Tampere University

Gonzalo Díaz Alcaraz

REDESIGN OF IMORO ROBOT

Industrial engineer
Bachelor´s thesis
April 2019

ABSTRACT

Gonzalo Diaz: Redesign of iMoro robot
Bachelor´s thesis
Tampere University
Exchange student
April 2019

IMoro is a robot in which students and researchers had been working on for more than 10 years. Some thesis and post-doctoral researches have been made about iMoro. This robot is actually in RoboLab, where it can be visited. Students have the freedom to go there and play or work with the different kind of robots, so this thesis is focused to ease this work with iMoro.

This thesis continues the work made by other students and improves the robot as much as possible. To achieve that, a series of improvements, mainly in hardware part, are going to be exposed in this thesis. Some improvements in the software parts will be also made, but only as first step of what can be done.

IMoro is a multidirectional robot with four wheels which can be moved independently forward or backward, and around Z axis. A 6 degree of freedom manipulator is also attached to the platform with a tiltable base which can be inclined 45º. The power supply is administrated to the robot with a complex electrical circuit, which automatically switch between the batteries and the charger station. All these mechanisms, in addition to others, will be explained and in detail.

The phases of the development of this thesis involve: compilation of information related to past studies, disassembling the robot to understand its functioning, structuring of a plan for different improvements and implementation of this advances.

The improvements made on iMoro in the Hardware part are: device redistribution, redo of the wiring, creation of a second platform, creation of an external case and creation of a charger station. On the other hand, in the software part, a user´s manual is started with the steps to connect your personal computer with the onboard computer robot.

PREFACE

This thesis has been developed in the RoboLab at Tampere University. The following persons had contributed in the work done:

Contribution	Name	Organization
Author	Gonzalo Díaz Alcaraz	UC3M
Collaborators	Luis Picos Feijoo	UC3M
	Luis Fernández Palomino	UC3M
	Arnau Sastre Martínez	UPC
Reviewer	Reza Ghabcheloo	TUNI/TAU
	Janne Uusi-Heikkilä	TUNI/TAU
	Veli-Pekka Pyrhönen	TUNI/TAU
Additional helpers	Mohammad Mohammadi Aref	TUNI/TAU
	Janne Koivumäki	TUNI/TAU

Tampere, 21 April 2019

Gonzalo Díaz Alcaraz

CONTENTS

LIST OF SYMBOLS AND ABBREVIATIONS

TUNI	Tampere University
UC3M	Universidad Carlos III de Madrid
UPC	Universidad Politécnica de Cataluña
CAD	Computer-aid Design
V	Volts
PC	Personal Computer
USB	Universal serial bus
A	Amps
PLA	Polylactic acid
FabLab	Fabrication Laboratory
IP	Internet Protocol
PCI	Peripheral Component Interconnect
TCP/IP	Transmission Control Protocol/Internet Protocol
WIFI	Wireless Fidelity
MAC	Media Access Control
BIOS	Basic Input/Output System
VGA	Video Graphics Array

1. INTRODUCTION

A machine (or mechanical device) is a mechanical structure that uses power to apply forces and control movement to perform an intended action [1]. In this case, iMoro robot is a machine which complex merge many engineer fields to achieve its expected performance. This performance is mainly focused on its freely of movements. Thanks to its multidirectional wheel system, iMoro can follow complex paths with high accuracy, therefore it allows a high level of reachability points for its manipulator.

The objective of this project is look into this complex robot and, at the same time introduces some improvements on it, as many other researchers and students made before. These series of improvements are going to be detailed and explained in the following chapters. Additionally, there is going to be a redistribution of the devices and an update of the wired to obtain a better performance and ease future work on iMoro.

First, the thesis starts with the explanation of all the work made previously in the robot, which is going to be explained in the *Chapter 2*. This chapter also includes documentation and explanations about the functioning of each device that is on the robot. This work includes explanation of schemes, bachelor's thesis and other documents so all the information related to the robot is gathered and organized.

Secondly, in the *Chapter 3.1*, are explained the series of improvements applied on the hardware part of the robot, from the idea to the final implementation. On the *Chapter 3.2,* a user´s manual is created to ease the use of the robot for people external to the project.

Thirdly, in the *Chapter 4*, a series of future developments and ideas are exposed as they couldn´t be done in that thesis because of the lack of time or specialized background knowledge.

Finally, on the *Chapter 5*, the results achieved in that thesis will be exposed and analysed. A Chapter of references and bibliography will be included at the end of the thesis.

2. BACKGROUND INFORMATION

iMoro is a complex robot which was created several years ago. This chapter aims to collect and organize all the information related to the robot which can be useful for future implementations or for a better understanding of its functioning.

2.1 Projects and programs

On this section of the background information, the writing reports and electrical scheme are going to be explained to have an overall view of its contribution to the robot. Then the Computer-aid design (CAD) files will be introduced and explained, and finally, the Matlab program will be explained with their files.

The reports write about the robot are two. The first one is *Conceptual design document for `Modular Mobile Manipulator System´*. This document is the design description of iMoro robot. It covers all the conceptual design done for the manipulator platform. The document focuses mainly on mechanical and electrical design of the system, which aims to clarify the current design and background behind design decisions. Also available interfaces are covered in case the system is extended in the future [2]. The appendixes of the document are: *Modular Wheel Unit for Mobile Platform, Structural Design of Mobile Platform and Electrical System.*

The second report is the Bachelor´s thesis of Eemeli Anttila, named *Human-Machine Interface Design for an Omnidirectional Mobile Manipulator.* This thesis aims at designing a human-machine interface, consisting of a graphical user interface (GUI) and a physical input device, for operating the iMoro mobile manipulator for Laboratory of Automation and Hydraulics (AUT) [3].

On the other hand, the electrical scheme document shows how the electrical circuit of the robot and their connections are. In this thesis, this circuit will be explained in detail in the *Chapter 3.1.3*, as well as the future modifications that are going to be introduced in the robot.

Moreover, there are CAD files about iMoro, which had been made by Janne Koivumaki on the aim to model the mechanical part of the robot and some of the devices. The *Figure 1* shows how the CAD model was when this thesis started. As we can see, many of the devices were not created yet, and some of them will have to be modified.

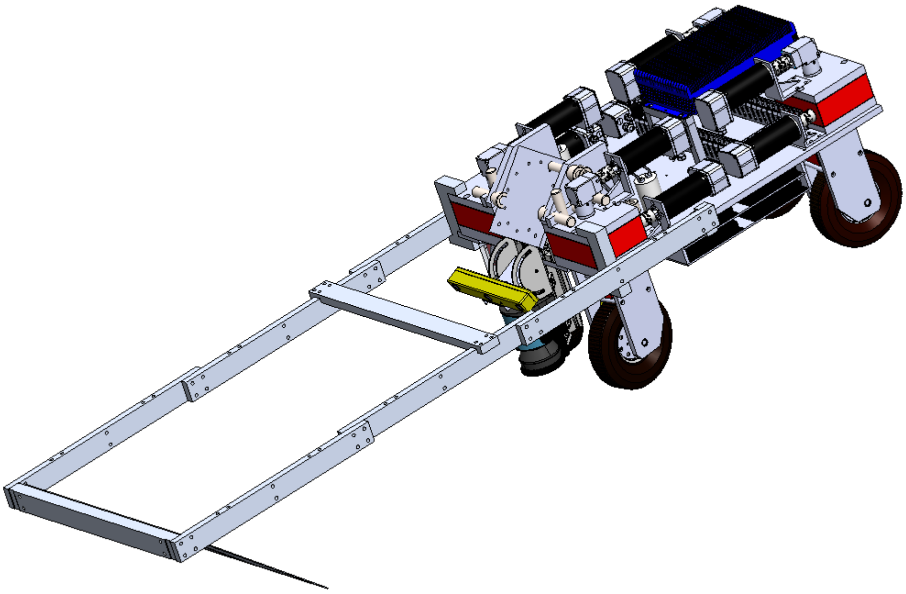


Figure 1. *Original CAD model of iMoro robot*

After finalizing this thesis, and modify the CAD files, they will be provided in a USB flash, where there are two folders: The first one, named `iMoro CAD Janne´ contains the old CAD files; and the second one, named `iMoro CAD Gonzalo´ contains the new CAD files updated after this thesis.

Finally, another folder with the Matlab files which controls the robot will be included in the USB. Some of the programs included in that folder will be explained more in detail in the *Chapter 3.2*, where the starting of a user´s manual is going to be created to use the robot.

The *Figure 2* shows how iMoro robot was when the bachelor´s thesis started.

Figure 2. *Appearance of iMoro robot on September 2018*

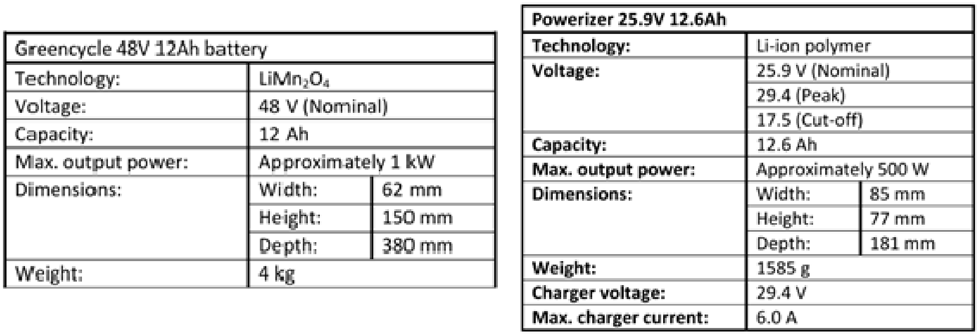
2.2 Device documentation

The previous report named *Conceptual design document for `Modular Mobile Manipulator System´* explains which devices has been included on the robot on its Appendix 3. But, instead of that, there is some misinformation or information that should be clarified. This chapters aims to update that document with more detailed and clear information to ease the understanding of the robot for future improvements.

Although in this chapter we intend to update the information, there is an omission of some devices which actually are out of use or they are not in the robot. The encoders, camera and IMU system, are some of those devices that hadn´t been included. For future implementations, check the document *Conceptual design document for `Modular Mobile Manipulator System´* for complete the information related to the devices.

2.2.1 Power supply

The power can be supplied by the batteries or by the charger station, which is directly connected to the current. In this case, the device has a good documentation for the batteries, so only the table with the specifications are shown in the *Table 1*.



Greencycle 48V 12Ah battery		
Technology:	$LiMn_2O_4$	
Voltage:	48 V (Nominal)	
Capacity:	12 Ah	
Max. output power:	Approximately 1 kW	
Dimensions:	Width:	62 mm
	Height:	150 mm
	Depth:	380 mm
Weight:	4 kg	

Powerizer 25.9V 12.6Ah		
Technology:	Li-ion polymer	
Voltage:	25.9 V (Nominal)	
	29.4 (Peak)	
	17.5 (Cut-off)	
Capacity:	12.6 Ah	
Max. output power:	Approximately 500 W	
Dimensions:	Width:	85 mm
	Height:	77 mm
	Depth:	181 mm
Weight:	1585 g	
Charger voltage:	29.4 V	
Max. charger current:	6.0 A	

Table 1 [2]. *Batteries specifications*

Two converters AC/DC of 48V and 24V had been used for the charger station. The converters specifications are summarized in *Table 2*.

Device	Model	Linkable datasheet
48 V Converter	Power supply unit - TRIO-PS/1AC/48DC/10	https://www.phoenixcontact.com/online/portal/us?uri=pxc-oc-itemdetail:pid=2866501&library=usen&tab=1
24 V Converter	Powernet ADC5423-3	http://windcluster.com/wp-content/uploads/Power-Supply-ADC5000.pdf

Table 2. *Converters specifications*

The last device of the charger station is a switch button, which function is better explained in the *Chapter 3.1.5*. The *Table 3* summaries the related information about it.

Device	Model	Linkable datasheet
Switch	OC10G03PNBN00NB3	https://new.abb.com/products/1SCA134996R1001/oc10g03pnbn00nb3-cam-switch

Table 3. *Switch specifications*

2.2.2 Computer

The robot is actually controlled by one onboard computer. However, the idea is to change that computer to a new version named SpeedGoat, which is especially designed for the using of Simulink Real-Time and has many programs optimized for it. Furthermore, there is the idea to add a second computer to distribute computational calculus and increase the performance of the robot. The *Table 4* shows the specification of the actual computer that is been used.

Device	Model	Linkable datasheet
Onboard computer	iEi TANK-720-Q67	https://dls.ieiworld.com/IEIWeb/PDC_OBJ/PLM/WFP003451-IEI/TANK-720_UMN_v2.02.pdf

Table 4. *Onboard computer specifications*

2.2.3 Drivers

The robot has four wheels, each of them is ruled by two drivers which are in charge of the forward and backward movement and the rotation around Z axis. Each of these drivers are connected with the motors to provide the instructions about how to execute the movement, and then connected to the computer. The *Table 5* shows the specifications of the drivers.

Device	Model	Linkable datasheet
Drivers	Maxon Motors EPOS2 70/10, Digital positioning controller, 10 A, 11 - 70 VDC	https://www.maxonmotor.com/maxon/view/product/control/Positionierung/375711

Table 5. *Drivers specifications*

In the case of this device, there are manuals about how to do the connections and use the drivers in the link of the *Table 5*.

2.2.4 Motors

The motors are connected to the wheels, as it was explained before, and execute the information about the movement requested. On the *Table 6* there is the related information related about the motors that iMoro is using.

Device	Model	Linkable datasheet
Motors	Maxon Motors EC 400W 167132	https://www.maxonmotor.com/maxon/view/product/167131

***Table 6**. Motors specifications*

The brand is the same as the drivers, which ease the synchronisation with the manuals and guides in both links.

2.2.5 I/O Components

The communications of iMoro are done by CAN bus system. Because of that, two CAN converters from analogical to digital, and from digital to analogical signal, had been added to provide a wider range of option in future implementations. The specifications of these converters are summarized in the *Table 7*.

Device	Model	Linkable datasheet
A/D Converter	CAN-CBX-AI814	https://www.esd-electronics-usa.com/shared/handbooks/CAN-CBX-AI814_Manual.pdf
D/A Converter	CAN-CBX-DIO8	https://www.esd-electronics-usa.com/shared/handbooks/CAN-CBX-DIO8_Manual.pdf

***Table 7.** Analogic converters specifications*

2.2.6 Laser

The laser supports the orientation system of iMoro. The laser chosen is SICK S3000 and its specifications are reflected in the *Table 8*.

Device	Model	Linkable datasheet
Laser Range Finder	SICK S3000	https://www.sick.com/ag/en/opto-electronic-protective-devices/safety-laser-scanners/s3000-standard/c/g187231

Table 8. *Laser specifications*

2.2.7 Manipulator

IMoro is divided in two main parts, the body and the manipulator. This manipulator is placed in a platform which can be tilted 45º to increase the reachability of the manipulator. This configuration is better explained in the Appendix 2 of the document *Conceptual design document for `Modular Mobile Manipulator System´*. The *Table 9* shows the specifications of the manipulator selected.

Device	Model	Linkable datasheet
Manipulator	Schunk Powerball Lightweight Arm LWA 4P	http://www.schunk-modular-robotics.com/en/home/products/powerball-lightweight-arm-lwa-4p.html

Table 9. *Manipulator specifications*

2.2.8 Router

The robot is controlled by the computer TANK 720, but to ease the changes and live-work on the robot, a router has been added. With this device, the user is able to interact with the onboard computer with its own laptop. The steps to synchronise both are explained in the *Chapter 3.2.1*. The *Table 10* shows the specification of the router.

Device	Model	Linkable datasheet
Router	MOXA AirWorks 1100 Series	http://www.citltda.cl/r_industrial/AWK-1100.pdf

Table 10*. Router specifications*

2.2.9 Safety circuit

One of the main concerns is the safety of the robot. For that reason, a robust and complex safety circuit has been created to keep safe the human as well as the robot. The circuit functioning is explained in the *Chapter 3.1.3* in detail. This circuit is composed by a series of devices which specifications are detailed in *Table 11*.

Device	Model	Linkable datasheet
Modular contactor	Finder modular contactor 22.32.0.048.4520	https://gfinder.findernet.com/public/attachments/22/EN/IB2232EN.pdf
Main Switch	Kraus & Naimer, SP 2 Position Rotary Switch, 440 V, 10 A	https://fi.rsdelivers.com/product/kraus-naimer/cg4a290-gba120ef/kraus-amp-naimer-sp-2-position-rotary-switch-440/0340229
Emergency stop button	Schneider Electric ZBE102 (2 units)	https://www.schneider-electric.us/en/product/download-pdf/ZBE102
Safety relay	Pilz PNOZ s7	https://www.pilz.com/download/open/PNOZ_s7_Operat_Man_21399-EN-09.pdf
Modular contactor	Finder modular contactor 22.23.9.024.4000	https://gfinder.findernet.com/public/attachments/22/EN/S2221_22_23_24EN.pdf
Modular contactor	Finder modular contactor 22.22.9.024.4000	https://gfinder.findernet.com/public/attachments/22/EN/S2221_22_23_24EN.pdf
Fuses	Eaton FAZ-C 20/1,10/1,4/1	http://www.eaton.com/ecm/groups/public/@pub/@seasia/@elec/documents/content/pct_401308.pdf

Table 11. *Safety circuit devices specifications*

3. RESEARCH AND IMPLEMENTATION

The background information part illustrates the great work that has been done previously in the robot. But, despite this work, there are many improvements to make on iMoro. In this chapter we are going to start with the changes and improvements that had been implemented.

The first part of the changes is related to the hardware part, where the aim was to achieve a robust robot which can be used in an industrial environment. Moreover, the robot should be accessible for students and ease the labours of maintenance, work and future development.

The second part is related to the software part. Here, a user´s manual has been created to make the robot more accessible for anyone who wants to use it.

3.1 Hardware

Five main approaches have been made in this part: device redistribution, building of a new platform, redo of the wiring, build an external case and make a new charger station. In advance, these changes are going to be explained in detail, and the reasons to introduce them.

3.1.1 Device redistribution

In the chapter dedicated to the background information we show how chaotic were the main platform distribution. There were devices which were not connected to anything, some of them weren´t fixed and some were missed.

To solve that, all the devices and connections were disassembled and retired to make a study of what was available and how it could be ordered in the most efficiently way to use as little space as possible.

After disassembling the robot, all the connections and devices were catalogued and modelled in CAD. Then, the device distribution of the main platform was created, with the wire guides and the devices which were not in the second platform.

This distribution is showed with an upper view in the *Figure 3*, without the second platform and the motors for a better visualization. In the *Figure 4* the distribution is showed with all the motors in different angle. Finally, in the *Figure 5*, the final distribution is showed including the second platform.

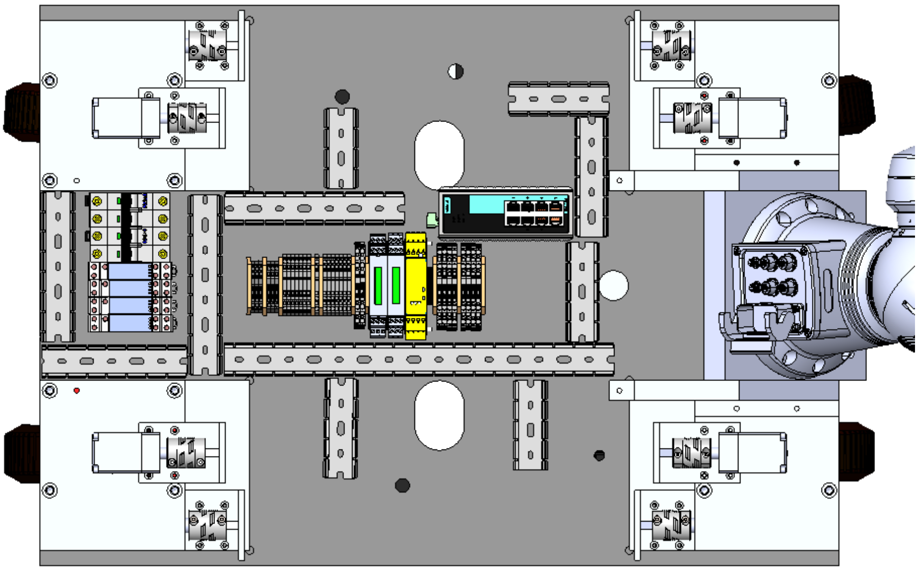


Figure 3. *Upper view of the main platform device distribution*

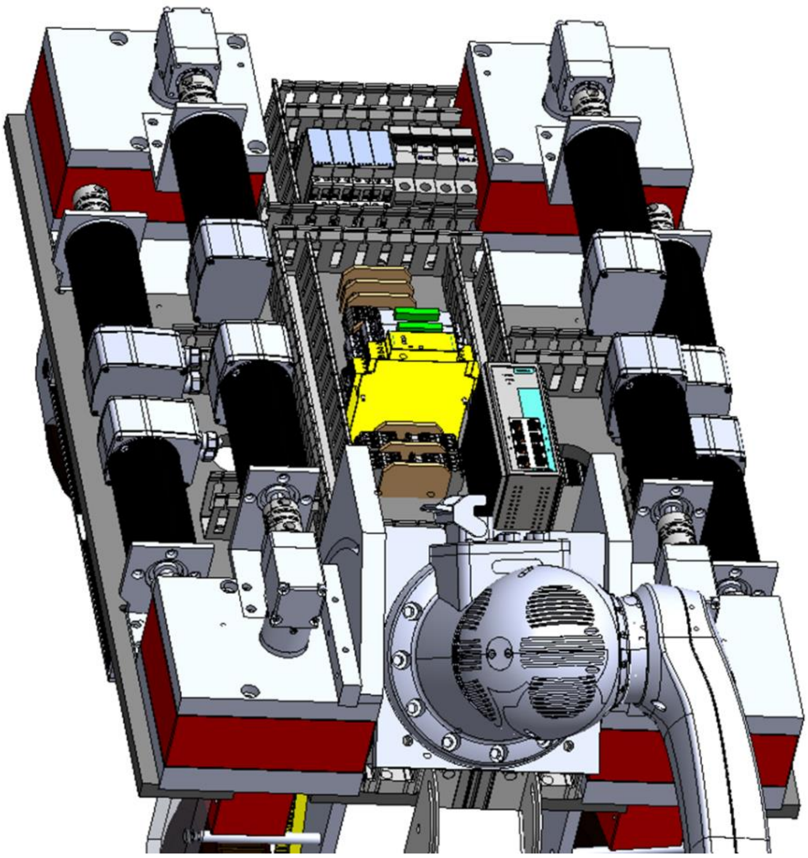


Figure 4. *View of the Main platform and the motors*

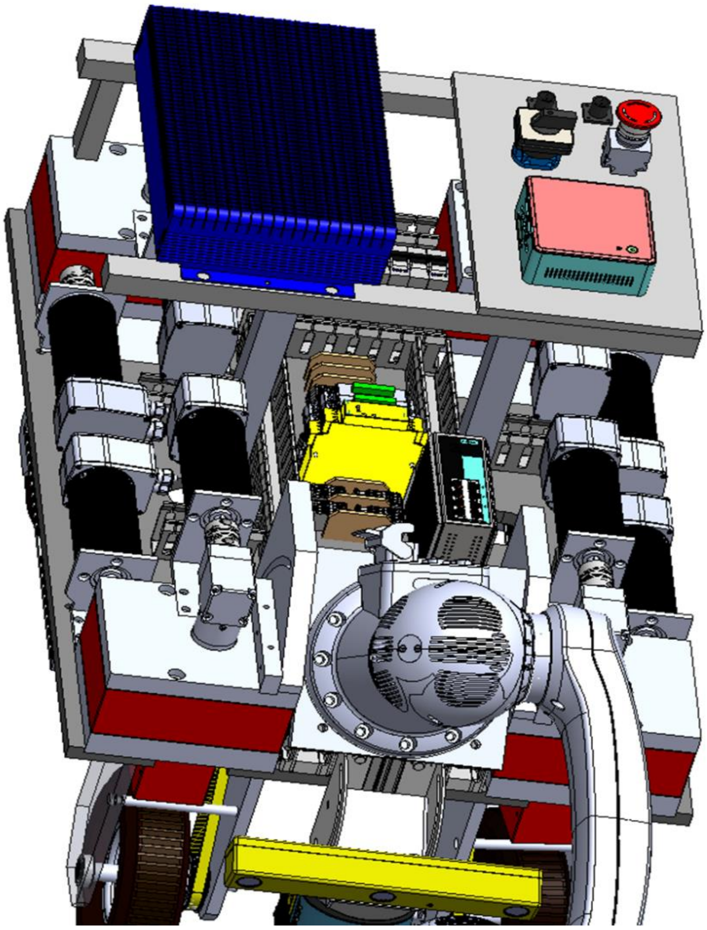


__Figure 5.__ View of the main platform, the motors and the second platform

The next stage of planning is the implementation on the robot. To do that, new holes had been made in the platform to attach the wire guides and the electrical devices. The second platform had also required new holes for the screws and their connections.

These holes, as the pieces of the second platform, have been created in the laboratories of Tampere University, sometimes by technicians, sometimes by the author. To make these holes, multiple drilling machines have been used, as well as other tools and machinery to build the pieces.

On the other hand, there was a problem with the CAN bus signal and the power supply of the drivers. The drivers were attached below the main platform and they were connected to the motor with wires which go through the big holes in the side, which can be shown in the *Figure 3*. The problem was that the power supply wires of 48V goes through the same hole, which caused interferences in the CAN bus signal.

The solution of this problem was easy: make new holes which go exactly over the power supply connection of each driver. With these new holes, we could separate the CAN bus and drivers supply as much as possible to reduce the interferences. In the *Figure 3* we can see the wire guides that go to the new holes made.

Extra modification

The device redistribution is not the only modification that is going to be covered on this chapter. The robot has a special holder for the laser, which helps in the orientation system, and the camera. These devices have a high precision, so they require a previous calibration.

The problem comes when the robot starts moving. The holder shakes a bit with this movement, which difficult the labours of calibration and it is necessary to do this again every time the robot is started.

In addition, the part where is placed the holder have a minimum working space, which difficult the removing of the holder to change the place of the camera. To solve these problems, a special piece has been designed to reduce the movement of the holder to the minimum and, at the same time, ease the removing of the holder. The *Figure 6* shows the new piece and how it fits in the holder structure.

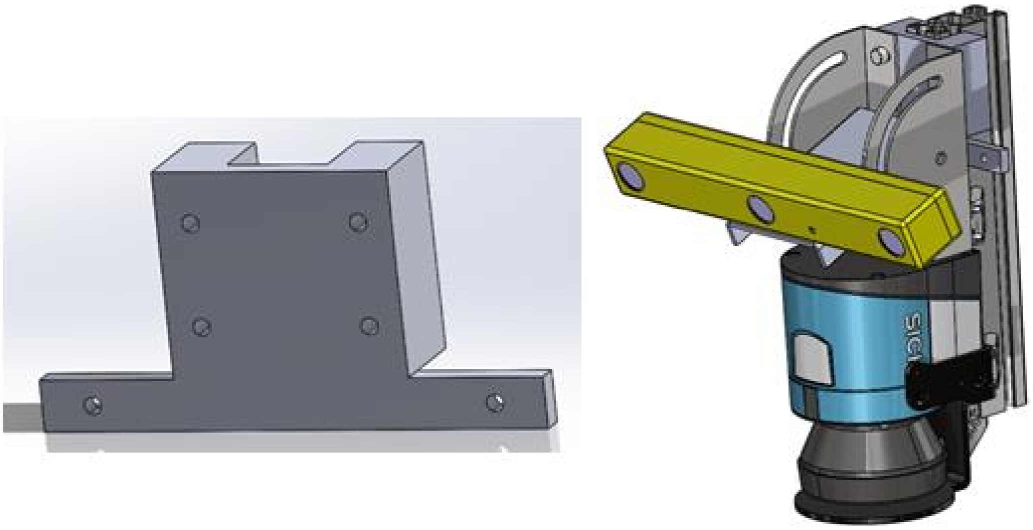


Figure 6. *New holder piece on the left. Laser and camera with the new holder piece in the right.*

As it can be seen in *Figure 6*, the piece is attached to the main platform with the holes of the sides, while the centre holes allow a different high for the support of the camera. This part is very important in the case that the manipulator is working with its base at its maximum angle, because it would collide with the camera support when it is placed in the upper holes as we can see in the left image of the *Figure 7*.

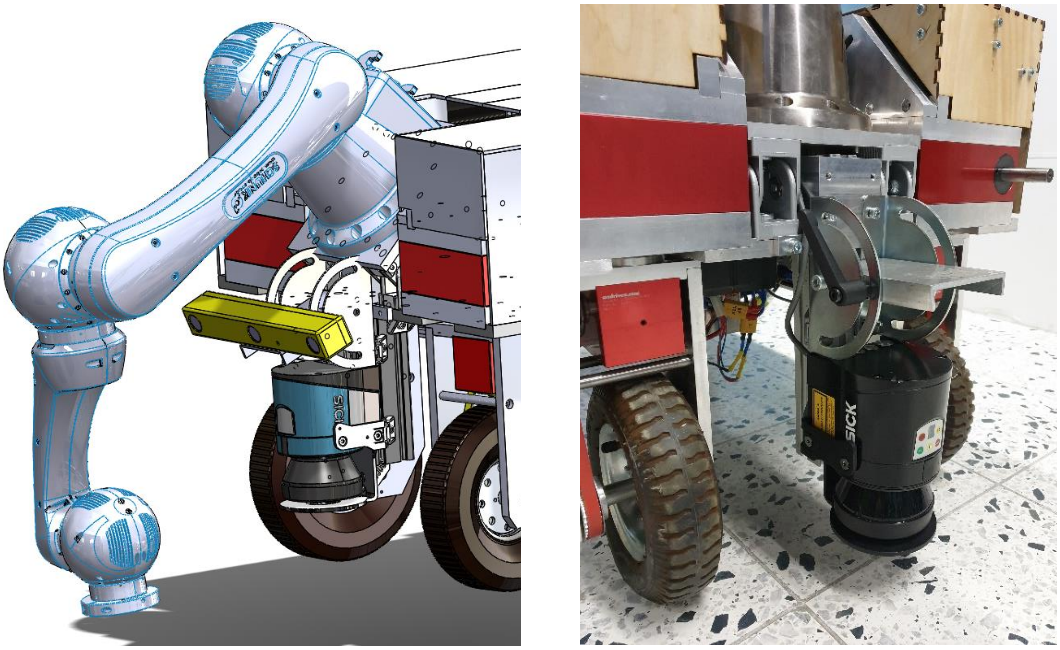


***Figure 7.** Laser holder platform in CAD model and in real life*

On the right of the image of the *Figure 7,* we can see the final appearance of the laser holder platform. In this case the base of the manipulator is not in angle.

3.1.2 New platform

The robot has many devices to be placed, so with only one level (main platform) there is not enough space to do it. For this reason, a new second platform has been created. In the previous disposition of the robot, there was one platform which holds the big PC, but now, with the introduction of a new PC and the new device distribution in the main platform, there was to build a new one.

This new platform should be able to hold the old PC (iEi TANK-720-Q67) and other devices as the emergency stop button, the main switch, the charger connections and the second PC. The place for the actual onboard PC should allow the change to a new one (SpeedGoat) which will be made in the future. But, as both PC have the same dimensions, this would not be a problem.

The connections to charge the batteries have been also added below one of the horizontal bars, for not confusing the connections between batteries and the charger station.

The *Figure 8* shows the platform designed with CAD, a simple but effective structure to place the mentioned devices. In the *Figure 9* we can see these devices placed in their positions.

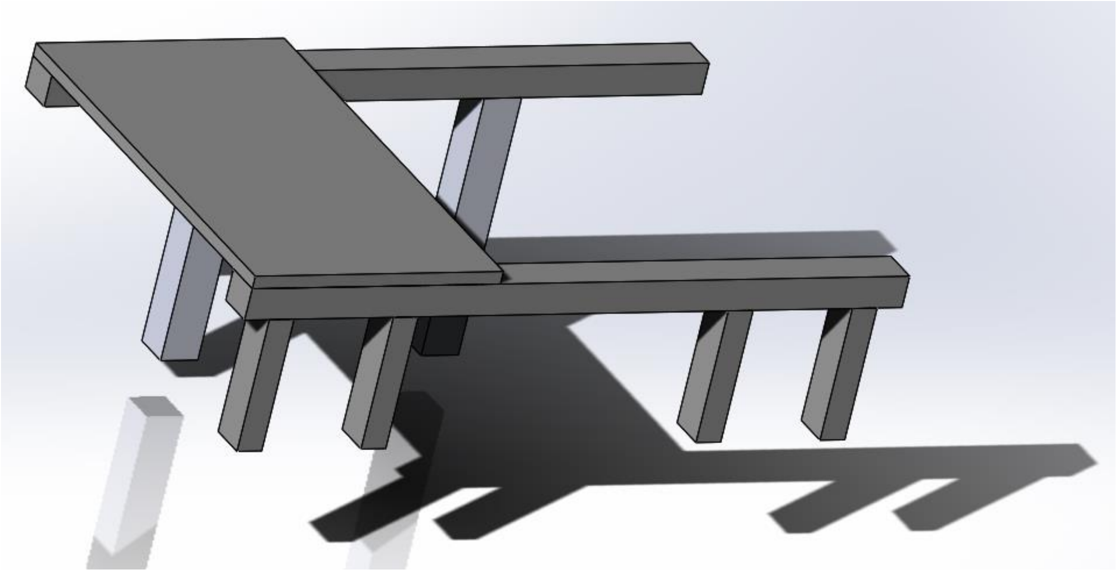


Figure 8. *CAD model of the new platform*

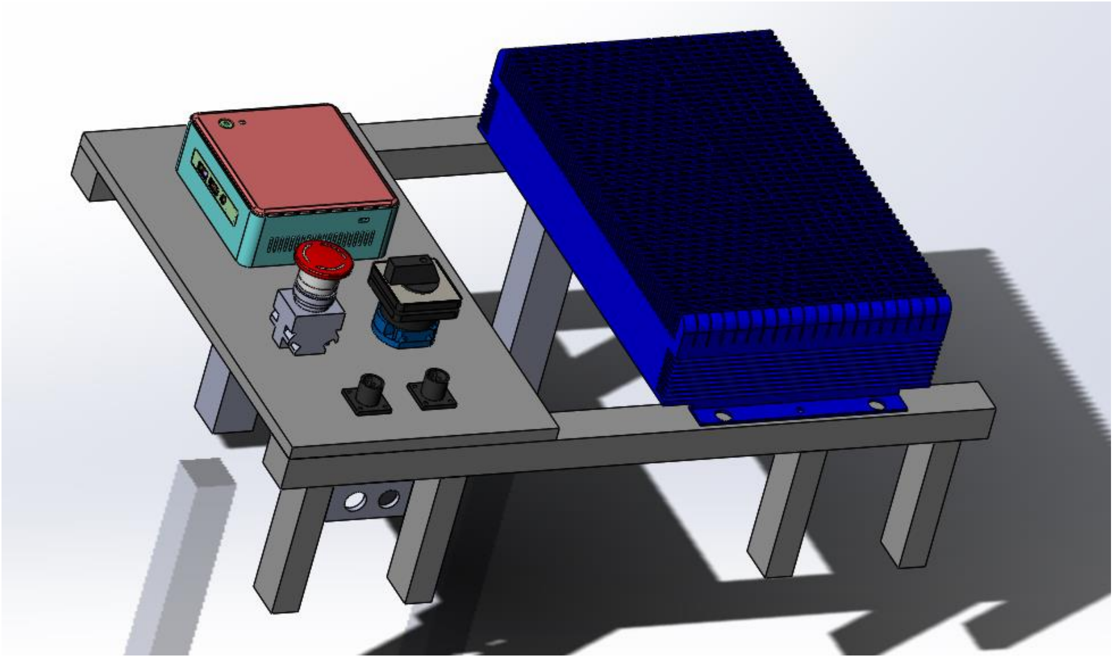


Figure 9. *CAD model of the new platform with its devices*

The platform in the left provide the freedom to replace some devices or even introduce new one. The left side is reserved for the big PC, which in the future can be substituted by a SpeedGoat, as it was explained before, has the same size and will perfectly fit in the structure.

The processes to accomplish this task were:

1. Measure the big PC and the space we have to avoid collisions with the manipulator and optimise the space as much as possible.

2. Create the CAD model and send them to build to the laboratory.

3. After receiving the pieces, go to the laboratory again to make the holes in the correct places with the drilling machine.

4. Finally, think how to attach the different devices to the platform once the platform was fixed properly. For this task, several holes where made with different sizes and shapes depending of the device that had to be attached and how it will be fixed better to make a robust connection.

The *Figure 10* shows the final appearance of the new platform on the robot.

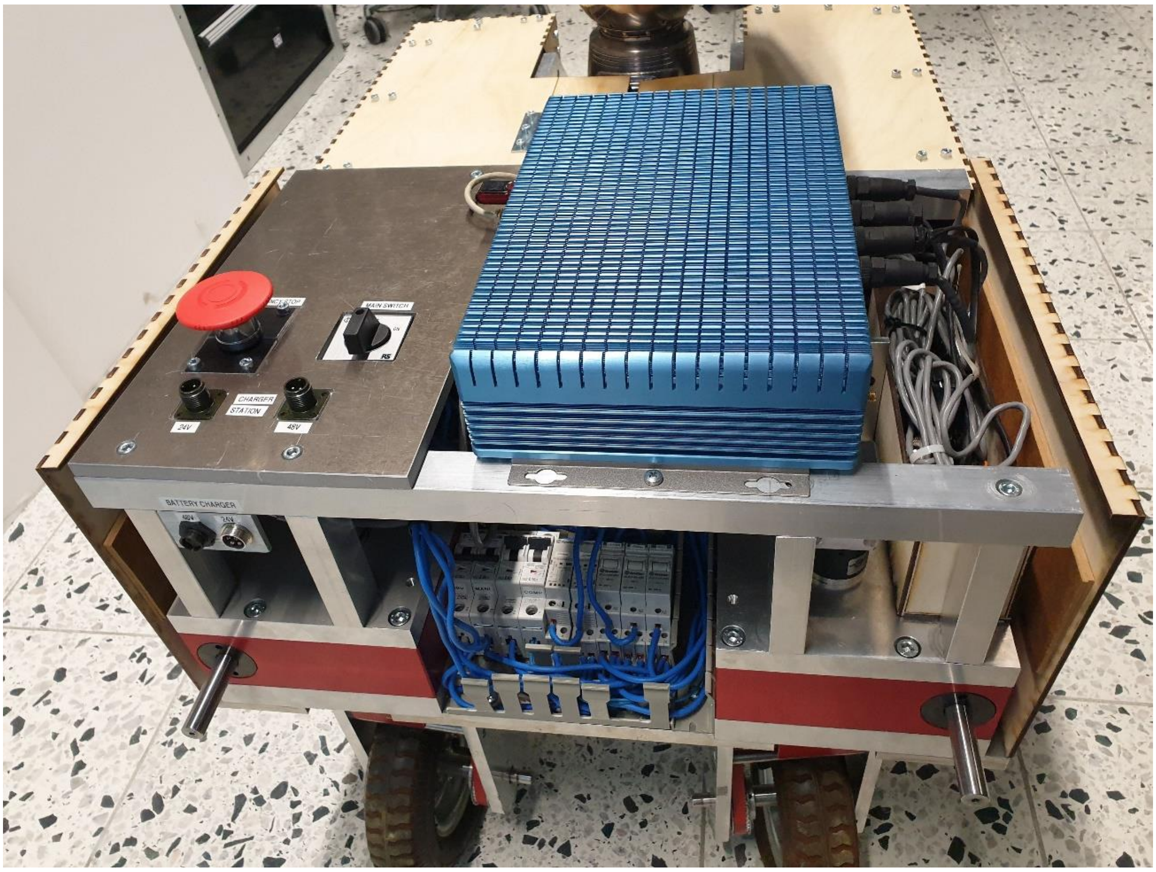


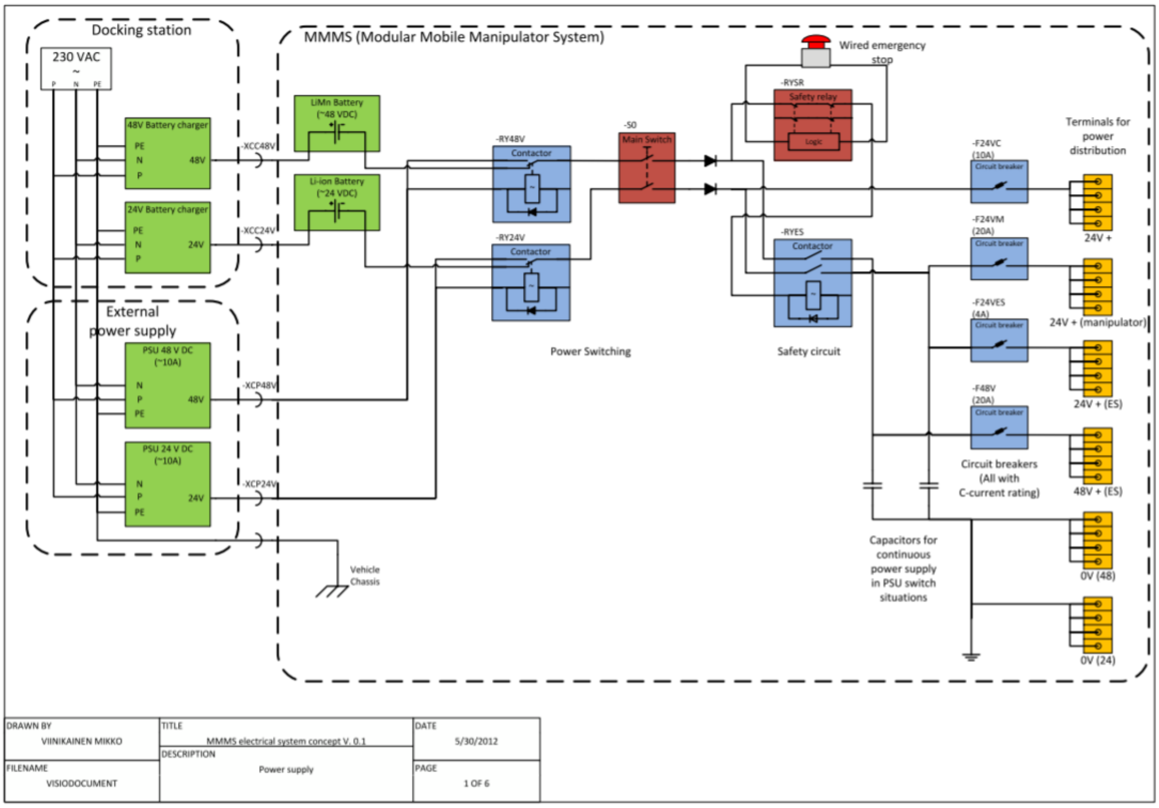
Figure 10. *Final appearance of the new platform*

3.1.3 Redo of the wiring

The electrical part of the robot is rather complex. It is composed of many parts which are going to be analysed independently. Each of these parts is collected in the electrical scheme which has been modified to correct some mistakes of numeration and connections. This modified electrical scheme can be found in the *Appendix A*.

The electrical part can be divided in two main parts: the first one involves the safety circuit and power supply of the robot, and the second one, the power supply of the different devices.

The first part to explain is the connections from the batteries and electric current to the wire connector. We must remember that iMoro is a robot that can work with the autonomy of its batteries, or directly connected to the current. To achieve that, a complex design was created to switch automatically when the charger station is connected or not. This idea is schemed in the *Figure 11*.



DRAWN BY VIINIKAINEN MIKKO	TITLE MMMS electrical system concept V. 0.1	DATE 5/30/2012
FILENAME VISIODOCUMENT	DESCRIPTION Power supply	PAGE 1 OF 6

Figure 11 [2]. *Scheme of the idea for the electrical power supply and safety circuit before changes*

The first part of the circuit is the docking station. This part is composed by two smart chargers for the batteries of 48V and 24V, which are directly connected to the batteries. In order to charge the batteries, the rest of the circuit have to be isolated from the batteries, which means the robot should be turned off.

For the situation to be connected to the current, there are two converters from 230V AC to 48V DC and 24V DC respectively, with their correspondent ground connections. The external power supply has been modified and is explained in detail in the *Chapter 3.1.5* with the building of the new charger station.

After that, we can find the switching devices connected to these supply sources. To switch between these two kinds of supply, there is a power switching circuit. The connections of this complex circuit are detailed in the electrical scheme where we can see the using of modular contactors to accomplish this task. After that, the main switch is positioned, which are in charge to allow the functioning of the robot or not.

Finally, in this circuit, we found the safety part. This one is composed by:

- An emergency stop button.
- A safety relay, which automatically open the circuit when there is an overcurrent.
- Four circuit breakers which acts when the rest of components fail, to keep the circuit and the human safe. These four fuses are connected to the different positive part of the wire connector, while the negative is connected without going through these circuit breakers.

On the other hand, the *Figure 11* shows two capacitors that are not in the current scheme. These capacitors have been removed from the actual disposition to reduce noise in the robot signal and other problems caused for having two big capacitors in the robot. These capacitors were placed for the situation of the switching the power supply, to keep the PC on. The problem was that the relay doesn´t switch the power fast enough, and the PC switch off when the change of power supply happened. But after considering the balance between benefits and losses of keeping these capacitors, it has been decided to remove them.

The second part of the electrical circuit involves the power supply of the different devices of the robot. Each part works with a different voltage and current, so they should be separated in the wire connector. To ease the visualization of the connections in the robot, the *Figure 12* was created to show how the wire connector is connected which the different devices.

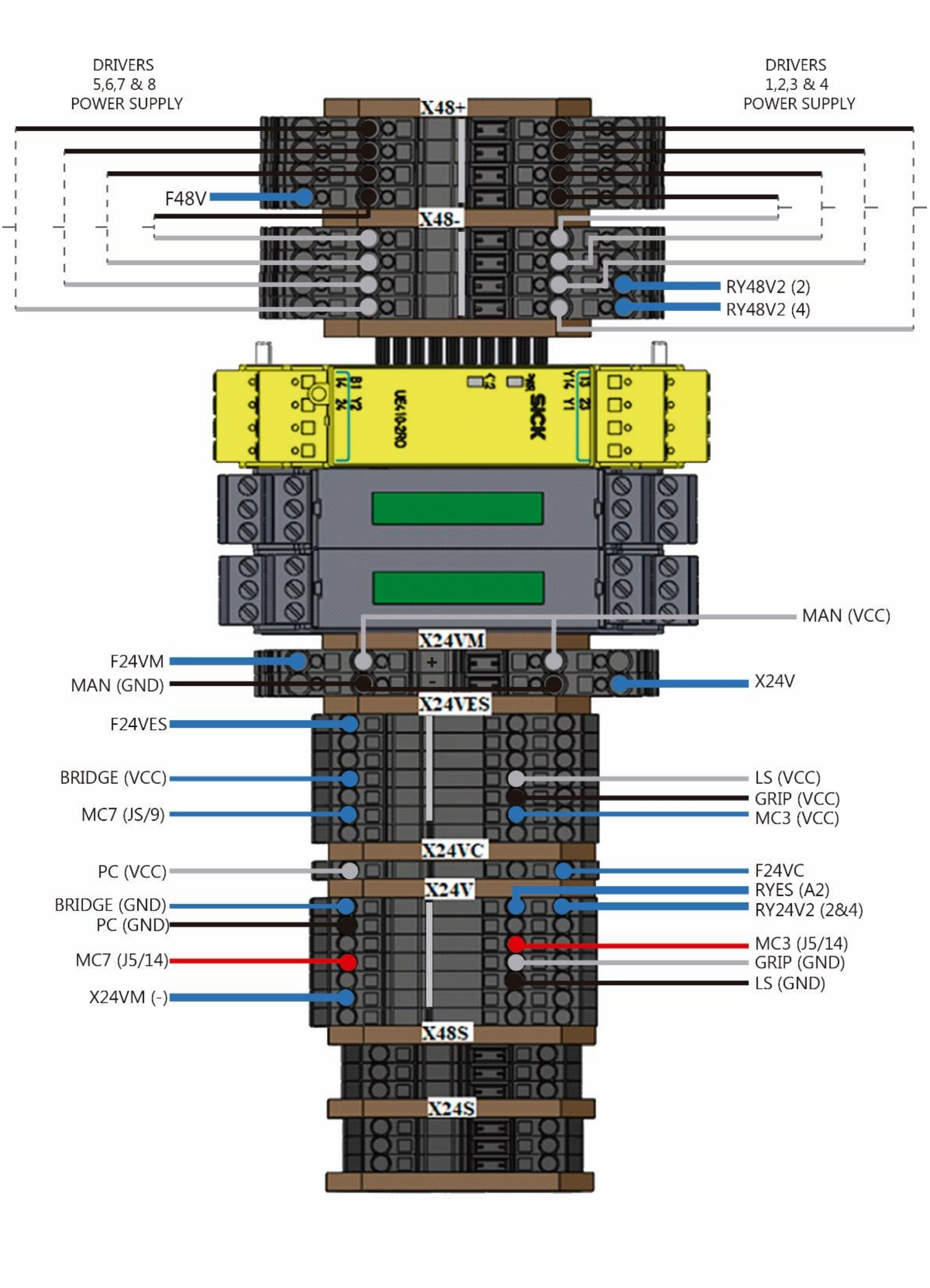


***Figure 12**. Connections on the wire connector*

The part of 48V is reserved for the power supply of the drivers and the motors. Each driver is connected with one wire to the X48+ and other to the X48- in the wire connector. Then, the drivers have connections with the motors to provide them the power supply necessary for their functioning. In this way, the power supply is provided to the motor through the drivers. Both devices use the same amperage (10A), so there is no problem to connect the circuit like that.

Following the *Figure 12*, we can see that the rest of connections are powered by 24V. The reason of having these different sections is the amperage required for each device. The part of X24VM is reserved for the manipulator, which use 20A. Then, the part named X24VES provides current to the rest of the Electrical System, which work with a similar amperage.

Because of its great importance, an extra slack has been added for the PC, named X24VC, to have the possibility to program the computer without having the rest of the robot switched on. This is possible by connecting only the fuse that goes to the power supply slack of the computer.

Finally, the part of X24V has been introduced to connect all the negative wires of 24V, because the robot works in DC. As it has been explained before, each part has their own circuit breaker, which allows the researcher to test the parts independently, without giving power to the entire robot.

On the other hand, the *Figure 12* shows two ranks that have no connections: X24S and X48S. These ranks where created to have a voltage division with some resistances but, due to the removing of the device that used this voltage, they are now useless. However, they hadn’t being removed of the circuit for future improvements and investigations.

To finalize this chapter, the series of modifications made in the electrical scheme of the *Appendix A* are going to be detailed and explained. This document is composed of 11 pages, 10 for the electrical connections and one for the CAN bus system. The *Table 12* summarizes all these changes.

Page number	**Modification**	**Explanation**
1	Creation of an external power station and introduction of the smart charger connections for the batteries	Explained in detail on the *Chapter 3.1.5*
2	No modification	The switching system is well designed and works properly
3	Removing of both capacitors	As it was explained before, for reducing a series of problems generated by two big capacitors
4	Removing the connections of X24S and X48S	As in was explained before, the robot is not using these ranks now, but they are kept in the circuit for future developments
5	Renumbering of the drivers	Renumbering of the drivers according of the numeration in the page of the CAN bus system
6	Renumbering of the drivers	Renumbering of the drivers according of the numeration in the page of the CAN bus system
7	Introduction of the Bridge	Introduction of the connections that have the device: Moxa Bridge
8	Explanation of the connections of the DIO and AIO	These devices are kept in the robot for future investigation but, as they are not in use now, they are isolated from the rest of the circuit. Instead of that, the connections are kept in the electrical scheme to show the future developers how to do it.
9	Explanation of the connections of ENC1, ENC2 and CAM	These devices are not actually in the robot, but as they are a future idea for developing, the connections are introduced in the electrical scheme to show the future developers how to do it.
10	No modifications	The manipulator is well connected
11	Removing of this page	This page has been removed because it belongs to the CAN bus system, which should be explained in a different scheme. This page is included in the *Appendix B*. Their changes are included in this appendix also.

Table 12*. Electrical scheme modifications*

3.1.4 External case

The external case is an important part in a robot. It protects the internal circuits from shocks and provides an appearance more robust to the robot. It also serves to protect the internal parts from misconnections and wrong modifications caused by the unknowledge of the user and keep the necessary devices accessible to the user. IMoro robot did not have any, so in this chapter, the design and building of the external case will be explained in detail.

First, it was necessary to decide which parts would be visible and which would not. The new platform was designed to hold the main components of the robot which should be accessible to the user, so this part would be out of the case. The rest of the robot would be covered by a case which should be easy removed when someone wants to make modifications inside the robot.

To achieve that goal, the case was designed as a two-part case, which could be opened as a wardrobe door. Each part of the case will be attached to one side of the robot with a mechanism of hinges to ease the opening and closing. The *Figure 13* shows the case on the main platform and the *Figure 14* illustrates how the mechanism works.

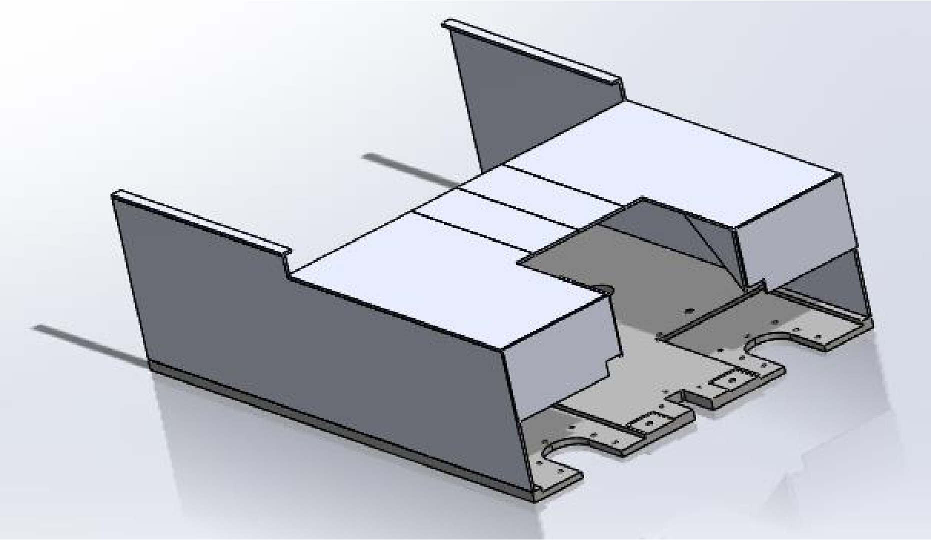


Figure 13. *External case assembled only in the main platform*

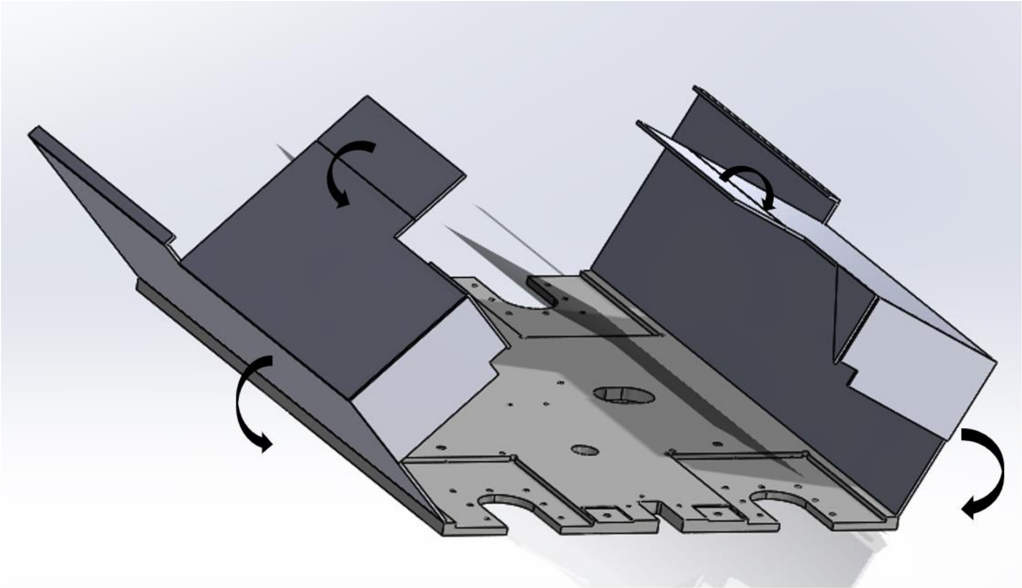


Figure 14. *Mechanism of opening for the external case*

As we can see in the *Figure 14*, there is a small subdivision on each part of the case. This design is introduced because the manipulator of the robot is placed in a plate that can be moved at 45 degrees angle from the horizontal position clockwise. This movement is associated to the movement of its respective wires, which will collide in this situation. To avoid it, a second wardrobe door mechanism was implemented to be able to open only this part of the case and work with the manipulator in this angle.

The *Figure 15* shows how the case fits over the robot, which in this case, has the base of the manipulator in angle.

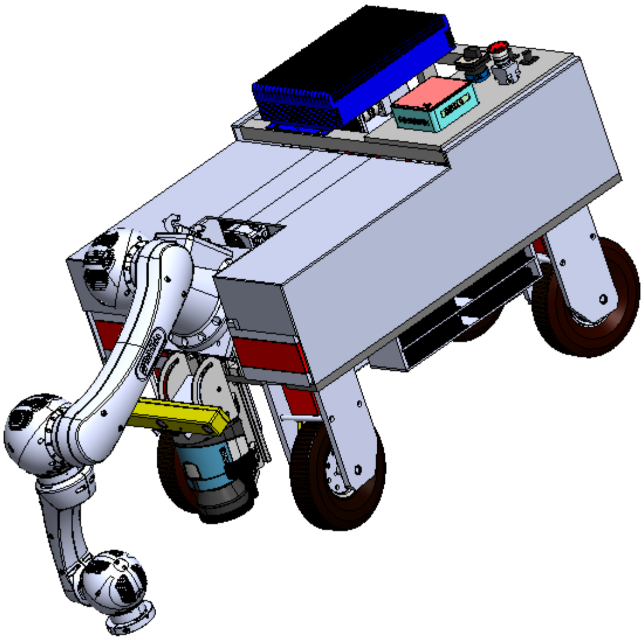


Figure 15. *CAD model of iMoro with the external case assembled*

For the building of this case a series of steps were followed. First, there was to think about the design and take the measurements on the robot for the drawing of the model. Then, there was to consider about which the material of the case will be and think about the method of building.

The material chosen for the building of the case were (Plywood 4mm), and the place of building was the FabLab. Due to the use of this type of material and method of building (by laser cutting of the wood, and then assembling it with glue and mechanic elements), the case couldn´t be built at once.

The process for building the case in the lab where the following:

1. Model each of these parts in Adobe Illustrator, with a `Zipper´ mechanism to ease the assemble of the parts.
2. Cut the parts with wood in FabLab with the Laser cutting machine.
3. Create the holes for the carpenter´s square, which are going to be the mechanism that keeps united the parts.
4. Varnish the parts
5. Include some pieces of a rigid material to correct the natural curvature of the wood in long pieces. These pieces can be seen on the *Figure 17*, in the internal part of the case.

Finally, when the case has been built, there was to assemble it to the robot. For this, as it was explained before, a series of hinges were used to ensure the easy opening of the case. Also, some mechanics elements where added to ensure the endurance and the strength of the case in each situation. The *Figure 16* shows the real case assembled over iMoro robot, and the *Figure 17* shows the case open.

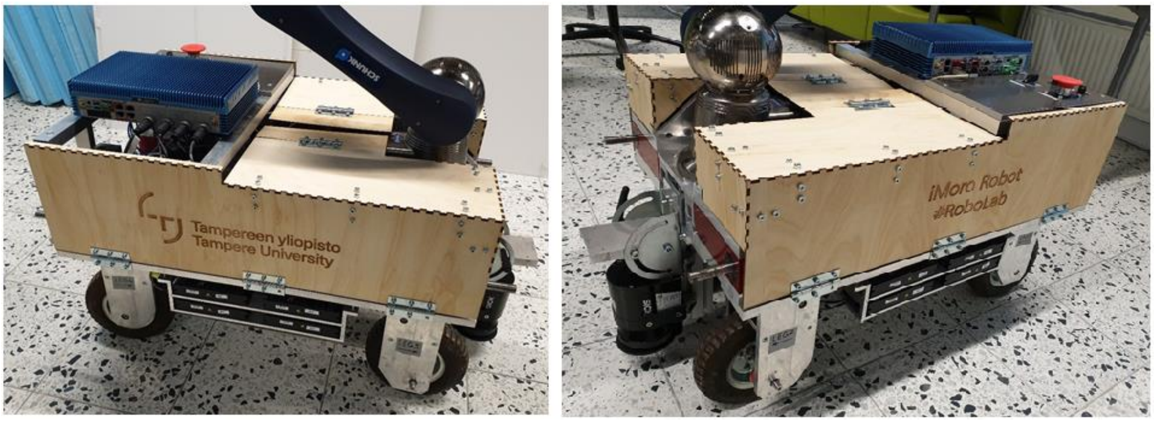


Figure 16. *Final appearance of the external case*

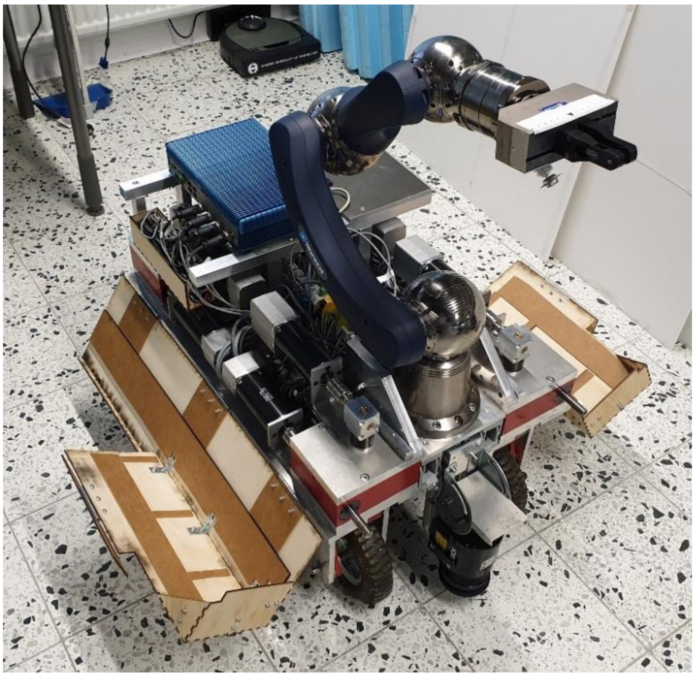


***Figure 17**. Case of opening of the external case*

3.1.5 Charger Station

The final chapter of the hardware part is the charger station. iMoro have a complex system that switch between the supply of the batteries and the supply of the charger station automatically, so when the batteries run out of power, the robot can keep functioning. For that, a good charger station is required to cover the case where the robot work connected to the current.

IMoro work with DC current, so a couple of converters from AC/DC are required in this station. These converters chosen are *TRIO-PS/1AC/48DC/10* and *Powernet ADC5423-3*, which satisfy the need of a supply of 24V and 48V respectively. When the thesis started, there weren’t even the necessary converters, so it was necessary to find them in the university.

After having both converters, there was to design how it was going to be the charger station. This charger station should be able to hold both converters and provide a good air flow to avoid overheating; have a system to roll the wire, which was going to be 8 m long for each converter in the part of DC and 2 meters long in the part of AC; and include a Switch button to be able to disconnect the power before unplug the wire from the robot, to avoid sparks.

With these required specifications, the charger station was designed and built. The building was also made in FabLab, because the material was the same as the external case, so the process followed were the same. The *Figure 18* shows how the external case of the charger station was designed.

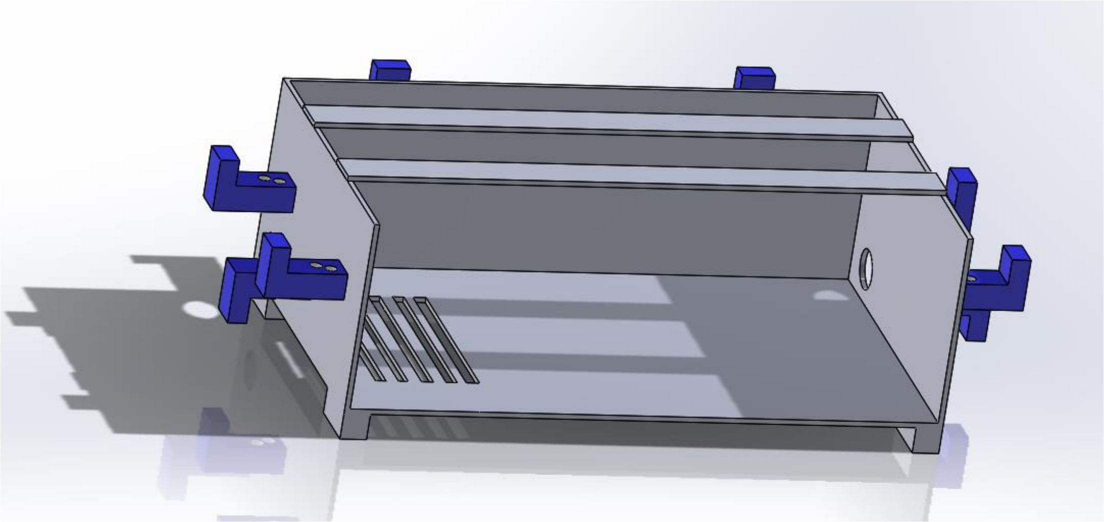


***Figure 18**. CAD model of the external case of the charger station*

All the parts of the station were built in wood except the holders for the wire, which were 3D printed with PLA in FabLab. That decision was made because of the need of a bigger strength in these parts and the fact that 4 mm thick wood wouldn´t provide it. After build the case, there was needed to attach the converters and the circuit breaker as it showed in the *Figure 19*.

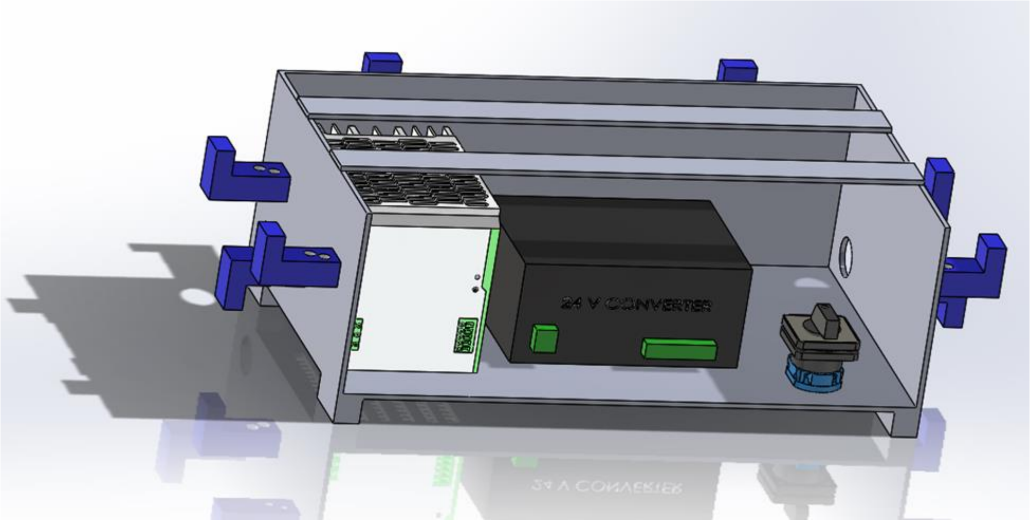


***Figure 19.** CAD model of the charger station complete*

The converter of 48V has the air vents on the top and the bottom surface, so this is the reason that the floor has these holes, to dissipate the heat. Conversely, the 24V converters have the air vents in the top and the adjacent faces, so it was no necessary to make holes on the case.

Finally, a pair of guides for wires was included to provide some order inside the charger station and then, connect the wires. These wires had to be replaced by a new and longer model, to increase the working range of the robot when it is connected. To do that, it was necessary to solder the plug-in connections in the FabLab and adjust the length of each wire. To make robust and clean the workspace, both wires were merged in an only one with a cover. *Figure 20* shows the final appearance of the Charger station.

Figure 20. *Final appearance of the charger station*

3.2 Software

The software part of a robot are the instructions that control what a computer does [4]. For iMoro, the software part is mainly ruled by Simulink Real-Time, but a series of steps need to be done before starting to use it. In this chapter, it´s going to be explained how to initialize the robot and pare it with your personal computer. There are three steps in this paring process.

3.2.1 Configuration of your personal computer

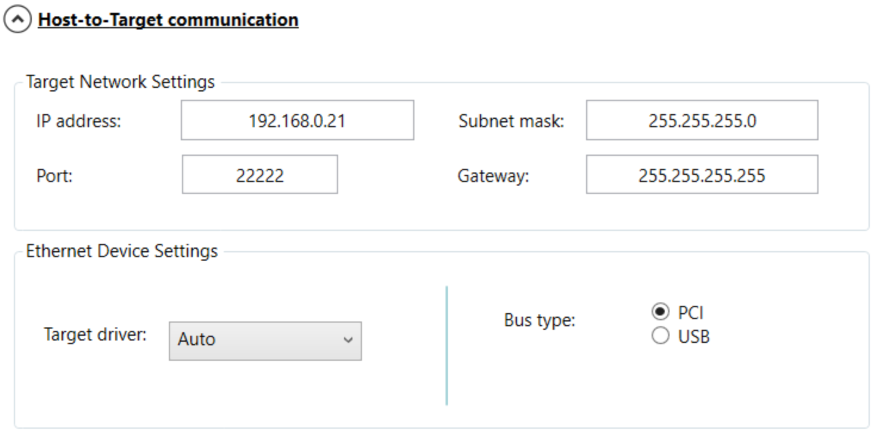
The first part of the synchronization between devices begins with your personal computer. For that, the program Matlab and the extension Simulink Real-Time must be previously installed in your PC. Then, these steps need to be followed.

First, open Matlab and import the folder `Platform´, which have all the required files to use the robot. When this has been done, introduce in the command window `slrtexplr´ to open Simulink Real-Time on the computer. Now, that Simulink Real-Time has been opened, add a new target, which in this case is the computer of the robot: Tank 720. These steps are illustrated in the *Figure 21*.

```
>> slrtexplr
```

***Figure 21**. Process to run Simulink Real-Time*

Then, open the menu of properties to set the parameters of the connections. There are three parts that need to be filled. The first part is Host-to-target communication, here define the IP that Tank 720 has, which in this case is 192.168.0.21, so Simulink Real-Time can identify the target where to send the information. The other parameters should be set as the *Figure 22* shows.



Host-to-Target communication

Target Network Settings

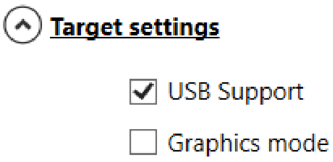
IP address: 192.168.0.21
Subnet mask: 255.255.255.0
Port: 22222
Gateway: 255.255.255.255

Ethernet Device Settings

Target driver: Auto
Bus type: ● PCI ○ USB

***Figure 22.** Settings of the Host-to-targert communication window*

The second part concerns the target settings. Here selects only USB support, which means that we extract the information for the starting of the PC. *Figure 23* shows the requested configuration.



Target settings

☑ USB Support

☐ Graphics mode

Figure 23. *Settings of the Target settings window*

The third part is the boot configuration, for this is important to know that the computer Tank 720 use the PCI Bus type. For running this settings Simulink offer different ways (Network, CD…) in this case it has been made a bootable USB, so now it is needed to create this USB.

With the button `create boot disk´ we create the necessary files for running the Simulink program in the Tank 720, *Figure 24* shows how to do it.

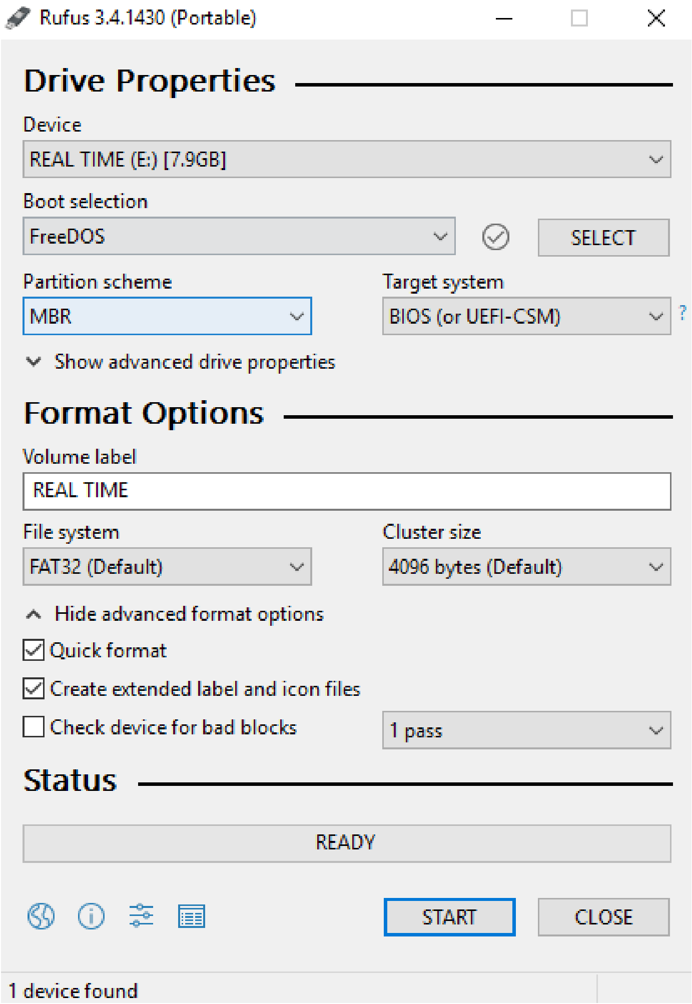
Boot configuration

Boot mode: DOS Loader

Create boot disk

Figure 24. *Settings of the Boot configuration window*

After doing that, it is necessary to create a bootable USB with free-DOS and then copy inside the files created by Simulink. In this case we use the RUFUS program and we set the characteristics showed in the *Figure 25* for the creation of bootable files. The program to create a bootable USB don´t have to be that exactly that, but in this case, it has been created with RUFUS program.



Rufus 3.4.1430 (Portable)

Drive Properties

Device
REAL TIME (E:) [7.9GB]

Boot selection
FreeDOS

Partition scheme
MBR

Target system
BIOS (or UEFI-CSM)

Show advanced drive properties

Format Options

Volume label
REAL TIME

File system
FAT32 (Default)

Cluster size
4096 bytes (Default)

Hide advanced format options

☑ Quick format
☑ Create extended label and icon files
☐ Check device for bad blocks
1 pass

Status

READY

1 device found

Figure 25. *Settings of the RUFUS program to create the bootable USB*

3.2.2 Configuration of the router

In this case, the router used is MOXA AirWorks 1100 Series, whose configuration can be modified entering in its IP direction. These are the steps that have to be followed to configure the router.

First, connect your personal computer to MOXA WIFI, which should appear the robot´s electrical system is turned on. Then, enter on internet and introduce the address 192.168.0.0, which is the IP address of the device. To access the page you must enter the user name: *admin*, and the password: *root*. There, it is possible to configure the connexion of your personal computer with the WIFI, so you can work with the computer without being connected by a wire. *Figure 26* shows the general menu that appears when you access to the webpage.

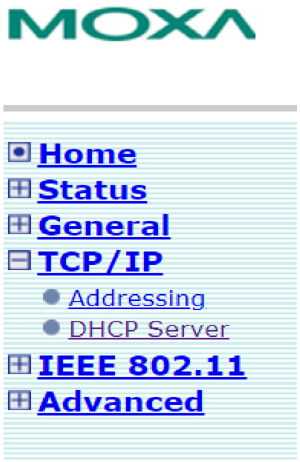
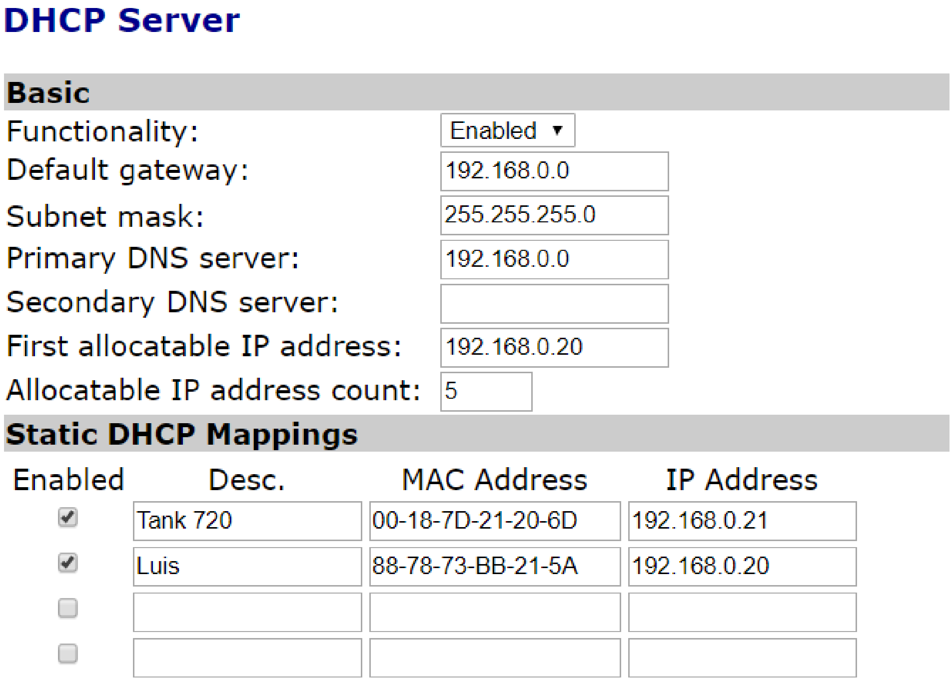


Figure 26. *Principal menu of MOXA router webpage*

Now it is time to configure the protocol TCP/IP. For doing that, open DHCP server. When this is opened, a menu like one in *Figure 27* can be seen on the screen. Here, it is possible to set the initial parameters of the WIFI connections, and reserve some IP directions, so when your personal computer is connected to the WIFI, this IP address is assigned to it instead a random IP. That is specially important for setting the Tank 720 IP address, because if we don´t do it, Simulink couldn´t connect with it. As we can see in the *Figure 27*, in have the same IP as Simulink Real-Time: 192.168.0.0.



DHCP Server

Basic

Functionality:	Enabled ▾
Default gateway:	192.168.0.0
Subnet mask:	255.255.255.0
Primary DNS server:	192.168.0.0
Secondary DNS server:	
First allocatable IP address:	192.168.0.20
Allocatable IP address count:	5

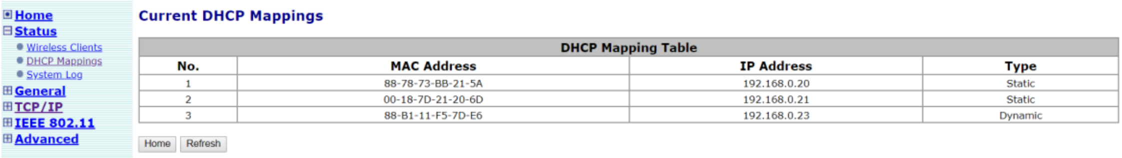
Static DHCP Mappings

Enabled	Desc.	MAC Address	IP Address
☑	Tank 720	00-18-7D-21-20-6D	192.168.0.21
☑	Luis	88-78-73-BB-21-5A	192.168.0.20
☐			
☐			

Figure 27. *DHCP server window of MOXA router webpage*

The IP address of the target computer should match with the Host to target configuration defined above. To do this, the most reliable option is to configure a MAC filter on the router for the IP protocol. With that, every time that MOXA detect this MAC address, it automatically assigns that IP address, which is reserved for the TANK 720.

Now, that the router is set, it is possible to see which devices are actually connected to the router in the link \`Status´ and \`DHCP Mappings´, as the *Figure 28* shows.



Current DHCP Mappings

DHCP Mapping Table

No.	MAC Address	IP Address	Type
1	88-78-73-BB-21-5A	192.168.0.20	Static
2	00-18-7D-21-20-6D	192.168.0.21	Static
3	88-B1-11-F5-7D-E6	192.168.0.23	Dynamic

Figure 28. *Current DHCP Mappings window of MOXA router webpage*

3.2.3 Configuration of Tank 720

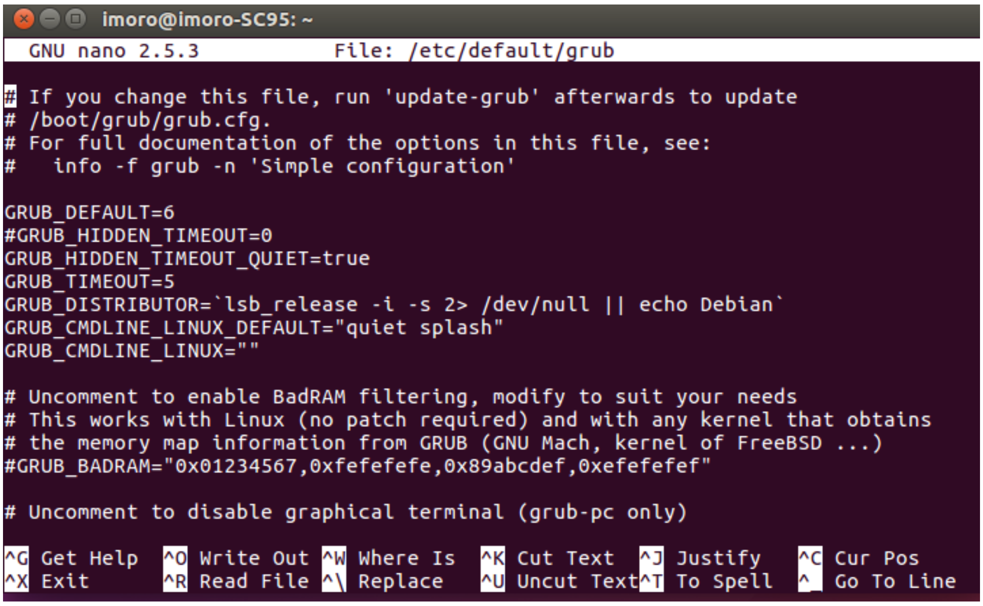
The PC that uses the robot is iEi TANK-720-Q67. It had installed Linux, but to start running the program Simulink Real-Time, it has been created a bootable USB in the previous steps.

First, connect the USB and turn on the Tank 720. Then, wait until it finishes it starting. If you want, it is possible to connect a screen to the VGA connection and interact with the starting. If not, it will be automatically started. The Tank 720 has been configurated to make the starting automatically.

For some reason, the BIOS of this computer do not save any changes on the boot configuration. This problem can be solved editing the Groub of Ubuntu. First of all, it is important to know that the password of iMoro for starting the program or modifing files is: ***R0b0lab*** (Ubuntu password).

To configure this automatical initiation there are to start the onboard computer and run the Ubuntu operative system. Then, from the searcher window, look for the window \`*Terminal*´ and write : **$ sudo nano /etc/default/grub** .

The part sudo gives you root permissions, the nano part open a text editor and the rest is the localization of the file that is neccesary to modify. When the order is introduced, the window ilustrated in the *Figure 29* is opened.



```
  GNU nano 2.5.3             File: /etc/default/grub

# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'

GRUB_DEFAULT=6
#GRUB_HIDDEN_TIMEOUT=0
GRUB_HIDDEN_TIMEOUT_QUIET=true
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""

# Uncomment to enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel that obtains
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"

# Uncomment to disable graphical terminal (grub-pc only)

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

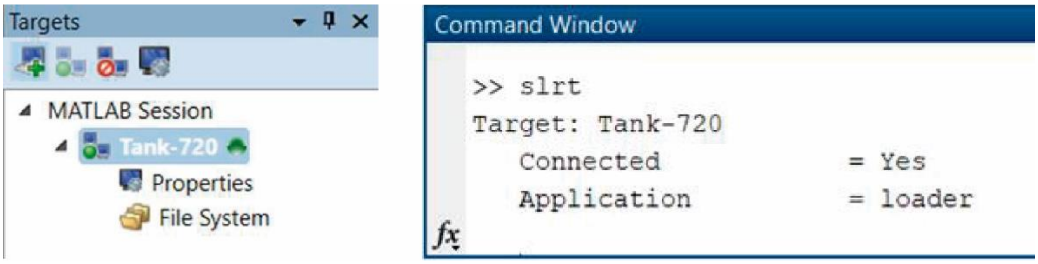
Figure 29. *Command window to introduce changes in the running system*

In that text, the parameter GRUB_DEFAULT= 0 can be changed. This define default booting option. In this case USB boot is the option 6 (the position in the initial booting menu, GRUB) so that line should be changed to GRUB_DEFAULT= 6. This change makes that the PC automatically starts in the GRUB 6, which means the bootable USB.

After doing that, the changes have to be saved in a special way. When the line is changed, get out of this scrub window and then, in the terminal window, write the following text to apply changes: **$ sudo update-grub && sudo update-grub2**

Now, that the previous configuration is done, open Matlab and run the file RunMeFirst.m. Then go to Simulink Real-Time and press the button connect. If everything is correct the part of connections will be done.

To check that the connections are working, go to the command window of Matlab and introduce the command `*slrt*´ to show the actual status of Simulink Real-Time. *Figure 30* shows what should appear in Simulink Real-Time and in the command window of Matlab when is connected correctly.



```
>> slrt
Target: Tank-720
   Connected            = Yes
   Application          = loader
```

Figure 30. *Comprobation of the right connection*

4. FUTURE IMPLEMENTATIONS

A series of research has been done before this thesis, and many will be done in the future. This chapter summarizes ideas and work that couldn´t be done in this thesis. Some of this work need to be done immediately, but other are just ideas for improving the performance of the robot in the future.

4.1 Can Bus scheme

The robot uses the CAN bus system to do its communication between devices. The lack of a CAN bus scheme that clarifies the connections complicate the work on the robot. Because of that, the elaboration of a CAN bus scheme is a primordial task to be accomplished.

Moreover, there is some work that have been done in this part. There is a CAN bus scheme of the connections of the drivers, but it is outdated and there are many connections that are currently in the robot that are not reflected in that scheme. This scheme is attached in the *Appendix B* with some annotations of the actual state of those connections.

4.2 Improvement of Charger Station

The new Charger Station is a perfectly operative power supplier which can work for years. But, even thought that fact, there were some improvement that can be introduced on it which couldn´t be done in this thesis because of the lack of time.

The wire has a simple system to be picked up: roll them over the sides of the charger station, but it wasn´t the first idea. The idea was to make a system like the vacuum cleaner has, which roll automatically the wire by pressing a button. This improvement, plus a big pole over the robot to guide the wire and avoid getting meshed, will provide the autonomous movement that iMoro needs. Actually, you have to follow the robot, picking up the wire, so the robot doesn´t pass over them.

Other option instead of the vacuum cleaner mechanism is a motor connecter to a wire roll. This motor would be directly feed by the 24V converter and will transmit the rotatory movement to a coil which automatically pick up the wire.

On the other hand, other improvement that were considered but couldn´t be implemented was made a close case. The main reason for not doing this idea is that the converters generate a big heat that need to be dissipated. For this reason, the case of the charger station is open, even it makes it weak. Some reinforcement was added to the case to make it strong, but a close disposition would make it stronger.

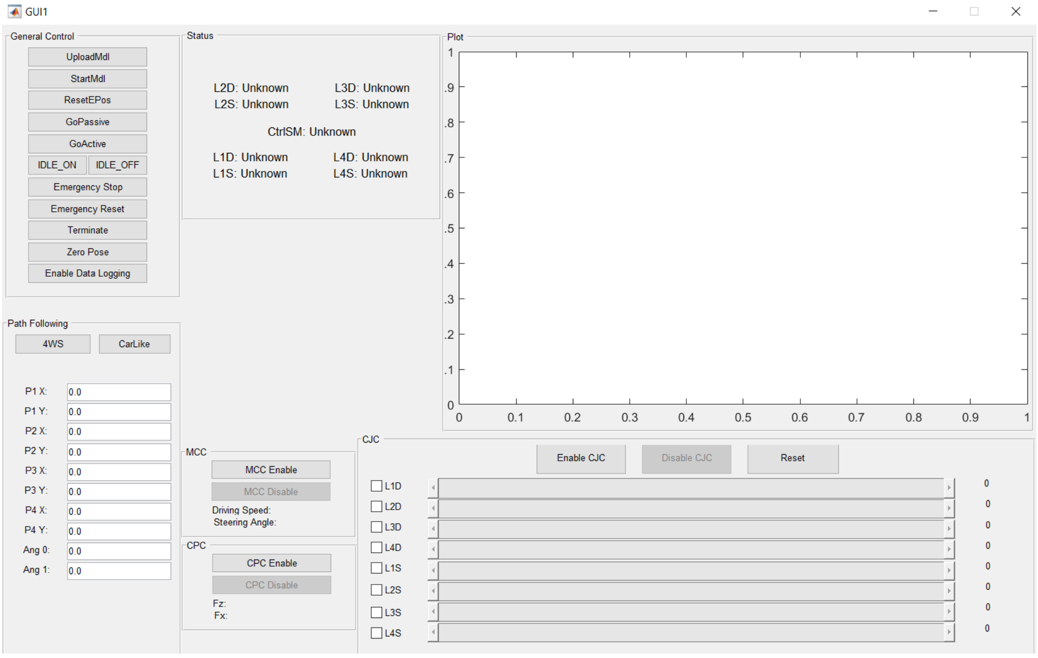
Coming back to the problem of overheating in a closed case, the solution for that could be introduce a couple of air extractors that dissipates the heat. This mechanism is actually used by the laptops and could be a good solution to this problem. If you want to go beyond that, a thermic sensor could be added inside of the case to detect when its needed to activate the air extractors so that they are not in constant functioning.

4.3 Continue with software part

This part is extremely important for the functioning of the robot. Without the instructions that rule its performance, the robot would be a fixed group of devices, but in this case, it has a complex group of programs to control its functioning as well as determine their performance.

This chapter is not included in the *Research and implementations* part because it isn´t finish due to the lack of time. But, although it is no complete, all the advances done in this part are going to be explained, so the future researchers could start from there if they wanted.

Once the paring between the onboard computer and your personal laptop have been done, open on Matlab the file named **GUI1.fig**. This file creates an interface for use each of the motors of the robot. It allows the user to be able to visualize the variations and the orders on each of them. The *Figure 31* shows the controller window of the program.



L2D: Unknown L3D: Unknown
L2S: Unknown L3S: Unknown
CtrlSM: Unknown
L1D: Unknown L4D: Unknown
L1S: Unknown L4S: Unknown

P1 X: 0.0
P1 Y: 0.0
P2 X: 0.0
P2 Y: 0.0
P3 X: 0.0
P3 Y: 0.0
P4 X: 0.0
P4 Y: 0.0
Ang 0: 0.0
Ang 1: 0.0

Driving Speed:
Steering Angle:

Fz:
Fx:

Figure 31. *GUI1 program window*

After that, open again Matlab and run the program **TxMP_Commands.slx**. This program aims to synchronize the clock of the onboard computer and the one on your personal laptop. The *Figure 32* shows the command window that appears when you introduce this command on Matlab. As this file is already created, there is only to press the button play and that's all.

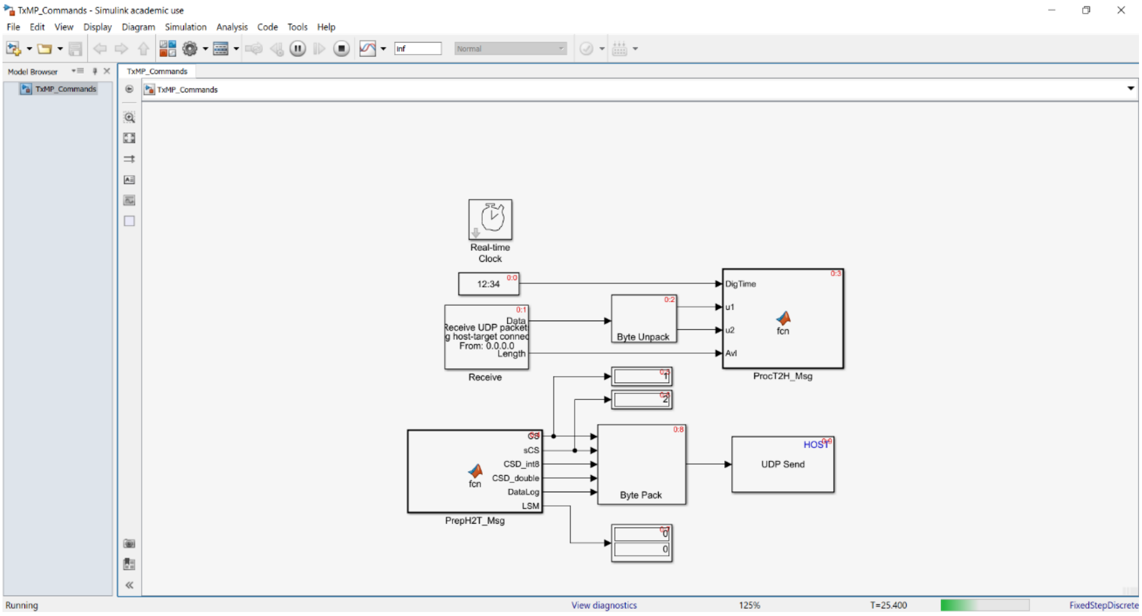


Figure 32. *TxMP_Commands window*

Then, open the Simulink model of the robot. It is highly complex and has many sub models that must be understood to be able to introduce changes and improvements properly. The first thing to do is open the file **R3T1_2015.slx** and run it. The *Figure 33* shows the window that is opened.

Here, to import the model into the onboard computer, you have to follow these steps:

1. Press *Parameters configuration* and goes to the *Code generation > Simulink Real-Time options*
2. Set the parameters as the *Figure 34* shows
3. Apply changes and run the model

After all these steps, the model should have been uploaded to the onboard computer, but here is where the problems come. Simulink gives many problems that we couldn´t solve yet and would have to be solved by the future investigators.

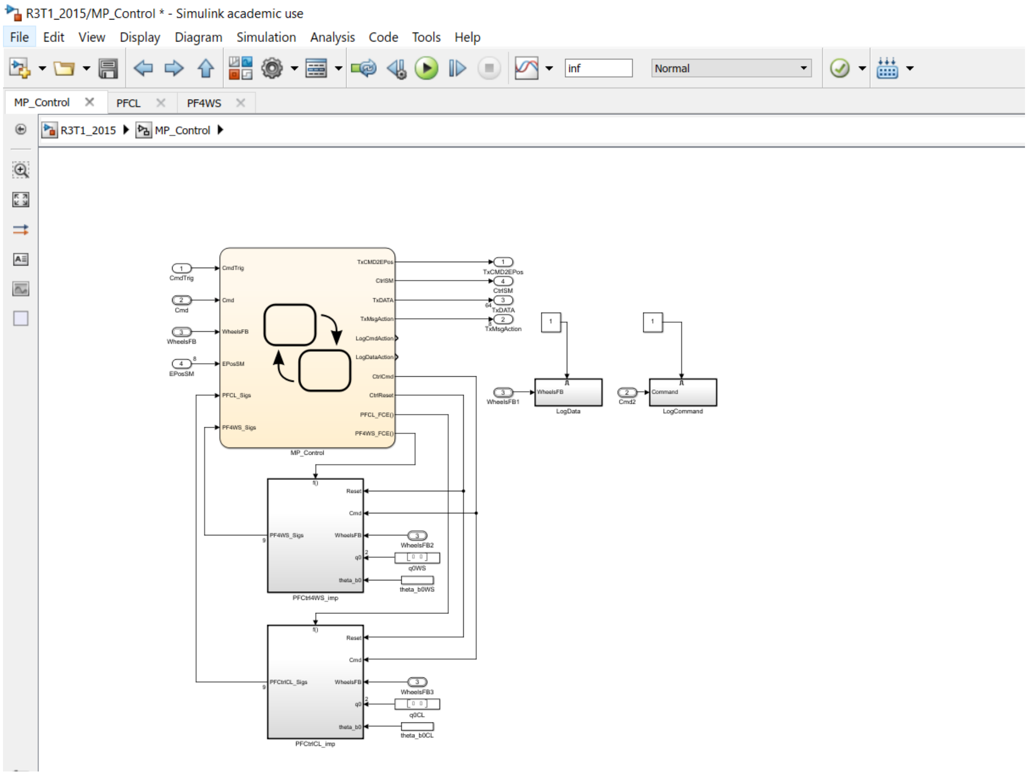
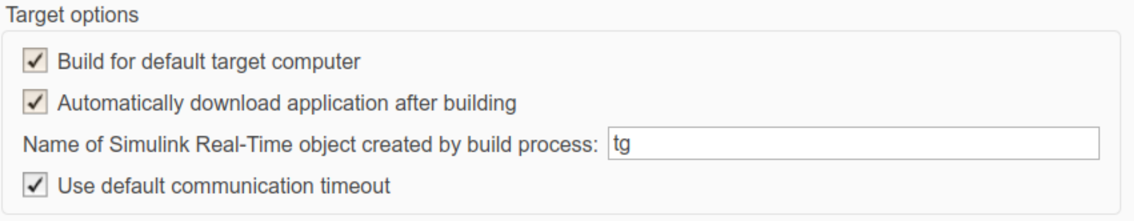


Figure 33. *R3T1_2015 window, Simulink model*


Target options

☑ Build for default target computer

☑ Automatically download application after building

Name of Simulink Real-Time object created by build process: tg

☑ Use default communication timeout

Figure 34. *Parameters on Simulink Real-Time window of the Parameters window*

In advance, some links to Simulink tutorials and manuals are going to be provided, so the investigators can work with them. The idea is to create a simple Simulink model, charge it in the onboard computer and check if it is working on your laptop; before starting with the complex model. If that's works, go step by step until find the problem of synchronization and find out how to solve it. The *Table 13* shows these useful links.

Description	Link
Set Up and Configure Simulink Real-Time	https://se.mathworks.com/help/xpc/gs/set-up-and-configure-xpc-target.html
Create and Run Real-Time Application from Simulink Model	https://se.mathworks.com/help/xpc/gs/run-a-real-time-application.html
Apply Simulink Real-Time Model Template to Create Real-Time Application	https://se.mathworks.com/help/xpc/gs/apply-xpc-model-template-to-create-real-time-app.html
Real-Time Simulation and Testing	https://se.mathworks.com/help/xpc/gs/real-time-simulation-and-testing.html
Configure and Control a Real-Time Application	https://se.mathworks.com/help/xpc/gs/interact-with-a-real-time-application.html
PDF Documentation for Simulink Real-Time	https://se.mathworks.com/help/pdf_doc/xpc/index.html?s_tid=mwa_osa_a

Table 13. *Links to manuals and guides*

5. RESULTS AND CONCLUSIONS

IMoro was a robot with a high development potential thanks to its multitude of options, but due to the age of the robot and the lack of updates in the hardware part, it was quite useless. This thesis has strongly reinforced the hardware part, leaving the robot ready for programming and using.

The chaos in the main platform was one of the main problems. When the thesis started it was impossible to check the wire connections or play with some devices because they were placed in the platform with no order. Thanks to the changes introduced on (see *Chapter 3.1.1* and *1.1.2),* this problem can be considered solved. It´s true that still can be a little messy, but this is as much ordered as it can be with all the devices that it has and its complex electrical circuit.

Other problem was the outdated of the electrical scheme together with the lack of clearness on the electrical connections and the goal of the design in the case of the electrical circuit. The solution to that has been achieved and explained at *Chapter 3.1.3*, where detailed figures like the wire connector, explanation of the parts of the circuit and the updating of the electrical scheme has clarified a lot the functioning of the electrical part of the robot.

On the other hand, the lack of external case in the robot was a problem for the safety of the human and for the robot but also it was a problem of appearance. The case built during this thesis provide a robust state to the robot, increase the safety in the case of accidental interaction with the electrical part, and ease the accessibility to the main platform thanks to its hinges mechanism. The reason of the design and its process is detailed in the *Chapter 3.1.4*.

Related to the last point, there was a problem with the lack of a charger station. In this case there even wasn´t a primary design of it, only two converters. This problem is related to the last one because it aims to reach the same objectives: improve human and robot safety, achieve a better visualization and ease the changes on it. The *Chapter 3.1.5* explains how these goals have been achieved thanks to the creation of a charger station. Now, the robot has a charged station appropriate for its using, which will obtain the required performance.

Finally, other of the mains problems in the robot was the lack of a user´s manual for the people who want to work with iMoro. The *Chapter 3.2* starts with the creation of this manual to ease this work. In this part, only the guide of how pare your personal computer with the onboard computer has been accomplished. But, instead of that, the starting of the study about the actual programs, models and how to use them has been introduced in the part of Future implementations, so the future investigators can continue from there the creation of the manual.

To summarize everything that this thesis has involve let's make a reflection of the engineering fields that have participated in it. First of all, without the electric background, this thesis would be impossible, because one of the most complex parts has been the updating of the electrical system.

The mechanical engineering has also had a huge importance due the redistribution of the devices, creation of an external case, creation of a second platform, creation of a charger station, model the parts in CAD and the using of drilling machines as well as other type of machinery to achieve all these tasks.

Finally, informatics and automation knowledge has been really useful in order to understand the functioning of the robot and the starting of working on its programming.

REFERENCES

[1] Wikipedia, the free encyclopaedia – Machine, updated 16 March 2019
Available: https://en.wikipedia.org/wiki/Machine

[2] Mohammad M. Aref, Reza Oftadeh, Mikko Viinikainen, Janne Koivumäki – Conceptual design document for `Modular Mobile Manipulator System´, 56 p.

[3] Eemeli Anttila – Human-Machine interface design for an omnidirectional mobile manipulator, August 2018, 34p.

[4] Cambridge Dictionary – Software
Available: https://dictionary.cambridge.org/dictionary/english/software