POLITECNICO DI TORINO - GRENOBLE INP-ENSE3

Master of Science in Electrical Engineering

Master Thesis Dissertation

LIFE Test Bench Design

Supervisors
Prof. Aldo Canova
Prof. Pierre Lefranc
Company supervisors:
Dr. Alessandro Fasolo
Dr. Xinchang Liu

Candidate
Eric Danese

September 2024

Summary

The LIFE time test bench project aim, is to advance ETEL S.A. equipment for testing the lifespan of insulating materials in torque motors. This project involved the design of an advanced test bench that can perform various motor tests, like AC (alternating current) dielectric, surge, and high-voltage PWM (pulse-width modulation) tests, while also controlling various aspects such as temperature and humidity. The test bench is required to automate the testing process and manage each motor coil independently to address the destructive nature of the tests.

The design takes into account both electrical and mechanical components, and uses a modular approach with Eurocard subracks to house the electronic components, which will facilitate possible future expansions. The system's architecture needs to be able to handle both high-current and high-voltage operations, therefore presenting challenges in circuit design, component selection, and safety measures. Even more, the project includes the development of firmware for microcontroller integration and software for a PC graphical interface which will control the entire testing process.

A multidisciplinary approach was necessary due to the project's complexity, integrating software development, mechanical hardware, and electrical hardware to create an advanced and dependable test bench.

Acknowledgements

This project has been both challenging and fulfilling, and I have truly enjoyed my time working on it. I would like to express my sincere gratitude to everyone who contributed to its success.

Firstly, I would like to thank my academic supervisors, Aldo Canova and Pierre Lefranc, for their guidance. I am also deeply grateful to my company tutors, Alessandro Fasolo and Xinchang Liu, whose support and mentorship made this project possible.

A special thank you goes to my colleagues and friends, who made every day enjoyable and full of new experiences. Lastly, but by no means least, I would like to express my deepest gratitude to my family. Their unconditional love and constant support over the years have been the foundation of all my achievements.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

LIFE time test bench is a project whose intent is to increment the existing equipment for testing the lifetime of insulating materials inside the torque motors produced by ETEL S.A. [1]

1.1 ETEL

ETEL S.A. is a Swiss Company established in 1974 as a spin-off from the Swiss Federal Institute of Technology of Lausanne. The Headquarter is located close to the city of Neuchâtel. Renowned for pioneering advancements in the high technology field, ETEL has established itself as a leader in the field of electrical motors and precision motion platforms, which are used in applications such as semiconductors, machine tools, and optical inspection systems. The company is known for the quality and reliability of its products, which are designed to deliver high performance and extreme precision.

1.2 Project overview

The new test bench required several key implementations. Firstly, it needed the capability to test motor dielectrics at their operating temperatures. Additionally, it was essential to control the air humidity within the motor testing enclosure. Automation was another critical requirement of the new system, which necessitated individual control of each motor coil due to the destructive nature of the tests. These aspects are illustrated in Figure 1.1, which provides an overview of the system.

The test bench must include several features, it should perform tests such as motor AC (alternating current) dielectric testing, motor heating, motor surge testing, and high voltage PWM (pulse-width modulation) signal testing (see Appendix A for more details). Moreover, it must manage cycle control (switching between different sources), humidity control, and motor temperature. At last, everything needs to be operated while monitoring and recording the ambient temperature, humidity, input power, and test process.

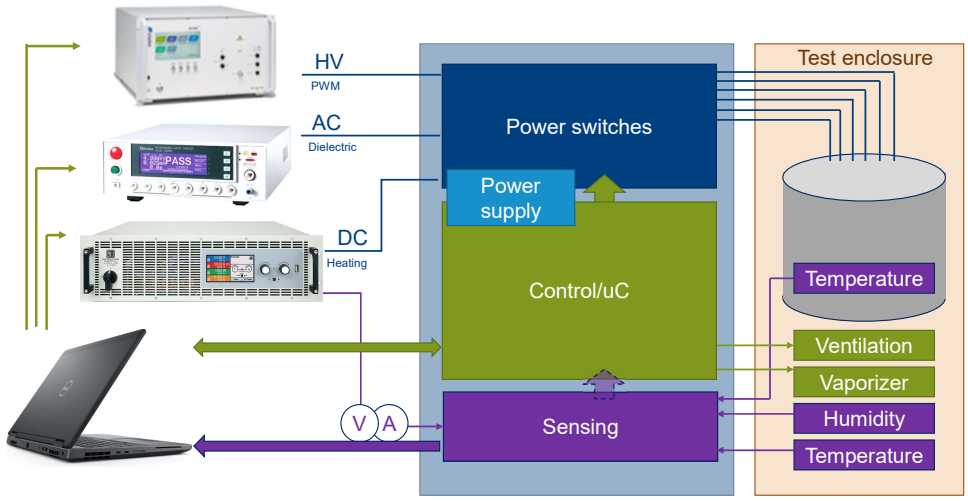


Figure 1.1: Test bench overview

Going a little deeper into the subject of motor tests, it can be observed that the AC dielectric test (or aging test), surge test, and the PWM signal all present an high voltage aspect. Whereas, the motor heating, achieved thanks to the losses in the coils as if the motor was running, will require to inject an high current through the coil. This introduces the challenge of managing both high currents and high voltages.

The goal of this internship is to work on the electrical and electronic aspects of the project (represented by the light blue box in Fig. 1.1) and the control systems, which include both developing the software on the PC (personal computer) and the firmware for the microcontroller (represented by the green box).

The fundamental requirement is to be able to design and construct a single subrack and establish its control system, which includes aspect like circuit and PCB (printed circuit board) design, mechanical aspects, and firmware programming. Even more, it involves establishing the initial steps of the software development, aspect which can be found in Figure 1.3. This will serve as a prototype to demonstrate the feasibility and functionality of the overall design before scaling up to additional subracks.

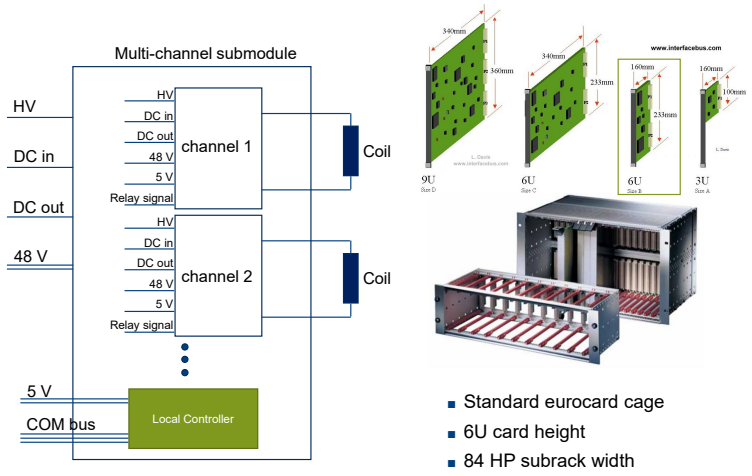


Figure 1.2: Modular design for analog channels

As for the electrical components, the objective is to build an electronic enclosure (rack) which contains some electronics boards (modules) for coil control (Figure 1.2 illustrates the concept, not the finalized design). This design will utilise the standard Eurocard format to create multiple subracks with standard-sized cards, providing a modular and easily expandable solution.

1.3 Roadmap

The project's main aspects have been outlined. The roadmap for the next six months is divided into several key areas:

- Electrical hardware
- Mechanical hardware
- Firmware
- Software

As for electrical hardware it is understood anything that relates to the key components for the control of the channel, the switches and therefore their driver circuits, as well as the printed boards. Mechanical hardware covers the electronic enclosure, including the rack and subracks where the electrical components will be mounted. The motor enclosure, instead, is not included in the intents of the internship. Firmware refers to the code that

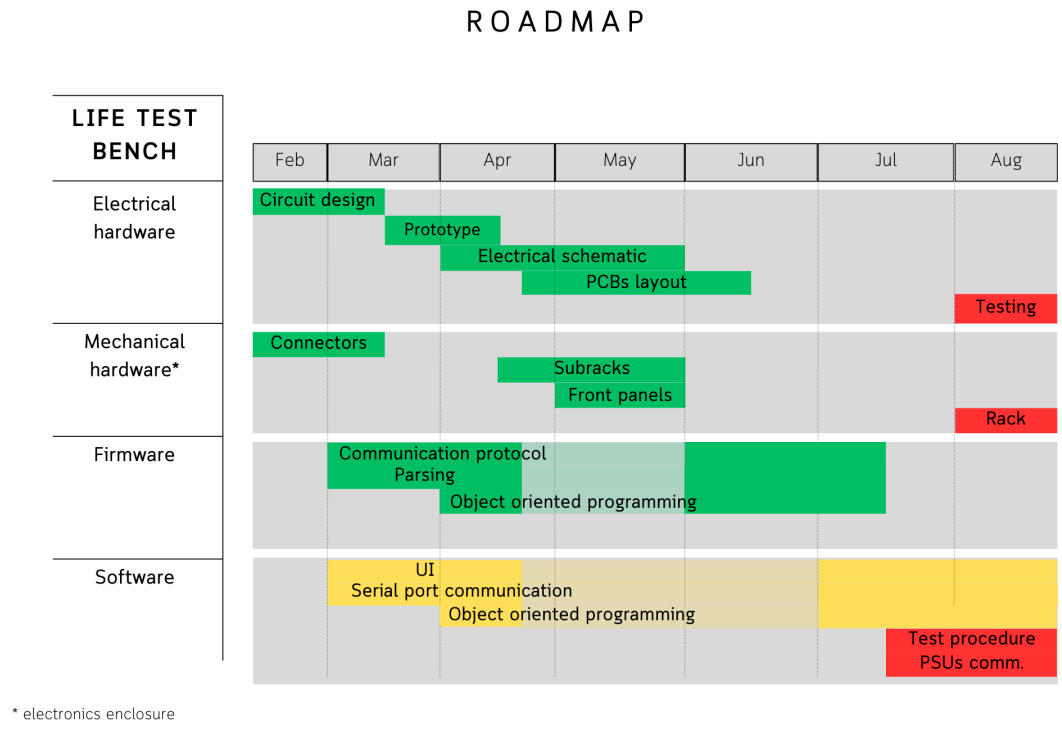


Figure 1.3: Roadmap

will be implemented in each "motherboard" to control the relays. Whereas, the software is the code developed on the PC to manage the entire process.

Figure 1.3 presents the roadmap of how the work was divided and conducted during the internship. It is evident that most activities overlapped at some point, therefore requiring to work on multiple tasks at the same time.

Initially, the project focused on electrical and mechanical studies, but it soon became necessary to start laying the groundwork for the firmware and software. Once again, the electrical and mechanical aspects required significant attention, but eventually, firmware and software became the main objectives to achieve the first tests.

In the following pages, the project will be described by subject rather than the chronological order of events. However, it should be noted that many aspects were closely interconnected and addressed simultaneously.

Chapter 2

Electrical hardware

The first objective was to design the electrical circuit for controlling each coil. This involved selecting all necessary components, based on the required specifications.

2.1 Coils data

The first step to approach is the study of the coils data for the motors to be tested by the new equipment.

While the voltage for the aging test can be predetermined to gather meaningful data on dielectric behavior, it does not depend on the specific motor being tested. In contrast, the heating process requires an high current, which means each motor will have a distinct amount of current due to their different winding configuration.

Starting from the datasheets of the motors, it shall be obtained the current that needs to flow in each coil to reach the working temperature steady state.

Figure 2.1 shows the current requirements for different lengths and winding configurations of the largest motor meant to be tested, with the final two values representing the smallest motor in the series. This illustrates the current needs for the high-current side of the channels.

Another important peace of information which can be retrieved is the power supply's required capacity. With the current values known, it is possible to estimate the necessary voltage in case the coils are all connected in series, as shown in Figure 2.2a. This time motor 1 clearly represents the highest constraint. All the data collected here will be essential when selecting the PSU (Power Supply Unit).

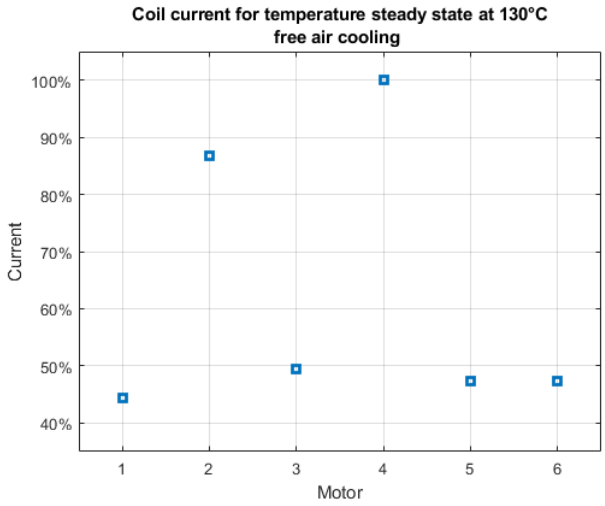
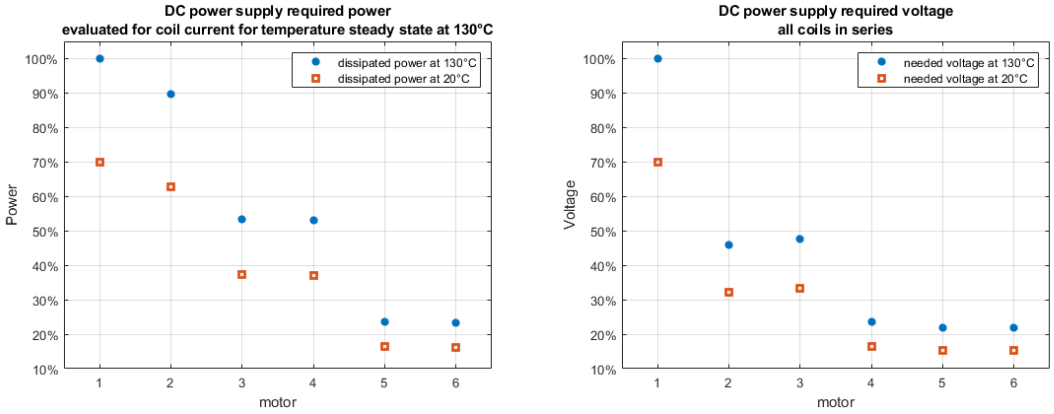


Figure 2.1: Coil current for temperature steady state



(a) DC PSU voltage

(b) DC PSU power

Figure 2.2: DC PSU data

2.2 Circuit design

At this point, it is fundamental to establish the configuration to control each coil individually.

Figure 2.3 outlines the initial concept, with a notable adjustment from the original design: each channel requires a high-voltage output for the surge test, which involves a fast-rising current to detect turn-to-turn faults.

The primary components of interest are the relays. One High Voltage (HV) relay is needed to establish the connection with the high voltage power supplies (more than one power supply is required to be able to perform all the different tests, switching from one to another will be a challenge to take into account). Additionally, an High Current (HC)

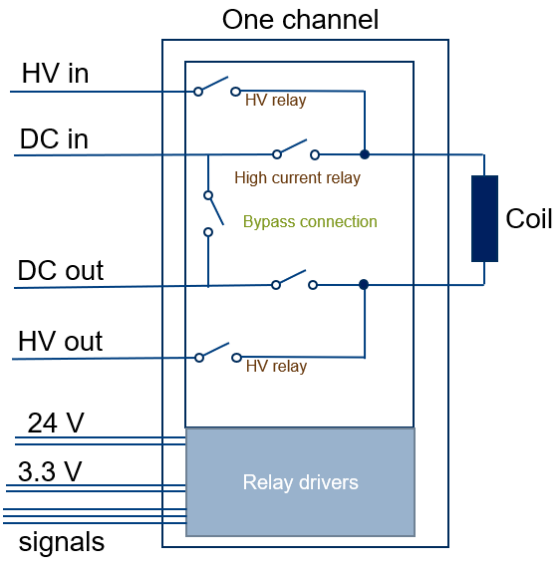


Figure 2.3: Channel configuration

relay for providing the elevated current level for the heating process. This component needs to be able to withstand the high voltage applied to the coil and it has to create a bypass connection in case of a faulty coil (hypothesis of connecting all coils in series during heating). A second high-voltage relay is needed for the low side of the coil.

The other part that needs to be mentioned is the relay driver, which will be discussed more in detail afterwards (section 2.2.2).

2.2.1 Relays choice

For what concerns the HV relays the choice fell back into a reed relay. Whereas, the main challenges were represented by the HC relay, which, as mentioned, needs to withstand an high voltage between its open contacts while being able to carry high current.

The configuration shown in figure 2.3 is not the only possible solution for achieving the bypass connection, multiple can be found and some are shown in Figure 2.4.

This aspects allowed for a broader relay selection, which meant being able to choose the configuration only once the best relay solution was found. After a long search for the component with a sufficiently high voltage withstand between open contacts, configuration number 4 (Figure 2.4) was chosen. Indeed, a DPST (Double Pole Single Throw) relay with a "wide contact gap" was found, essential for this application.

Power electronics solutions were investigated, however they required a high component count therefore producing significant losses, which made them unsuitable.

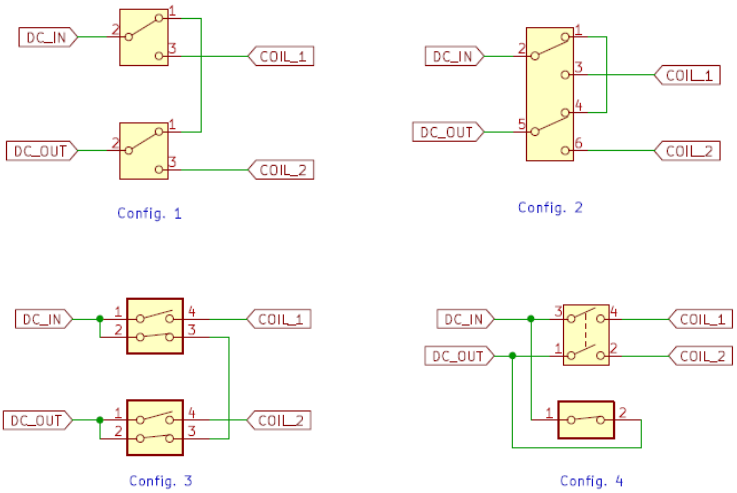


Figure 2.4: HC Relay configurations

2.2.2 Relay driver

As the relays have been chosen, the circuit to control them needs to be designed.

The relays were chosen with a coil working voltage of 24 VDC. Which meant finding a way to control them from a 3.3V pin of a microcontroller (μC). The easiest solution found is depicted in figure 2.5. A MOSFET (Metal-Oxide-Semiconductor Field-Effect Transistor) drives the coil side, whereas an optocoupler is used to achieve both insulation and control of the MOSFET.

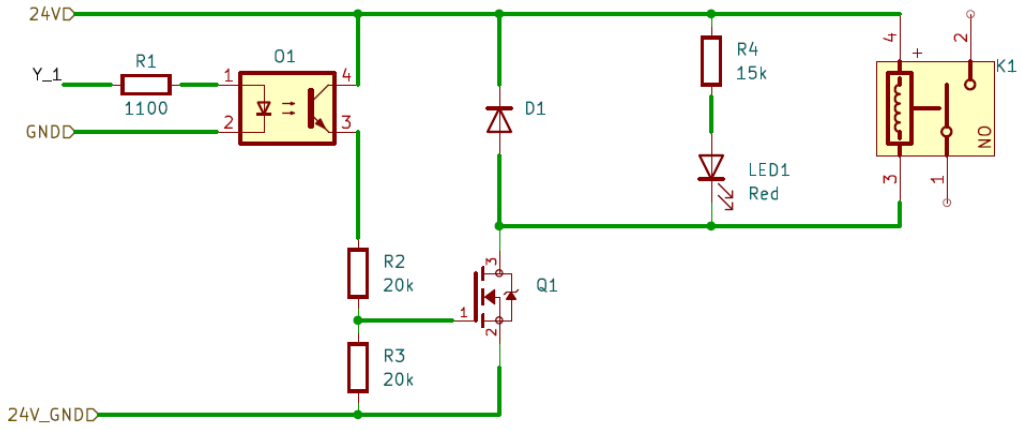


Figure 2.5: Relay driver circuit

The voltage divider is needed due to the MOSFET gate not being able to handle voltages above 20V. Although high resistances slow the switching speed, fast switching is not crucial for this application. A diode is included to suppress the relay coil surge, a voltage spike which occurs after de-energizing the coil. An LED (Light Emitting Diodes) is used to indicate the correct operation of the relay.

The resistances R1 and R4 were evaluated to obtain the desired value of current, in accordance to the following equation:

$$R > \frac{V_{in} - V_{drop}}{i_{max}} \tag{2.1}$$

The selection of the R1 value had to consider both the μC current limitation and the operating range of the optocoupler.

2.2.3 Safety concerns

Since high voltages are operated in the test bench, safety concerns obviously come to mind. Even though software-based safety measures will be implemented, a physical safety layer was added. The way the safety will be implemented, is by ensuring the HV relay and the HC relay cannot operate at the same time. This is achieved thanks to some logic gates, as described in Figure 2.6. In particular a NOT and an AND, whose truth table looks like Table 2.1, where V2 is the signal meant to control the output. As it can be seen, V1 being active avoids the relay to operate.

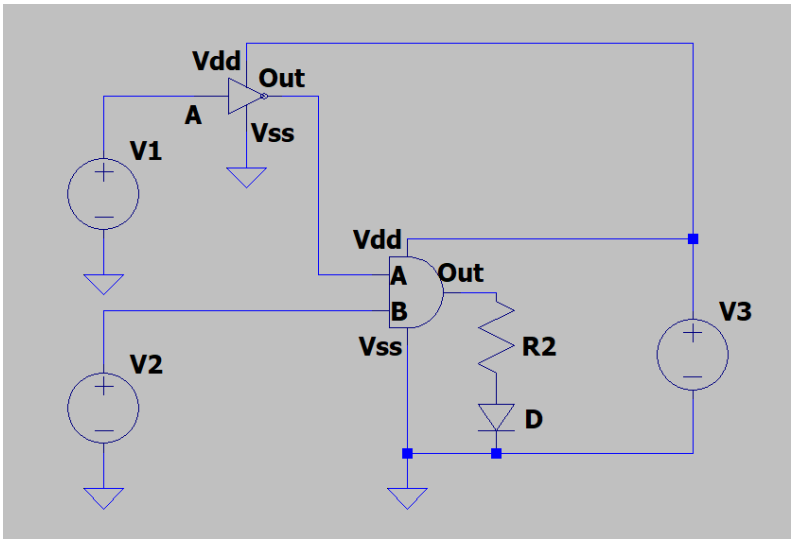


Figure 2.6: Logic gates circuit [2]

V1	V2	Out
0	0	0
0	1	1
1	0	0
1	1	0

Table 2.1: Truth table

2.3 First prototype

This prototype worked as a feasibility study for the project. Indeed, after a long research for the right relays, all the components needed to build a prototype were purchased and once they arrived, it was critical to focus on assembling the early design and prove that it was functional. The components were then soldered on some prototyping boards accordingly to the designed circuit (Fig. 2.7) and powered from a bench top power supply.

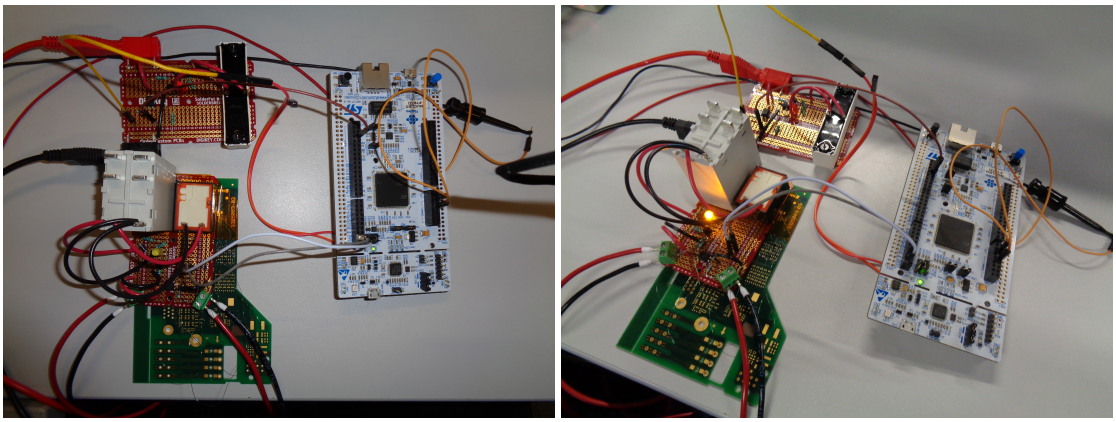


Figure 2.7: First prototype

Soldering the logic gates, which were the only surface-mount devices, was challenging. The solution involved using boards with the correct footprint and connecting them with wires to the proto-board.

All the connections established, it was time to test the operability. Controlling the pins of an STM (STMicroelectronics NV [3]) board, the design proved to be successful.

This first example was also important to establish whether some necessary features were left out during the design. For example, it felt clear there needed to be an indicator of the active/non-active state of the channel, which was implemented by adding a green LED that will be directly controlled by one pin of the μcontroller.

Another outcome this prototype revealed was the inaccuracy in the safety measure implemented by the logic gates, as the relays present some delay between the command

and the actual movement of the contacts. To avoid the superimposition of the relays, a delay circuit is implemented using a simple RC circuit (Fig. 2.8) that ensures the voltage remains above the threshold level of the logic component for the required time.

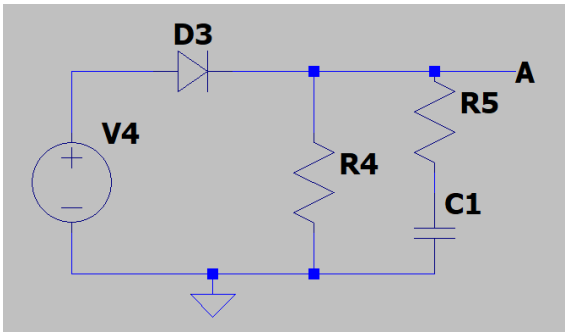


Figure 2.8: Signal delay circuit [2]

In the end, the last study made on the relays was to verify their heat dissipation. Indeed, this could prove to be an issue inside the rack due to stacking multiple cards and little ventilation.

The maximum ambient temperature was retrieved from the datasheets, the most pressing one being the HV relay with 70°C.

At the same time the temperature increase of the relays was checked thanks to a thermal camera. In this case it was seen how the bypass relay heats up the most (Fig. 2.9), due to its higher dissipation and its small package.

Since the relays of one channel will not be operated all at the same it probably will not represent an issue, otherwise forced ventilation will need to be added to the rack.

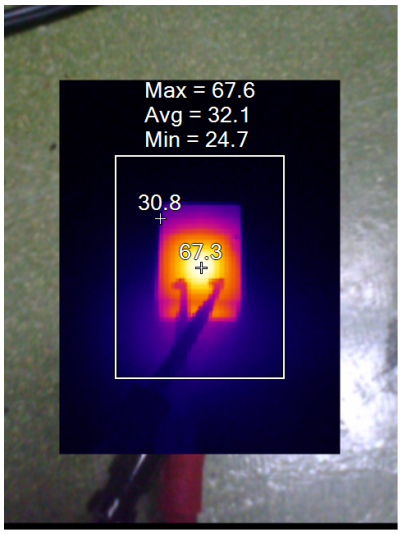


Figure 2.9: Bypass relay temperature increase

2.4 Electrical hardware overview

With a detailed study of the components for controlling the channels completed, a broader overview is needed before delving further into the electrical hardware.

Firstly, how many channels can be fit into a single card needs to be established, as well as, how many cards per subrack. These considerations, of course, come from the mechanical side and will therefore be discussed later (Chapter 3). These two parameters both influence the number of necessary pins, as it gives the number of channels and therefore relays to be controlled. Which means being able to choose if some μcontrollers have to be installed on each board or if they can be controlled by only one. Indeed, the STM32 Nucleo-144 was chosen to establish an interface between the PC and the "motherboard" of each subrack. This board provides an elevated number of pins, that may be sufficient for the application.

The mentioned motherboard is the subrack backplane, where all the routing will happen (Fig. 2.10).

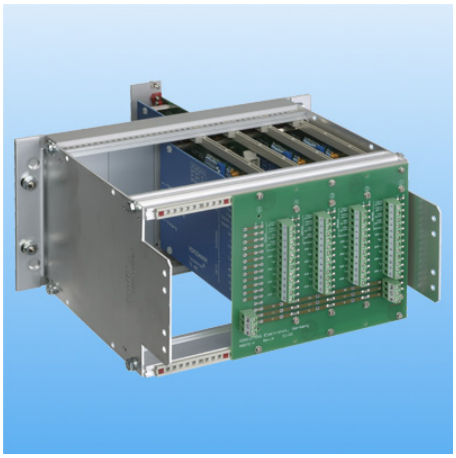


Figure 2.10: Subrack with backplane [4]

Another consideration is the need to switch between different high-voltage power sources for various tests. Therefore new HV relays needs to be placed; the original idea was to place them in the backplane, but space constraints need to be addressed.

As well, the NUCLEO board needs to be connected to the motherboard, either for communication with the μCs on the child-boards or routing its pins.

The solution evolved through trial and error, integrating electrical and mechanical aspects into a final design, to devote one of the child boards to manage the inputs and to fix the NUCLEO board.

The primary considerations now shift to mechanical aspects, a little deviation from the electrical hardware is therefore required and it will be discussed again later.

Chapter 3

Mechanical hardware

As previously mentioned, the mechanical aspects relate solely to the electronic enclosure. The subracks, which are the submodules housed within the rack, will constitute the final product. Given the number of channels to be controlled, multiple modules will likely be necessary.

At the very beginning, the point in question was the choice of the connectors. Both on the backplane and on the front panel. As an example, the choice for the front was to use banana connectors, as they felt like the easiest solution for interfacing with the motor enclosure. Whereas, for the child board connection to the backplane motherboard, the biggest constraint was ensuring simple connections without alignment issues, while at the same time guarantee the dielectric strength constraint due to the high voltage application. The choice fell back on standard DIN41612 (IEC 60603-2 [5]) connectors.

Once this little details have been chosen, the most important concerns regard the available space inside the subrack. Firstly, the module has a standard width of 19 inches (or 84TE, 1TE = 5.08 mm), which is important for determining the number of child boards that can be installed, while the height of 6U (standard Eurocard size) and a length of 160 mm are chosen. In order to establish how many channels can be fit into a child board, a study taking into account the components footprints needs to be done.

For all the aspects concerning dimensions and positions, the IEEE Std 1101.1 [6] was used.

3.1 Child board

The dimensions of the boards being fixed to 233.35 x 160 x 1.6 mm, it is now possible to evaluate if there is enough space for the components. A brief study of the footprints shows the most reasonable solution is two channels per child board.

On the other hand, the components height stands as another mechanical issue. Particularly the HC relay correctly positioned on the PCB would reduce excessively the maximum number of child boards inside one subrack, therefore increasing the number of

subracks needed to an unfeasible amount.

This issue needed to be addressed promptly to prevent delays in the electrical design caused by components not fitting within the mechanical layout.

The chosen solution was to position the relay on its tall side and connect it by soldering wires from its pins to through-holes on the PCB. To secure the relay in place, assistance was sought from a designer colleague, who developed a locking component that clips onto the board, as shown in Figure 3.1. This piece was subsequently 3D printed and tested, proving to be functional, and will therefore be incorporated into the final PCB design.

This new configuration helps keep the height of the child board under 7TE.

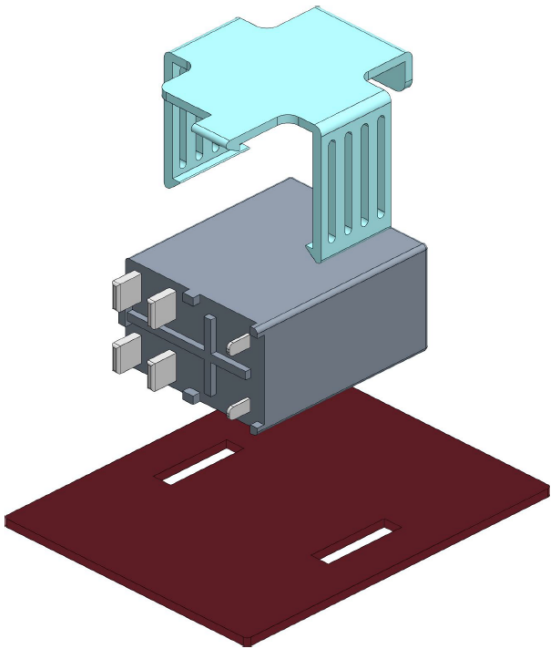


Figure 3.1: Locking part for high current relay

3.2 Motherboard

Once again, certain components need to be mounted on the board, so space must be allocated accordingly.

In this case, the subrack design plays a crucial role, as the plan is to include a backplane board. The enclosure requires appropriate rear horizontal rails for backplane mounting, resulting in a board size of 426.42 x 262.05 mm, which also allows for the precise positioning of the connectors.

At this point, it was clear that no space could be found to annex the NUCLEO board, so the initial idea was to stack the PCBs on the backplane, printing a PCB to route the pins of the STM board to a rectangular connector.

However, the relays needed to switch between different high-voltage power sources, could not be placed on the motherboard either, which meant that the stacked boards would also require power connectors. At this point, it became evident that this solution was not practical, and additional electrical considerations arose due to the high voltages involved.

As the concept required a new printed board anyway, it was chosen to dedicate one of the child boards (intended as the cards inserted in the subrack) to work as "inputs manager". This particular first child board, will present the routing for the NUCLEO board pins and the power inputs with all the needed relays.

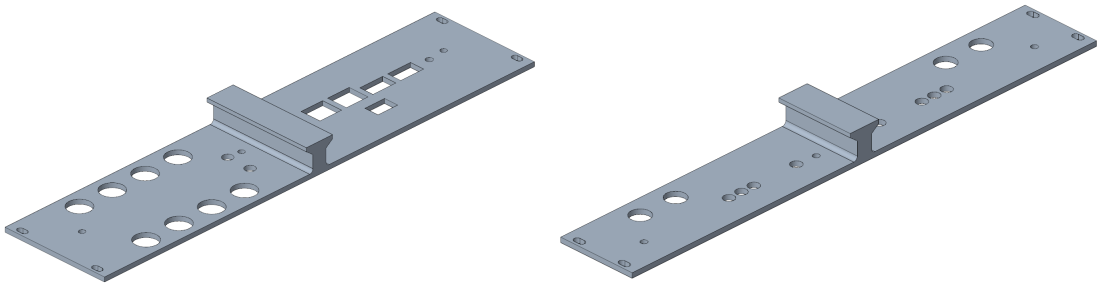
3.3 Front panels

Another factor to consider is the definition and design of the front panels. Indeed, as mentioned, the board could fit within 7TE; however, the front panel has restrictions imposed by Std 1101.1, which govern not only the width of the panel but also the relative position of the board. This is a significant concern because the front panel must be securely screwed in place to ensure a perfect fit for all the boards, which has brought to a final choice of a 8TE front panel for the child board.

For the first child board, which is dedicated to the NUCLEO and inputs, it was evident that a larger front panel was necessary due to the high number of banana connectors that needed to be mounted. This led to the conclusion that the optimal design would consist of nine 8TE child boards, plus one 12TE board.

Once the positions of the holes for mounting the front panel to the rack were determined, the remaining openings were aligned with the PCB layout. This includes the holes for securing the board to the front panel, which will be done using L-shaped mounting brackets.

The final products are displayed in Figure 3.2.



(a) First child board (b) Child board

Figure 3.2: Front panels

The banana connectors are positioned as far from the board as possible to maximize the available space for wiring connections. This is especially important for high-current circuits, where the thickness of the wires means that bending them can be a significant concern.

These front panels were 3D printed, similarly to the clip. Once it was confirmed that their flexibility was adequate, the decision was made to print additional panels rather than fabricate them from aluminum, at least during the initial stages of building the subracks.

Chapter 4

PCBs

Having addressed the mechanical aspects, the number of channels per child board was determined, along with the need for a dedicated child board for inputs and the microcontroller. Additionally, the total number of child boards and their dimensions were established. All of these factors will be crucial for designing and building the PCBs.

One aspect that still needed to be determined was whether the pins on the NUCLEO board are sufficient to control everything, which can now be assessed thanks to the known number of child boards. Each channel requires 4 pins, 3 for relay control and 1 for an LED indicating the channel's activation state, totaling 72 pins, which is within the limit of the NUCLEO board, assuming one NUCLEO board is used for each subrack.

This means there is no need of a μC on each child board, which is a significant change compared to what described in Figure 1.2. Indeed, the achieved design requires only one controller per each subrack.

4.1 Electrical schematic

The first step is to create the electrical schematics, with three different types of boards each requiring its own schematic.

The electrical schematic will be drawn in an EDA (Electronic Design Automation) environment, particularly KiCad [7] was chosen.

This work involves integrating all considerations from the circuit design, placing all components into the schematic, from relays to connectors, and including additional elements that may be useful in the circuit, such as capacitors, test points, and jumpers.

Additionally, all pins must be connected to their respective nets, with each net appropriately named, ensuring that the correct elements are connected together while avoiding improper connections.

One representation of schematic, the inputs board, is shown here in Figure 4.1, the others can be found in the Appendix B.

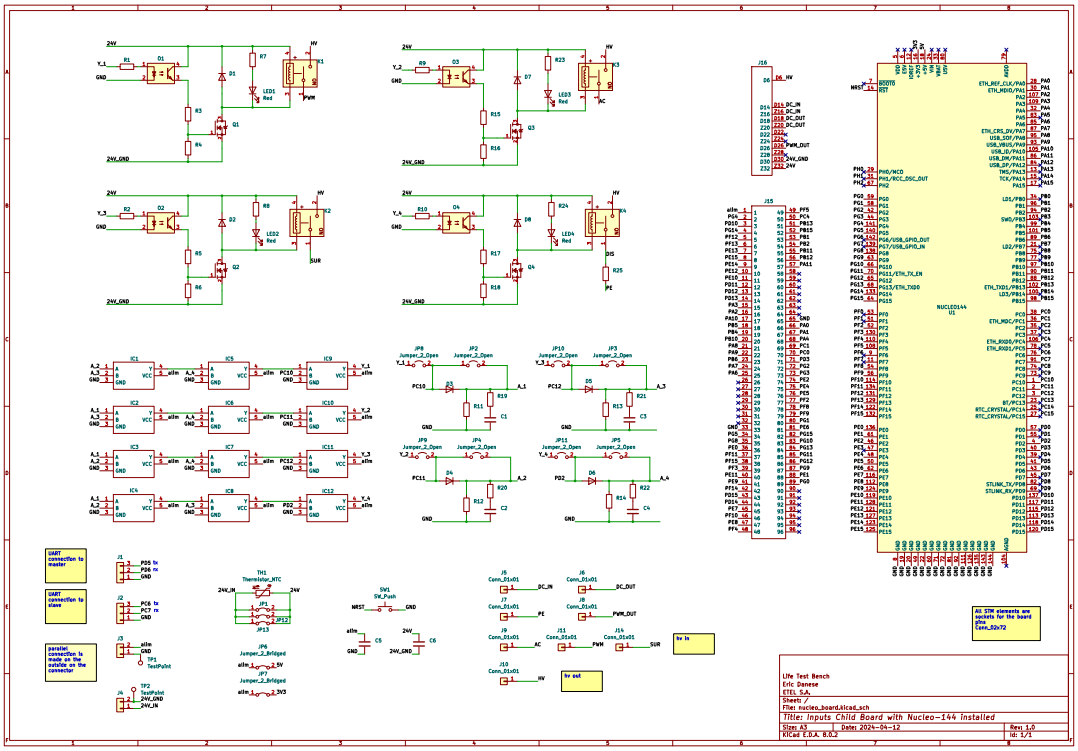


Figure 4.1: Inputs board electrical schematic

4.2 PCB layout

At first, the idea was to find a service provider of PCBs layout and give this job to them. Yet, this work proved to be extremely intertwined with the mechanical aspects, in a way that lots of it would have had to be done anyway before sending the data to the supplier. Consequently, it was chosen to be taken care of personally.

The most important concept of PCB layouts is understanding its layers. The main ones being:

- Insulation layer
- Copper layer
- Solder Mask
- Silkscreen

In a 2-layer PCB, which features 2 copper layers, this configuration is chosen for the project as more layers are unnecessary and would increase costs. The layers are duplicated for both the front and back sides of the board, resulting in the following stratification: back

silkscreen, back solder mask, back copper layer, insulating material, front copper layer, front solder mask, and front silkscreen.

Edge Cuts

This "layer" in KiCad refers, as the name says, to the mechanical aspects of the board, therefore all the cuts, holes, and millings that needs to be done to obtain the final outline of the board. It could be associated with the insulation layer as the FR-4 (a glass-reinforced epoxy) is the one being cut.

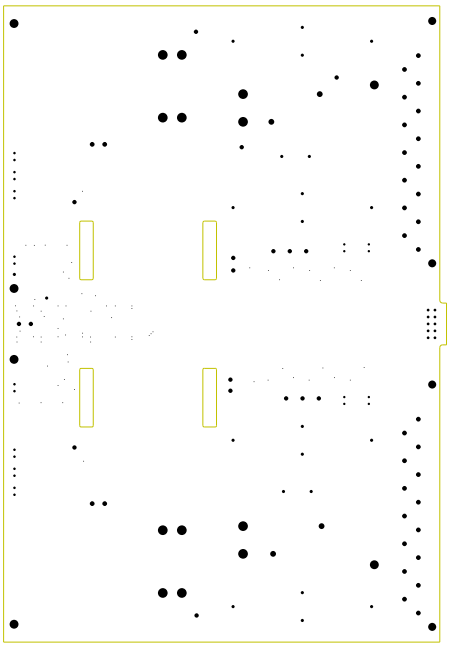


Figure 4.2: Edge cuts layer

Copper layer

The copper layer is chosen with a thickness of 70 μm (2 oz/sq ft), instead of the more common 35 μm, this way the high current traces will be more reasonably large. Indeed, this layer accounts for the evaluation of trace width, a track needs to be sufficiently large to avoid temperature increase issues. The very small thickness of the copper layer means very large traces to withstand high currents. Thankfully, in a 2 layers PCB, the copper layers are both external, which means there is no epoxy between the copper and air, therefore helping from the point of view of heat dissipation.

Another characteristic which needs to be taken into account is the voltage drop due to track resistance. This perspective influences more the high voltage traces, particularly

for the surge test. In fact, since the resistance of the motor coils is relatively small, the PCB tracks could be responsible for a significant voltage drop.

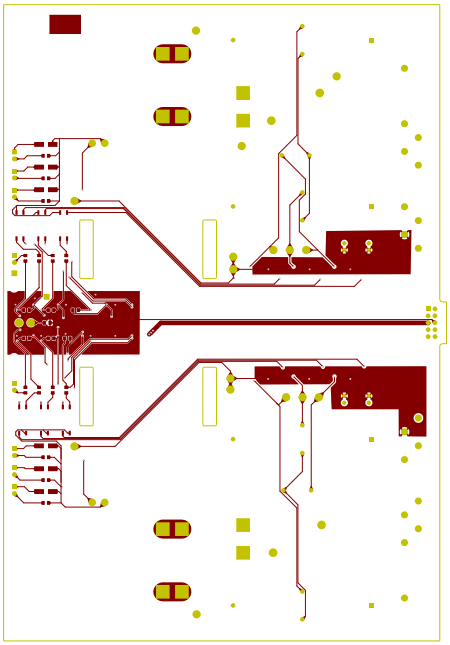


Figure 4.3: Copper layer

Solder Mask

The mask is a protective layer with many different objectives. First of all, provide insulation for the copper traces to prevent oxidation and protect the board from moisture and dust. Additionally, this helps avoid the components being soldered in the wrong places. Finally, it increases the breakdown voltage of the dielectric material. As it can be seen in Figure 4.4, this layer actually represent the openings in the mask, rather then where it needs to be.

Silkscreen

The silkscreen is a layer of ink, which is meant to facilitate the component recognition. As well, it can be used to write different aspect of the board which may be useful afterwards. Or even for some images, like the company logo or a little Easter egg (Fig. 4.5).

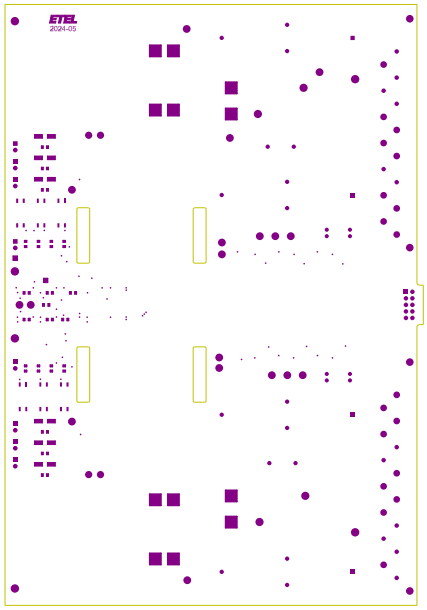


Figure 4.4: Solder Mask layer

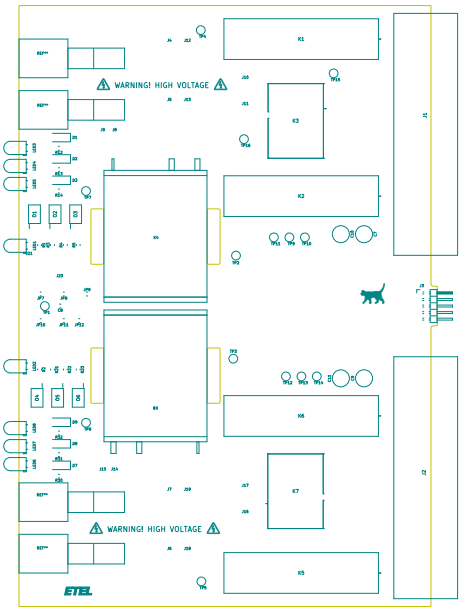


Figure 4.5: Silkscreen layer

4.2.1 Design

As it was explained how the PCB is made, it shall now be investigated what the PCB design is about.

The process to make the PCBs layout requires, at the very beginning, to search for or draw the exact footprints for each component. As an example, the logic gates were chosen to be the standard package SOT23-5, therefore already implemented in the EDA softwares like KiCad; whereas the high current relay, whose footprint was modified due to it being laid on the board, had to be manually drawn.

The footprints are a fundamental aspect of the PCB layout as a mistake in one of them can result in a component not fitting or in a pin which cannot be soldered correctly, this may even transform into a useless board.

It is very important when starting the design to put first all the fixed elements, like the outline of the board and the backplane connectors, fixed by the Std 1101.1.

Then, all the other components need to be placed in the available space. This process already needs to keep in mind the electrical connections and constraints.

Electrical constraints

As mentioned above one of the constraints is the trace width, with particular interest for the high current tracks. As well, the through holes for the wire connections had to be sized accordingly to the wire thickness.

Even more, actually the most challenging constraint due to the high voltages involved, is the observance of the insulation distances between the traces. The two dimensions, clearance and creepage, both need to be respected. The clearance being the distance between the conductors through air while creepage is the distance along the surface (Fig. 4.6). In order to do so, the international standard IEC 61800-5-1 [8] is followed.

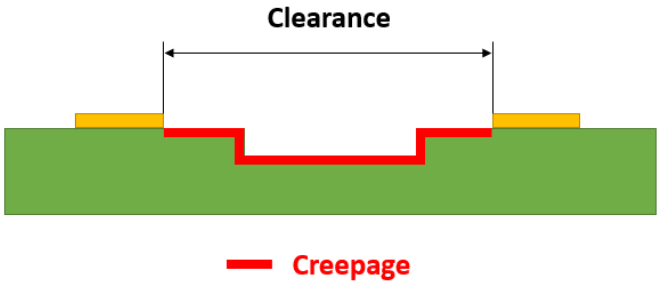


Figure 4.6: Clearance and creepage [9]

Moreover, it was chosen to keep the distance required for the high voltage conductors between the relay coil circuits (24V side) and the control signals from the NUCLEO board (3.3V side). This choice is not strictly necessary, yet it felt like it would grant better safety in case of one relay dielectric failing. Hence, the objective of the optocoupler in the relay driver circuit.

4.2.2 3D view

The EDA software KiCad offers a 3D viewer inside the PCB editor, which can be really helpful to understand the space taken by the components and have an idea of the final product. An example, the child board, is shown here (Figure 4.7), the other two boards can be found in the Appendix B.

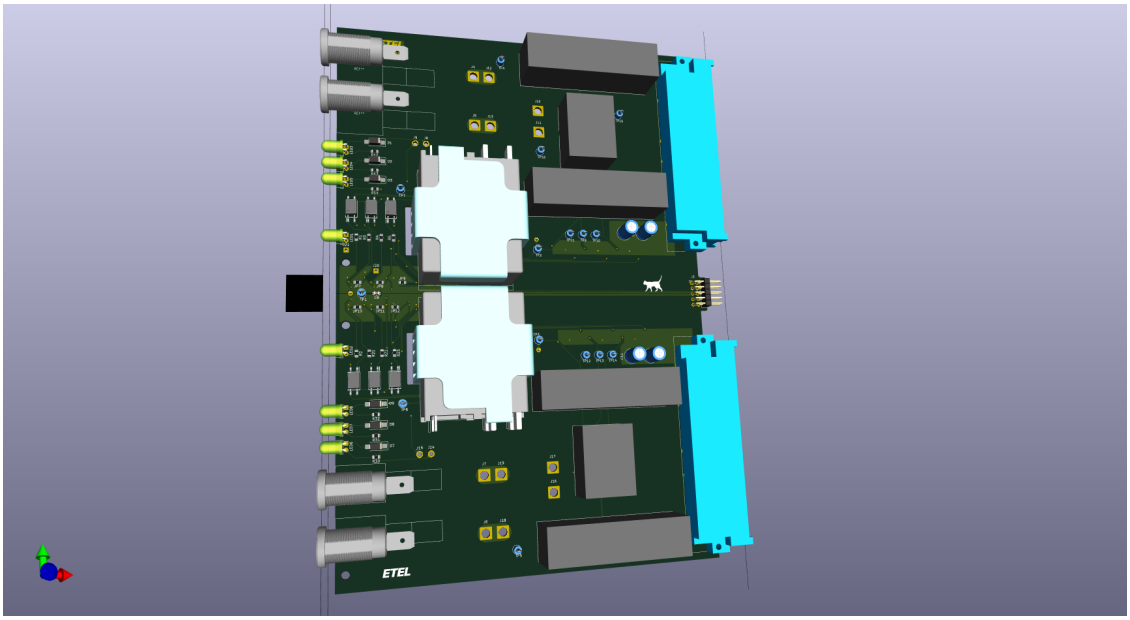


Figure 4.7: Child board 3d view

4.2.3 Printing

After completing the design, the boards were ordered from a PCB manufacturer. A small quantity of PCBs was selected, sufficient for the initial prototyping phase, allowing for adjustments in case any design flaws were identified.

Chapter 5

Softwares

Having analysed in detail the hardware components of the project, this report will now shift its focus on the software domain.

As previously mentioned, two distinct software parts are required: firmware for the NUCLEO boards and software for the PC.

5.1 Firmware

The firmware is responsible for controlling the relays by altering the state of the board's output pins, in accordance to the message received from its master connection.

It is important to note that the communication setup is in a cascade configuration, meaning each message must pass through the preceding board to reach the next one, as shown in Figure 5.1. Which requires the firmware not only to recognise the command, but also to identify the intended recipient of the message.

Figure 5.1: Communication configuration

5.1.1 Code

The firmware was developed using C language within the STM32CubeIDE [10] environment. The code must be identical across all boards, regardless of the motherboard index set during hardware setup, and this requires careful attention due to the different port used for board-to-board communication compared to board-to-PC communication.

Nevertheless, the main challenge was represented by developing an object-oriented program in C, a language which is not inherently designed for this approach. Thanks to the specific application and the small size of the project, it was possible to work around this issue without risking problems like using the same name for multiple functions. Even more, many capabilities given by an object-oriented program, like the need for private properties or the inheritance mechanism, were not required.

The IDE (integrated development environment) supported coding in both C and C++ language; even though C++ could have helped with the object-oriented coding, C was still the easiest choice. In particular, it did not create memory allocation issues; even more, it was the language of the libraries used in the program (HAL, hardware abstraction layer, library) and each of the examples used to understand the processes of GPIO (general purpose input-output) pins control and UART (universal asynchronous receiver / transmitter) communication.

The code is subdivided in different hierarchical layers (Fig. 5.2), starting from the highest, the main, to the lowest levels which include GPIO pins and UART communication. Whereas in the middle there can be found the objects. This subdivision is meant to simplify the code, in particular a low level should not see, or call, a higher level.

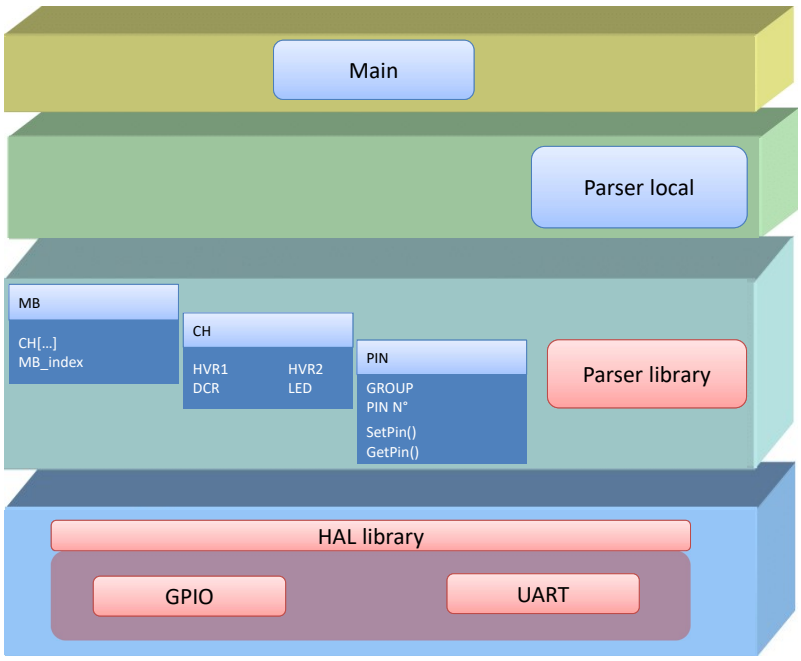


Figure 5.2: Firmware layers

A typical sequence of interaction between the different code parts can be seen in Figure 5.3. Receiving a command triggers the codes for reading the message, which is sent to the parser (message recognition), then the correct object function is called, and finally

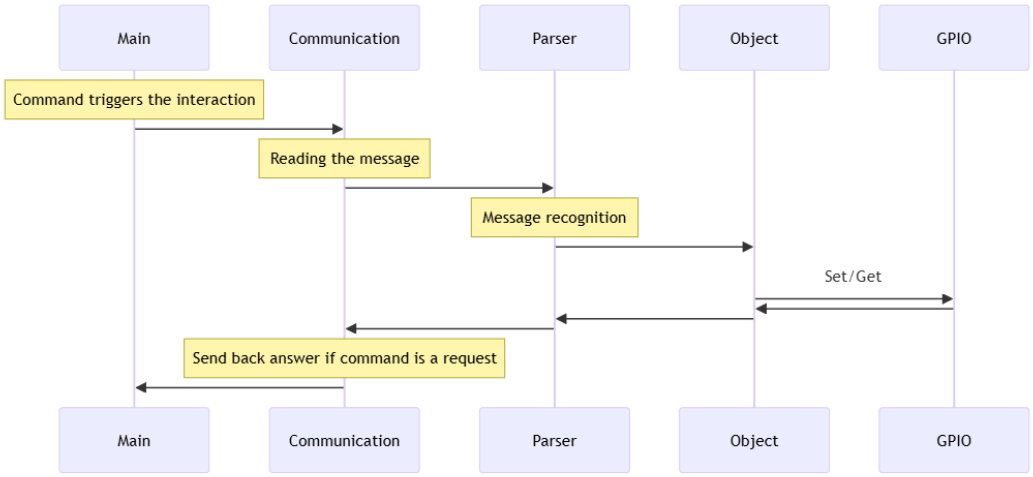


Figure 5.3: Firmware operation sequence

the code to set the pins state. At this point, an answer can be sent back if needed, rather than a command it was a request.

On the other hand, an example of the procedures which are followed when receiving a command is represented in Figure 5.4. In particular, it shows the flow for recognising if the board is the addressee of the board count request (a check that the board count command went well, during the hardware setup the board needs to know its position in the rack) and each possible action to be taken. In case the next board is not answering a timeout needs to break the process and reply no board was found.

5.2 Communication protocol

A slight pause from the coding part is now taken to describe a little bit the communication protocol between the PC and the boards.

Establishing the protocol from scratch was not an easy assignment, the solution was found by basing it on SCPI commands. The SCPI (Standard Commands for Programmable Instruments) commands syntax is very simple and can be easily implemented for this application.

The final design looks like:

MB:CONFIGuration 1,18,4

The first part states the object which needs to do the action and the second part is the action itself. Moreover, the first parameter always represents the component index.

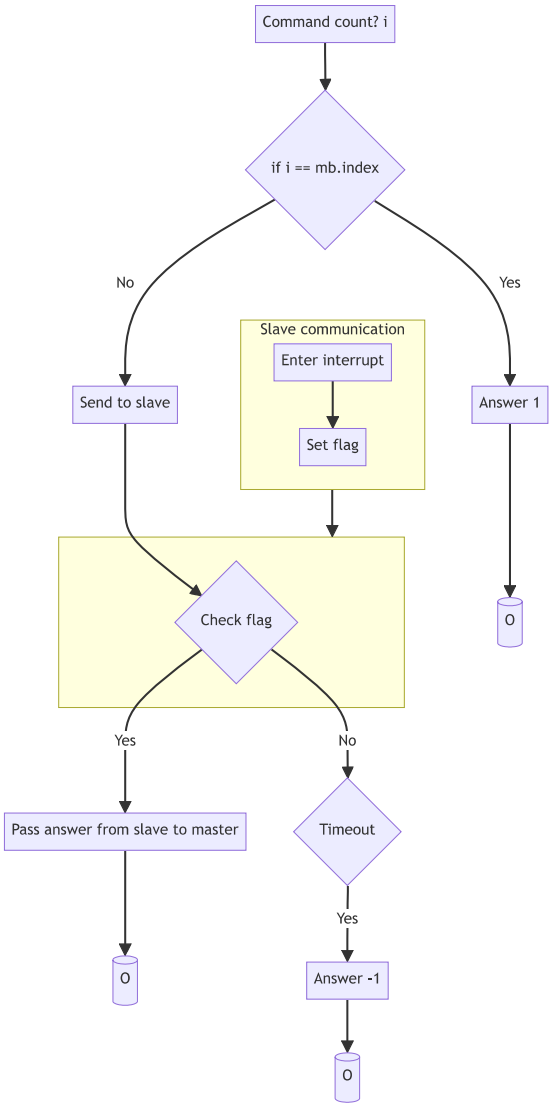


Figure 5.4: Motherboard count request

As it should be clear each action has its own command, which therefore required it to be coded into the parser for it to recognise the intentions of the message.

A special mention goes to the parser, which is the part of the code responsible for the message recognition; parsing is, in fact, the analysis into logical syntactic components. Most of the job is done by reusing a code, whose purpose was indeed to treat SCPI commands, and by adding the syntax of the newly established commands.

5.3 Software

Moving on to the PC software, it was developed in MATLAB [11], with the help of MATLAB App Designer [12]. In this case the choice of the coding language is merely due to a personal preference, correlated with a better knowledge of it compared to the other possible solutions.

The software is meant to provide a graphical user interface (GUI) whose intents are multiple. First of all, the GUI needs to show each channel state, with each of its relays, and the power supplies states; even more, it needs to manage the communication with the NUCLEO boards. As well, it needs to give the possibility to operate both automatic and manual tests; further more, an emergency stop needs to be implemented. Finally, some admin components like save or load some data.

A look at the design of the channels state interface is shown in Figure 5.5.

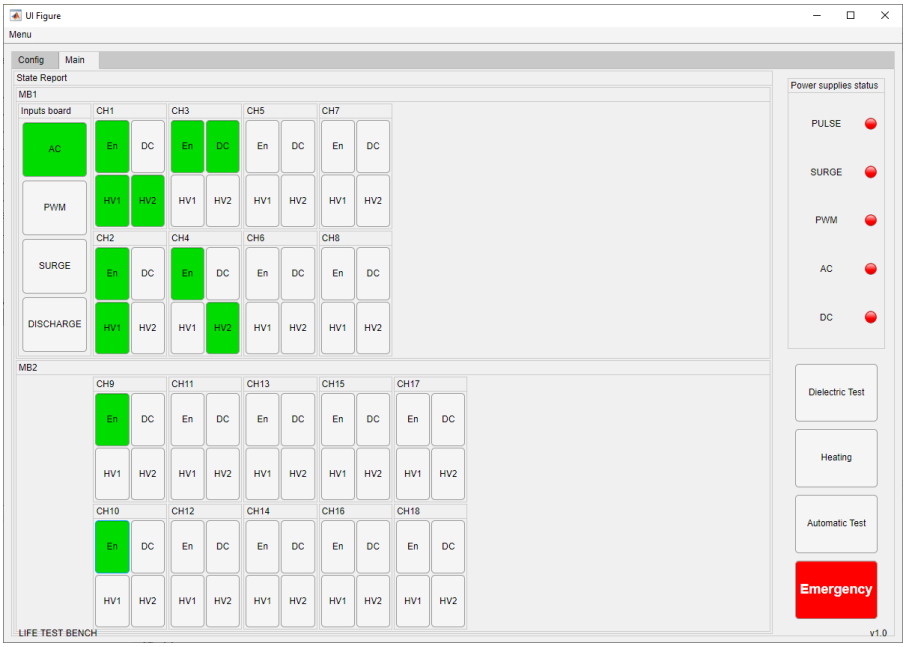


Figure 5.5: GUI state report tab

The graphical side needs to be combined with the object-oriented program, whose structure replicates the one applied to the firmware.

The very first aspect of the software was enabling the serial communication with the first NUCLEO board, which requires some input from the user, as the serial port number may change accordingly to the devices plugged in.

Since the number of subracks and the number of child boards in each one is not fixed, thanks to its modular design as well, the creation of both the objects and their graphical representation has to happen "dynamically". In this case, it was chosen to provide the user with the possibility to both load or manually fill in the rack configuration, at which

point it must be pushed a button to confirm the choice and start the hardware and software setups. The creation of the channel panels, shown in Figure 5.5, happens only once the hardware is correctly configured, this way any mistake should be avoided.

Once again, an example of the sequence taken by the software code is shown here (Fig. 5.6). During the manual operation on the relays, a change caused by the button being pushed, recalls the function inside the object which sends the commands to the board. Before any change in the object properties, the correct execution of the command is checked by sending the corresponding request to the board. The change in the object property triggers the change in the GUI.

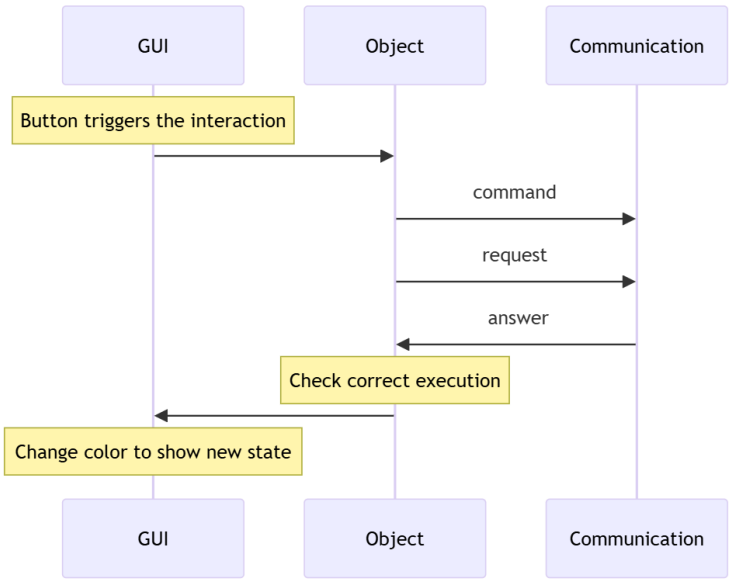


Figure 5.6: Software operation sequence

Chapter 6

Conclusions

The objective set at very the beginning of the internship was to design and construct the test bench, which turned out to be an intriguing, yet fulfilling, challenge. To successfully complete the project, it was necessary to precisely develop a great amount of aspects which characterise the test bench.

To summarise, the first phase required to identify the key parameters for choosing the components, which was accomplished through an analysis of motor datasheets and the study of different potential relay configurations. Once the components were selected, a first prototype was assembled, whose objective was to validate the design of the electrical circuit. Afterwards, the printed circuit boards were designed, which required careful attention to both electrical and mechanical constraints. The final phase involved developing the firmware and software components of the project.

While the firmware was successfully finished, there was limited time to fully develop all desired software functions. However, the manual operations were successfully implemented, which allowed for the initial tests to be conducted.

From the hardware point of view, at least one child board was assembled, which required the components to be soldered onto it, therefore enabling verification of the test bench's correct operation.

The project's achievement is depicted in Figure 6.1, which presents a subrack in its testing phase, fitted with its input board and a child board. The white front panel hides a bypass for the high current circuit.

In conclusion, while the project is still ongoing, a solid foundation has been laid. The electrical and mechanical designs were successfully developed, the firmware is functional, and the software is capable of performing manual operations on the channels, therefore achieving the fundamental requirement of the internship.

Moreover, this work represents a significant step forward in enhancing and expanding the testing facilities at ETEL S.A., the progress made thus far provides, indeed, a robust platform upon which the full functionality of the test bench can be built.

On a personal note, I found the project highly rewarding, particularly due to the valuable exchanges with my colleagues throughout the internship, which not only contributed

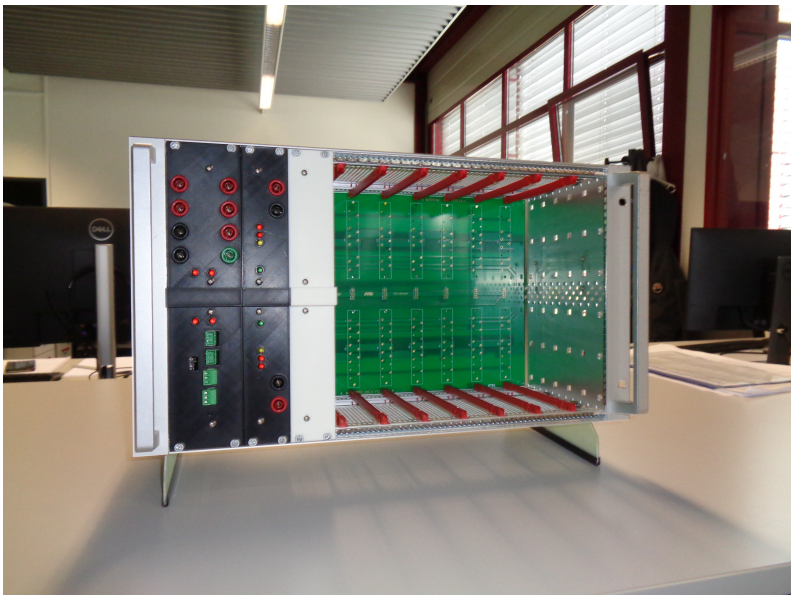


Figure 6.1: Final product

to the project's success but also enriched my learning and professional growth. Moreover, the hands-on experience and problem-solving skills I developed will be crucial as I advance in my career, providing a strong foundation for overcoming future engineering challenges.

Appendix A

Electrical tests

Dielectric tests are crucial for evaluating the reliability and longevity of insulation systems in electrical motors.

Accelerated high voltage dielectric aging tests simulate long-term aging by applying intensified electrical stress, allowing for the prediction of insulation lifespan and identification of potential failure mechanisms in a shorter timeframe.

Dielectric thermal aging tests assess the degradation of insulating materials due to prolonged exposure to high temperatures, which can lead to reduced dielectric strength and increased energy losses, critical factors for motor efficiency and durability.

Additionally, surge tests subject motor insulation to high-voltage pulses that replicate switching surges, helping to identify vulnerabilities that could lead to insulation failure, particularly about turn-to-turn short circuits.

Another important aspect of dielectric testing involves injecting a pulse-width modulation (PWM) high voltage signal. This test objective is to assess the insulation's performance under fast-rising voltages, which are characteristic of modern electronic controls and inverters used in motor systems. The PWM signal creates rapidly changing voltage conditions that simulate the effects of real-world switching operations and high-frequency electrical disturbances. By evaluating the insulation's response to these fast transients, engineers can better understand its resilience against high-speed voltage changes, which is essential for ensuring reliable operation in advanced motor control applications.

Together, these dielectric tests, ranging from accelerated aging and thermal stress assessments to surge and PWM signal evaluations, are vital for ensuring the long-term performance, safety, and reliability of electrical motors in demanding operational environments.

Appendix B

Additional figures

The remaining electrical schematics are represented in the following figures.

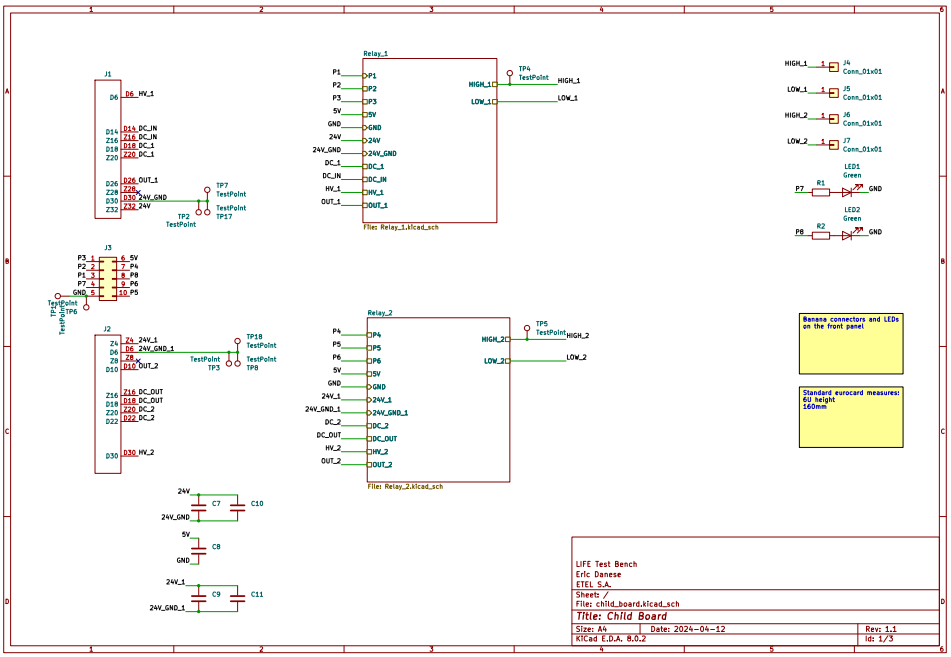


Figure B.1: Child board electrical schematic

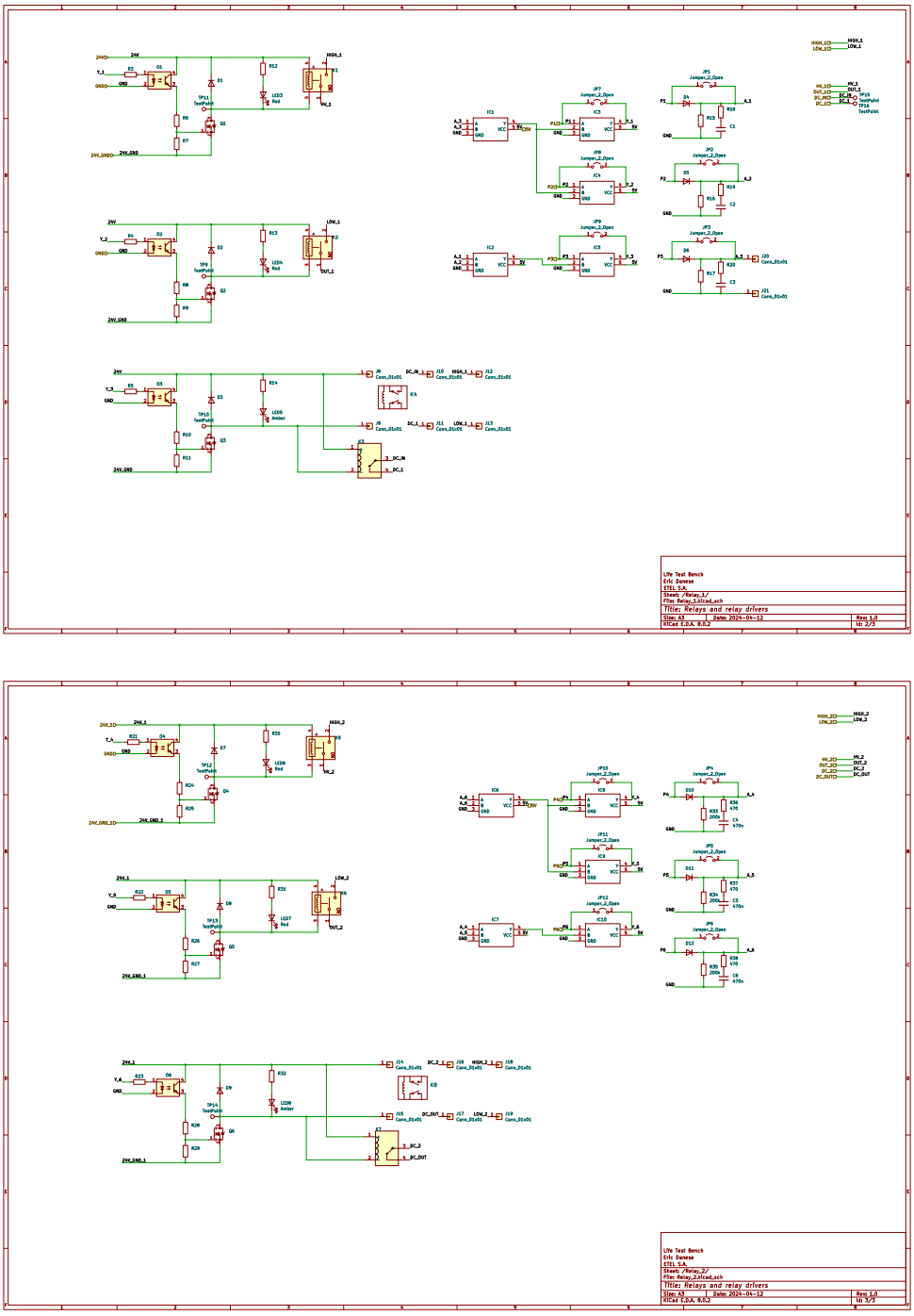


Figure B.1: Child board electrical schematic (cont.)

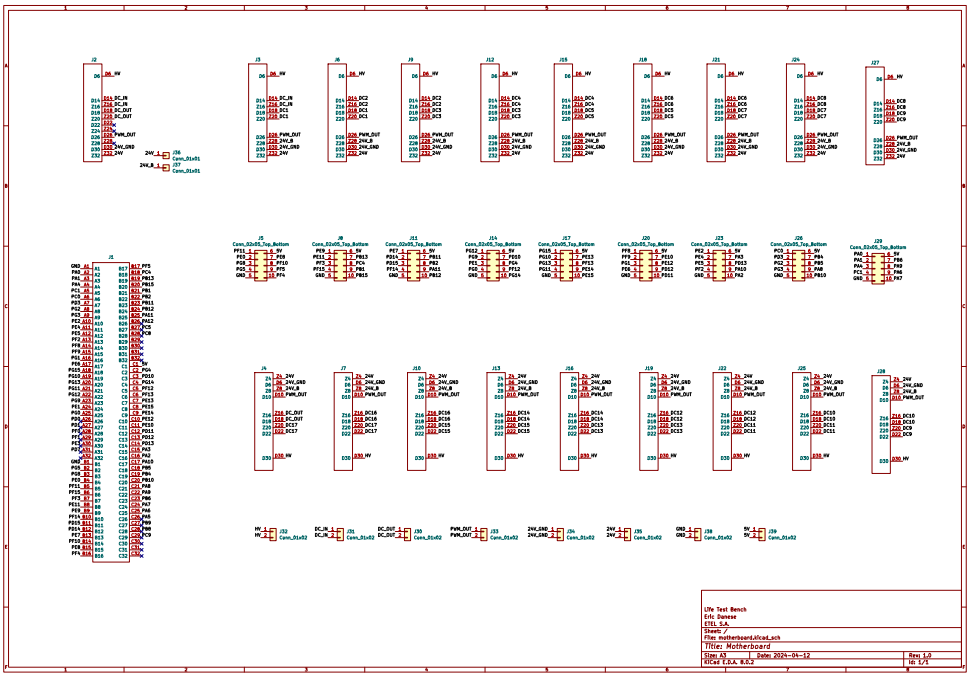


Figure B.2: Backplane electrical schematic

The two other board 3d views are shown here.

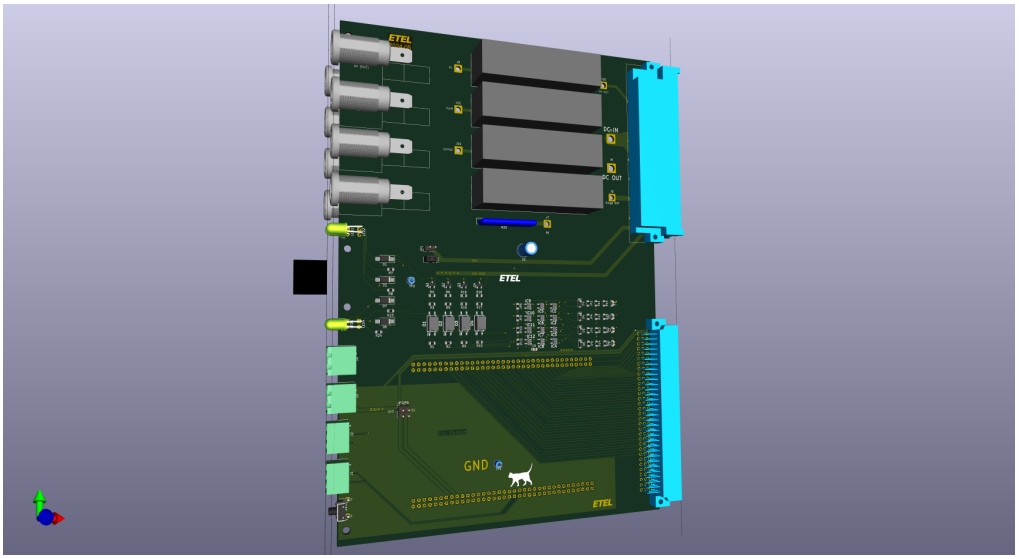


Figure B.3: Inputs board 3d view

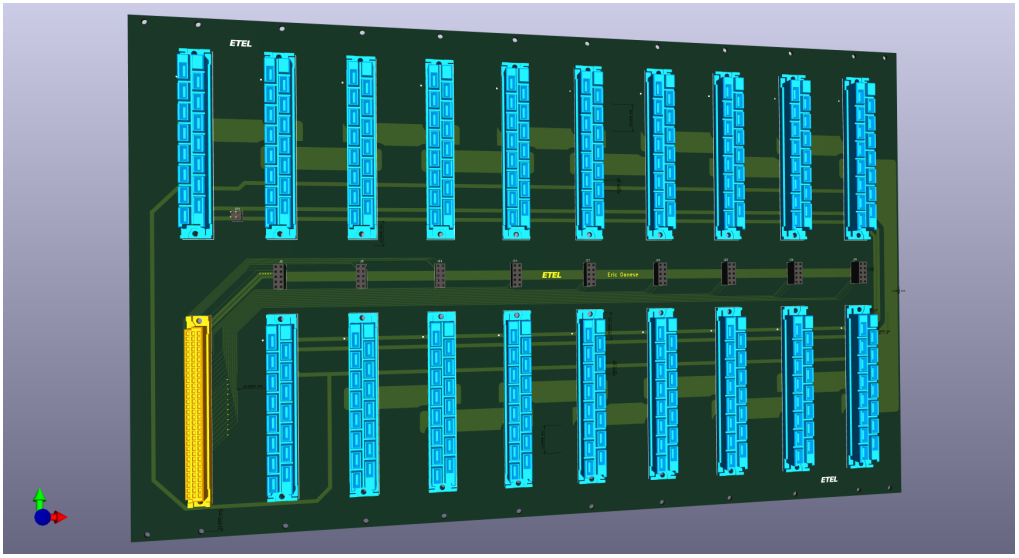


Figure B.4: Backplane 3d view

Bibliography

[1] *Home - ETEL S.A.* en. Jan. 2021. URL: https://www.etel.ch/ (visited on 08/09/2024).

[2] Analog Devices. *LTspice*. Wilmington, Massachusetts, United States, Aug. 2024. URL: https://www.analog.com/en/resources/design-tools-and-calculators/ltspice-simulator.html.

[3] STMicroelectronics. *STMicroelectronics*. https://www.st.com/content/st_com/en.html. Accessed: 2024-08-17. 2024.

[4] *Honigmann: Subrack with backplane*. URL: https://honigmann.com/i335/Subrack-with-backplane.html&lang=2 (visited on 07/09/2024).

[5] *Connectors for frequencies below 3 MHz for use with printed boards - Part 2: Detail specification for two-part connectors with assessed quality, for printed boards, for basic grid of 2,54 mm (0,1 in) with common mounting features*. en. International Standard. International Electrotechnical Commission, Sept. 1995. URL: https://webstore.iec.ch/en/publication/2696.

[6] *IEEE Standard for Mechanical Core Specifications for Microcomputers Using IEC 60603-2 Connectors*. 1998. DOI: 10.1109/IEEESTD.1998.88824.

[7] Jean-Pierre Charras. *KiCad*. July 2024. URL: https://www.kicad.org/.

[8] *Adjustable speed electrical power drive systems - Part 5-1: Safety requirements - Electrical, thermal and energy*. en. International Standard. International Electrotechnical Commission, Aug. 2022. URL: https://webstore.iec.ch/en/publication/62103.

[9] *High Voltage PCB Design: Creepage and Clearance Distances for High Voltage*. en. Aug. 2017. URL: https://resources.altium.com/p/high-voltage-pcb-design-creepage-and-clearance-distance (visited on 08/05/2024).

[10] STMicroelectronics NV. *STM32CubeIDE*. Plan-les-Ouates, Geneva, Switzerland, July 2024. URL: https://www.st.com/en/development-tools/stm32cubeide.html#st_description_sec-nav-tab.

[11] The MathWorks Inc. *MATLAB*. Natick, Massachusetts, United States, 2019. URL: https://ch.mathworks.com/products/matlab.html.

[12] The MathWorks Inc. *MATLAB App Designer (R2019b)*. Natick, Massachusetts, United States, 2019. URL: https://ch.mathworks.com/products/matlab/app-designer.html.