecture and a fully developed software platform, whose functionality had already been validated—it was possible to perform a comprehensive test on a DUT.

The DUT in question was commissioned to the host company and consists of a battery module intended for use in electric vehicles. The module comprises 12 energy cells arranged in series, with each cell configured in a parallel pair, yielding an aggregate of 24 cells. The battery system is monitored and managed by an integrated BMS, which oversees cell balancing, thermal regulation, and protection mechanisms during operation.

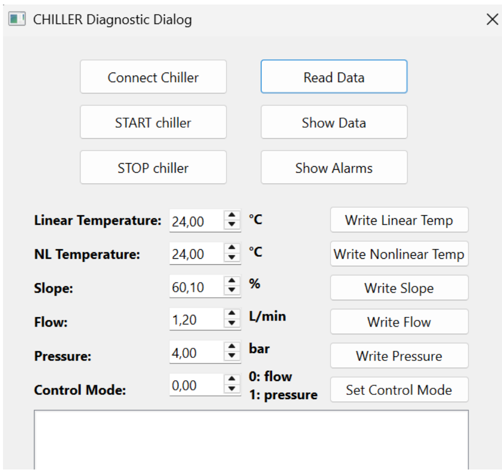

Figure 4.7: Chiller diagnostic window

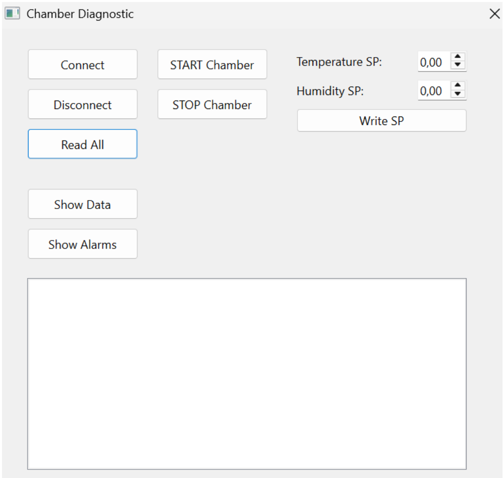

Figure 4.8: Climatic chamber diagnostic window

Laboratory Setup for DUT Testing

To conduct the test procedure, a dedicated area of the laboratory is prepared to accommodate the necessary instrumentation and ensure safe and efficient execution. The DUT, consisting of a battery module, is initially positioned near the test cage and the cyclators using a pallet jack, allowing for precise placement without manual lifting.

The BMS of the battery module is housed within the test cage and powered by a Rigol DP800 programmable power supply, which is positioned adjacent to the cyclators.

The LVPS, positioned near the test cage, is then connected to the battery module via a dedicated connector, which is physically placed inside the test cage to ensure electrical isolation and operator safety. This connector serves as the interface between the power supply and the DUT, enabling controlled energy exchange during the test.

Subsequently, the PSS board is positioned on top of the test cage and connected to the cyclator via a CAN bus interface. The PSS board is also interfaced with the contactor, which it controls by issuing open and close commands. In return the board receives feedback signals from the contactor.

The operator—positioned at a nearby workstation—uses a laptop running the Qt-based application to manage the test. The laptop is connected to the PSS board via a standard Ethernet cable, establishing the primary communication link between the software interface and the embedded control system.

This physical arrangement ensured clear separation between high-voltage components and the control interface, while maintaining accessibility for monitoring and intervention. The setup is designed to support both automated and manual control operations, facilitating a comprehensive and safe validation of the system's functionality.

DUT Testing

Several tests were conducted using the setup previously described. The company that commissioned the battery module provided a set of current and power profiles to be executed. These profiles were delivered as `.txt` files, each containing two values per line: the first represents the time index, while the second indicates the target value—either current or power—to be applied by the cyclator. The first rows of one of this `.txt` files are shown in Table 4.1.

These profiles are designed to simulate the behavior of the battery module under realistic vehicle operating conditions, allowing control either in current mode or power mode. Three distinct scenarios were considered:

- Short-Duration Urban-Suburban Driving: The first scenario consists of a two-part driving route: an initial urban segment followed by a suburban stretch. The urban portion is characterized by frequent stops and starts due to traffic lights, pedestrian crossings, and other obstacles. Consequently, the current or power values fluctuate rapidly but remain within moderate ranges. In contrast, the suburban segment involves higher maximum speeds and longer acceleration and deceleration phases, resulting in profiles with more pronounced peaks and sustained gradients. The files corresponding to this scenario are labeled `CurrentFrugal` and `PowerFrugal`; the plot of these profiles is depicted in Figure 4.9.

- Extended-Duration Urban–Suburban Driving: The second scenario mirrors the structure of the first, consisting of an urban segment followed by a suburban stretch. The key difference lies in the duration: this test spans approximately 45 minutes, compared to the 17-minute runtime of the shorter version. The extended timeframe allows for a more comprehensive evaluation of the battery module's behavior under prolonged dynamic conditions. Another notable distinction is that the peak current and power values—both maximum and minimum—are lower than those observed in the shorter profile, resulting in a less aggressive but more sustained load pattern. The files corresponding to this scenario are labeled `CurrentDual` and `PowerDual`; the plot of these profiles is depicted in Figure 4.10.

- Highway Driving: The third profile simulates a highway journey. It begins with a steady acceleration from a standstill up to a cruising speed of 130 km/h, which is then maintained for about 20 minutes. This test was intended to evaluate the battery's behavior under prolonged high-load conditions. However, due to thermal constraints and overheating risks, the full execution of this profile was not completed as originally planned. The files corresponding to this scenario are labeled `Current130` and `Power130`; the plot of these profiles is depicted in Figure 4.11.

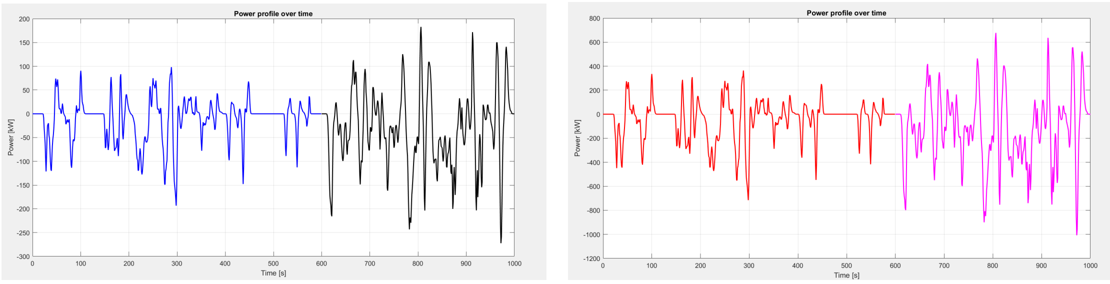

Figure 4.9: Short-duration urban-suburban driving profile

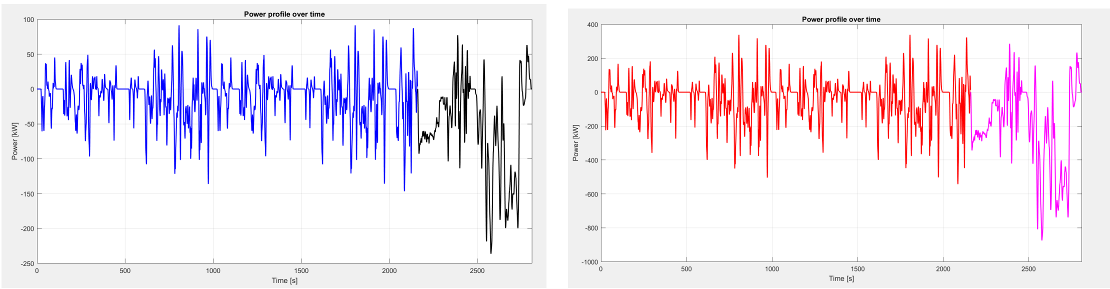

Figure 4.10: Extended-duration urban-suburban driving profile

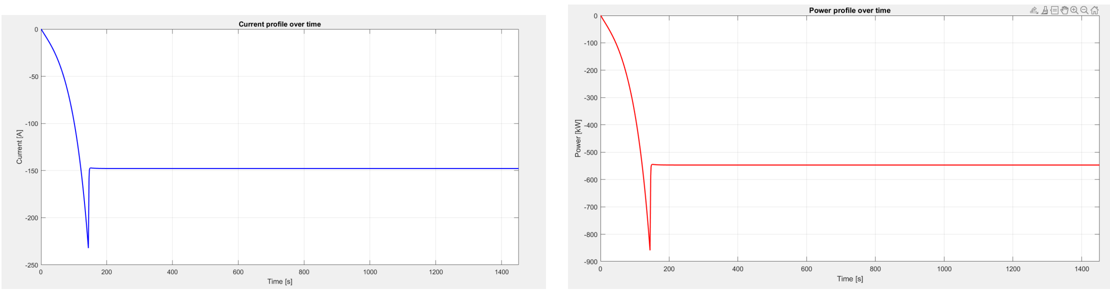

Figure 4.11: Highway driving profile

Figure 4.12: Profiles of current (on the left) and power (on the right) over time. Different colors in same diagram distinguish urban areas from suburban ones.

Time[s]	Current[A]	Time[s]	Current[A]	Time[s]	Current[A]
0	-0.00	35	-25.65	70	-10.24
1	-0.00	36	-37.27	71	-7.96
2	-0.00	37	-50.54	72	-6.02
3	-0.00	38	-58.70	73	-8.18
4	-0.00	39	-60.09	74	-9.09
5	-0.00	40	-47.21	75	-6.98
6	-0.00	41	-34.39	76	-5.76
7	-0.00	42	-26.83	77	-15.14
8	-0.00	43	-22.84	78	-29.71
9	-0.00	44	-15.86	79	-43.52
10	-0.00	45	-6.69	80	-53.91
11	-0.00	46	10.63	81	-56.82
12	-0.00	47	30.05	82	-48.20
13	-0.00	48	37.04	83	-35.60
14	-0.00	49	32.68	84	-27.93
15	-0.00	50	28.21	85	-26.87
16	-0.00	51	32.48	86	-28.87
17	-0.00	52	36.16	87	-26.74
18	-0.00	53	28.69	88	-17.75
19	-0.00	54	15.57	89	-4.39
20	-0.00	55	6.20	90	1.39
21	-0.00	56	4.75	91	8.86
22	-0.24	57	5.69	92	4.39
23	-4.44	58	3.85	93	0.87
24	-15.05	59	0.29	94	4.41
25	-22.46	60	0.35	95	4.78
26	-30.63	61	5.86	96	5.13
27	-47.19	62	10.73	97	2.97
28	-60.69	63	6.59	98	16.51
29	-44.70	64	1.71	99	38.02
30	-24.23	65	-0.49	100	45.27
31	-12.51	66	-1.24	101	38.61
32	-11.06	67	-2.52	102	22.43
33	-9.89	68	-3.28	103	12.81
34	-22.61	69	-11.00	104	10.48

Table 4.1: First rows of the *CurrentDual* profile

Chapter 5

Evaluation and Results

This chapter presents an evaluation of the outcomes achieved throughout the thesis project, with a focus on identifying effective design choices as well as limitations and inconsistencies. Particular attention is given to assessing the functional validity of the implemented solutions through a qualitative analysis of system components. The evaluation emphasizes performance, robustness, and compliance with initial design specifications, while also addressing usability, communication protocols, and sensor feedback to provide a comprehensive understanding of the system's operational behavior.

In this context, the results obtained from experimental tests are examined to verify the intended use and expected interaction behavior of the user interface, the machinery, and the control board. Observations are made regarding the responsiveness of commands, the consistency of timing coordination and the reliability of communication between subsystems.

5.1 Evaluation

This section provides a critical assessment of the system's operational features, highlighting both effective design choices and areas requiring improvement. The evaluation is based on empirical observations gathered during testing and integration phases.

5.1.1 Strengths and Effective Design Choices

With regard to data acquisition, the integration of pressure, temperature, and flow sensors alongside the chiller establishes a robust mechanism for cross-verification. In practical applications, data obtained from sensors mounted directly on the piping are generally deemed more reliable. Furthermore, acquiring readings from both the contactor sensors and the power cyclers introduces redundancy, which is particularly advantageous given the limited precision typically associated with cyclers. The system also benefits from dual environmental feedback sources—namely, the BMS, via sensors on the DUT, and the climatic chamber—thereby enhancing the accuracy of environmental monitoring.

A unified graphical user interface substantially improves system usability by consolidating control and visualization of nearly all operational parameters, commands, and feedback. Through the GUI, operators can initiate, halt, and test individual components or entire subsystems, maintaining comprehensive control over system behavior. Dedicated diagnostic windows for each machine and for the control board facilitate targeted testing and verification of communication protocols and response times, thereby streamlining troubleshooting and performance validation.

Centralized timing coordination across all machines ensures synchronized activation and deactivation, which is essential for preserving test integrity and reproducibility. Real-time monitoring of feedback from sensors, machines, and the BMS enables continuous supervision of system status. Any detected anomalies immediately trigger warnings or alarms, allowing operators to intervene promptly and mitigate potential failures.

Communication between the control board and the Qt application is secure, fast, and consistently reliable, contributing to stable system operation. The architecture supports timed transmission of buffered profiles, with both buffer length and quantity being configurable directly within the code. This design choice ensures efficient handling of large-scale profiles without compromising performance.

Finally, the system architecture is designed to support straightforward replacement and reconfiguration of instrumentation, thereby offering flexibility for future upgrades or experimental modifications.

5.1.2 Limitations and Areas for Improvement

Several limitations have been identified that impact the overall efficiency, usability, and maintainability of the system. One notable issue concerns the GUI, which currently displays feedback in separate windows for each machine. This fragmented visualization complicates monitoring and reduces operational efficiency. A unified interface offering consolidated feedback would significantly enhance usability. Moreover, the GUI does not expose the full range of functionalities supported by the control board. Some operations, such as opening or closing the contactors and advanced interactions with the DUT's BMS, are not currently accessible through the GUI, limiting the operator's control capabilities.

The contactor activation process remains manual and critical. Operators are required to visually monitor sensor readings and manually adjust cycler values, introducing potential risks and variability in test execution. This reliance on manual intervention compromises repeatability and increases the likelihood of human error.

From a procedural standpoint, the configuration of the test over the company's battery module must be performed after other system setups, adding complexity to the workflow. Last-minute changes were required due to profile format mismatches. Rather than adapting the profile to the application, the application itself was modified to accept the profile directly, enabling the operator to load a plain `.txt` file without manual editing. While this solution prioritizes simplicity for the operator, it highlights the need for more flexible and robust profile handling mechanisms.

The final test report currently lacks graphical representations, requiring operators to export raw data to external software environments such as MATLAB and develop custom scripts for visualization. This additional step introduces unnecessary overhead and could be mitigated by integrating native plotting capabilities directly into the system.

Lastly, the physical layout of the laboratory presents logistical challenges. The dispersion of machines across the workspace results in extensive cabling, which complicates maintenance and poses safety concerns. A more standardized configuration—with pre-planned equipment placement and optimized cable lengths—would improve both organization and safety. Additionally, the control board includes a large number of ports, many of which are unused and unlabeled. This lack of labeling contributes to confusion and increases the risk of misconnection. Improved port management and clear labeling are essential to enhance system reliability and ease of use.

5.2 Analysis of Results

This section presents a detailed analysis of the results obtained during the experimental phase, comparing actual system behavior with expected performance benchmarks. The evaluation is divided into two parts: partial results from individual subsystems and integrated results from test executions over the battery module.

5.2.1 Subsystem-Level Results

The power cyclers demonstrated flawless operation, consistently delivering the expected voltage, power or current profiles without deviation. Timed communication protocols with both the chiller and the climatic chamber were successfully validated in standalone mode, i.e., without the DUT connected. Although initial integration with the chiller presented challenges, the current configuration operates reliably and meets all thermal and flow control requirements.

The climatic chamber performed exceptionally well, offering a wide range of configuration options. However, several advanced features—such as recipe management—were not utilized during this phase and should be considered for future development.

The control board successfully acquired data from NTC temperature sensors and handled contactor feedback collection as expected. The sensor readings remained stable and consistent throughout the tests, confirming the reliability of the board's data acquisition and actuation functionalities.

5.2.2 Battery Module Test Results

Full-system tests conducted under simulated urban and suburban operating conditions yielded highly satisfactory outcomes. The test sequences ran smoothly, with minimal intervention required from the operator. The system's automation allowed the operator to focus exclusively on monitoring the DUT's behavior, rather than managing peripheral equipment or manual adjustments. This aspect is particularly important given that each test can last 45 minutes and multiple consecutive runs are often required.

The only significant stress point was observed during the initial and final phases of the test, where manual supervision remained necessary. However, in the final phase, this was substantially alleviated by the profile's capability to pre-load parameters that transition the system into an optimal idle state—also suitable for the safe opening of the contactors.

Overall, the control system demonstrated high reliability, operational coherence, and user-centric design, validating the core objectives of the thesis project.

Chapter 6

Conclusions and Future Works

The primary objective of this thesis was to simplify the workflow for test operators engaged in functional testing of newly developed battery modules or packs. Given the high number of tests typically required—each ranging from 15 minutes to several hours—it was essential to minimize operator workload and stress. From this perspective, the results achieved are highly satisfactory: the system enables the operator to focus exclusively on real-time data visualization, without the need to manually manage data transmission and timing, monitor the correct operation of auxiliary equipment, or intervene in the event of uncontrolled and potentially harmful interruptions.

A critical evaluation of the system reveals that while the test execution phase is robust and well-optimized—with accurate timing for profile transmission, correct instrumentation response, and appropriate safety mechanisms such as warnings and alarm interrupts—the initial setup phase, particularly the closure of the contactors, remains an area for improvement. The final phase of the test is well-handled: the procedure concludes correctly, output parameters transition smoothly to an idle state, and all auxiliary systems are properly suspended. However, the preparation required for safely opening the contactors is still entirely manual, indicating a gap in automation.

Moreover, the system's adaptability to different types of DUTs warrants further consideration. Significant variations in DUT architecture may necessitate modifications in BMS communication protocols, the selection and prioritization of feedback signals, and the criteria for triggering warnings or alarm interrupts. Consequently, both the DUT configuration interface and the diagnostic window should be designed with sufficient flexibility to accommodate different DUT specifications and operational requirements.

In light of the results obtained, several directions for future development can be identified to enhance both the functionality and usability of the system. First, the management of feedback signals could be refined. From a visualization standpoint, a more structured interface could be implemented, prioritizing the display of critical data within a unique primary window while relegating auxiliary information to a secondary view. Furthermore, the integration of automated procedures for generating graphical outputs and compiling a final, well-formatted `PDF` report summarizing key findings would streamline documentation and improve overall efficiency. On the hardware side, the adoption of more ergonomic and modular connectors would facilitate rapid and reliable cable connections and disconnections, particularly in these scenarios requiring frequent reconfiguration; this would address the limitations due to Molex connectors, which tend to become increasingly difficult to detach as more wires are crimped. Finally, the system architecture could be adapted to accommodate different DUTs types, such as power inverters. A relevant example is a dual high-voltage inverter test case, in which two DUTs operate simultaneously—one dedicated to speed control and the other to torque control—both interfacing with a motor located outside the test cage. Such an extension would broaden the applicability of the platform and support more complex testing scenarios.

In conclusion, this thesis project has successfully led to the development of a graphical interface and a test management system that are both intuitive and practical for the test operator. This enables the user to focus primarily on the DUT, while delegating the control of auxiliary equipment and boundary conditions to the automated system, thereby streamlining the testing workflow and enhancing operational efficiency.

Bibliography

[1] Adnan H Anbuky Phillip E Pascoe. «Automated battery test system». In: (2003) (cit. on p. 3).

[2] Wei Yao Yi Peng Wan. «Study on Mechanical Automation with Improvement of Battery Test System for Electric Vehicle». In: (2013) (cit. on p. 3).

[3] Emil Petkovski Silvia Colnago Marco Faifer and Luigi Piegari. «Automated test equipment for battery characterization: a proposal». In: (2022) (cit. on p. 3).

[4] Shahab Khalghani. «Streamlining Electrical Battery Testing Through Automation : A method to automate the testing of automotive batteries employing LabVIEW, TestStand, and Ardiuno». In: (2025) (cit. on p. 3).

[5] FLAG-MS. *PSS Hardware v0*. 2020 (cit. on p. 8).

[6] FLAG-MS. *PSS Manager r00 final (pdf)*. 2020 (cit. on p. 8).

[7] Texas Instrumentation. *TMS570LC43x Technical Reference Manual*. 2018 (cit. on p. 8).

[8] RIGOL. *DP800 Series: Programmable Linear DC Power Supply*. 2024 (cit. on p. 11).

[9] MSL. *Manuale 230033 EC300CA 23061 Ita R0*. 2023 (cit. on p. 13).

[10] LNEYA. *KRY-55-2T User manual*. 2024 (cit. on p. 13).

[11] EA Elektro-Automatik. *EA-PSB 9000 3U - Datasheet*. 2018 (cit. on p. 15).

[12] FACET. *0730 Facet Temperature Sender*. 2024 (cit. on p. 17).

[13] Sensata Technologies. *116CP/126CP Series Pressure Sensor*. 2020 (cit. on p. 17).

[14] TE Connectivity. *KILOVAC High Voltage DC Contactors*. 2025 (cit. on p. 17).

[15] FLAG-MS. *Software Coding Guidelines v1.1*. 2022 (cit. on p. 25).