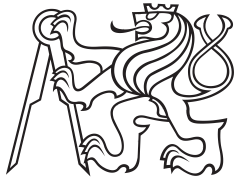


Master's Thesis



Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

Design and Construction of an Omnidirectional Base for Student Robotics Platform

Educational robotic platform specially designed for
university students

Václav Veselý

Supervisor: Mgr. Martin Pecka, Ph.D
Supervisor–specialist: Ing. Bedřich Himmel
Field of study: Cybernetics and Robotics
August 2023

Acknowledgements

I want to express my gratitude to my supervisor, Mgr. Martin Pecka, Ph.D., for his invaluable guidance, unwavering support, and profound expertise throughout this project. His mentorship has been instrumental in shaping the development of the educational robotic platform.

I am also deeply thankful to Ing. Bedřich Himmel, my supervisor-specialist, for his valuable insights and contributions, especially in the construction aspects of the platform. His expertise and assistance have been pivotal in ensuring the success and effectiveness of the design and construction process.

The combination of their knowledge, encouragement, and dedication has significantly enriched this work's quality, and I genuinely appreciate their contributions to this project.

Declaration

I declare that the presented work was developed independently, and I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

In Prague, 14. August 2023

.....

Abstract

The SRobot is a comprehensive indoor mobile robotic platform, meticulously designed as an educational tool for university students, covering fundamental robotics concepts encompassing both hardware and software aspects.

Distinct advantages of the SRobot include its modular design, extended operational lifetime, and innovative capacitive tactile bumper that grants true omnidirectional capabilities. The integration of BLDC motors ensures precision, smoother movement, and energy efficiency. Remarkably cost-effective, the SRobot stands out among comparable options.

The construction process is meticulously detailed, providing a replicable guide, including component selection, functionality, and battery management. The PCB design, integrated components, and electronics are thoroughly described, offering a clear assembly blueprint.

The firmware section offers insights into STM32 microcontrollers, code upload instructions, robot connection diagrams, BLDC motor control, PID controllers, various interrupts provided by ISR, communication protocols, and protective features. The platform's flexibility with connectors for diverse modules is highlighted.

The software section explores ROS packages, topics, messages, the startup sequence, robot model creation, odometry implementation, Gazebo simulation, and RVIZ application, providing a comprehensive understanding of the robot's functionality.

The platform's educational suitability is emphasized, showcasing a range of use cases, a dedicated robotics curriculum, and example lessons covering firmware,

SLAM algorithms, and more, making the SRobot an ideal educational platform.

The conclusion points out the SRobot to be an exemplary indoor mobile robotic platform that effectively fulfills its master's thesis assignment. Its modular design, hardware capabilities, sophisticated firmware, and versatile software make it a valuable educational tool, ready to teach essential robotics concepts to university students.

Keywords: SRobot, Indoor mobile robotic platform, Omnidirectional robot, Modular design, Capacitive tactile bumper, BLDC motors, STM32, Field-oriented control, Interrupt service routine, Touch Sensing Controller, LiDAR, ROS packages, Gazebo simulation, SLAM algorithms, Educational robotic platform, Kinematics and dynamics of omnidirectional robots

Supervisor: Mgr. Martin Pecka, Ph.D
Department of Cybernetics,
Resslova 307/9,
Praha 2

Abstrakt

SRobot je komplexní mobilní robotická platforma určená jako výukový nástroj pro studenty univerzit, pokrývající základní hardwarové a softwarové koncepty robotiky.

Výraznou výhodou této robotické platformy oproti ostatním robotům je její modulární design, vyšší výdrž a inovativní kapacitní nárazník, který poskytuje opravdovou všesměrovou možnost pohybu. Integrace BLDC motorů zajišťuje přesný a plynulý pohyb s výrazně vyšší účinností. SRobot je navíc cenově srovnatelný v porovnání s ostatními roboty určenými pro vzdělávací účely.

Stavební proces robota je pečlivě popsán a poskytuje možnost robota snadno zreplikovat. Zahrnuje kompletní výběr komponent, jejich funkčnost a jejich integraci v robotovi. Návrh desek plošných spojů s jednotlivými elektronickými částmi je podrobně popsán a nabízí jasný návod k vytvoření.

Ve firmwarové části jsou popsány jednotlivé STM32 mikrokontroléry integrované na deskách, s kompletními instrukcemi pro nahrání jejich kódu. Tato část navíc zobrazuje diagram jednotlivých komunikačních spojení robota, dále zmiňuje řízení BLDC motorů, regulátor PID, ISR, komunikační protokoly a ochranné prvky desek. Dále je v práci zmíněna možnost robota rozšířit o další různé moduly.

Softwarová část popisuje ROS balíčky implementované v robotu s jednotlivými tématy a zprávami, které balíčky vytváří. Dále zmiňuje startovací sekvenci robota, vytváření jeho modelu, implementaci odometrie, Gazebo simulaci a vizualizaci pomocí RViz aplikace.

V poslední části je důraz kladen na

vzdělávací vhodnost platformy. Ukazuje se zde její použitelnost a nasaditelnost při výuce robotiky. K tomuto účelu je vytvořený specializovaný učební plán, který pokrývá jak firmwarovou tak softwarovou část. Dále jsou zde ukázány ukázkové lekce, které se zaměřují na ovládání BLDC motoru, SLAM algoritmy další.

Závěrem je práce shrnuta a je poukázáno, že vytvořená robotická platforma, plně splňující své zadání, je skvělá mobilní platforma určená přímo k nasazení ve výuce. Její modulární design, hardwarové schopnosti, sofistikovaný firmware a flexibilní software ji činí cenným výukovým nástrojem, vhodným pro výuku základních konceptů robotiky na univerzitách.

Klíčová slova: SRobot, Mobilní robotická platforma, Všesměrový robot, Modulární design, Kapacitní dotykový nárazník, BLDC motory, STM32, FOC, ISR, TSC, LiDAR, ROS balíčky, Gazebo simulace, SLAM algoritmy, Vzdělávací robotická platforma, Kinematika a dynamika všesměrových robotů

Překlad názvu: Návrh a realizace všesměrového podvozku pro výukovou robotickou platformu — Vzdělávací robotická platforma speciálně navržená pro studenty vysokých škol

Contents

1 Introduction	1	8.3 PID controllers	55
2 Overview of the existing indoor educational robotic platforms	3	8.4 Interrupt service routine	55
Part I		8.5 Driver board faults	56
Construction		8.6 Communication	57
3 Robot chassis	11	9 Main board	59
3.1 Body	12	9.1 Power management	59
3.1.1 Lower part	12	9.1.1 Momentary switch	59
3.1.2 Upper part	12	9.1.2 Front panel connector	61
3.2 Motors and wheels	13	9.2 Analog data processing	61
3.2.1 Motor	14	9.3 Communication	62
3.2.2 Wheel	16	9.3.1 Driver boards	62
3.3 Capacitive bumper	17	9.3.2 Intel NUC	63
4 Batteries	21	9.4 Robot protection	64
4.1 Battery	21	9.5 Extensions	65
4.1.1 Charger	22	Part III	
4.1.2 Balancer	24	Software	
4.2 Battery connectors	25	10 Robot Operating System	69
5 Printed Circuit Boards	27	10.1 ROS packages	69
5.1 Main board PCB	28	10.1.1 Robot packages	69
5.1.1 Analog measurements	28	10.1.2 Other used packages	70
5.1.2 Power management	29	10.2 Start up sequence	71
5.1.3 Digital control	30	10.2.1 Demonstration	71
5.1.4 PCB schematic	31	10.3 Robot model	73
5.1.5 PCB design	31	10.4 Robot messages	74
5.2 Driver board PCB	36	10.5 Odometry	74
5.2.1 PCB schematic	37	10.6 RViz	76
5.2.2 PCB design	37	10.7 Gazebo simulator	77
5.3 Battery board PCB	41	Part IV	
5.3.1 PCB design	42	Possible usage of the educational robotics platform	
6 Electronics and control systems	43	11 Robotics curriculum	81
6.1 Onboard computer	43	12 Example lessons	85
6.2 LiDAR sensor	44	12.1 Introduction to BLDC Motor Control Using the Robotic Platform and STM32Cube IDE	85
6.3 Display	45	12.2 Introduction to Odometry and Robot Motion Control using a Mobile Robot Platform	88
Part II		12.3 Exploring SLAM Algorithms and Frontiers Detection using the Robotic Platform	91
Firmware		13 Conclusions	95
7 Programming instructions	49	13.1 Future improvements	96
7.1 Robot diagram	52		
8 Driver board	53		
8.1 Motor control	53		
8.2 Encoder	54		

13.1.1 Hardware	96
13.1.2 Software	97
13.2 Resources	98

Appendices

A Acronyms	101
B Bibliography	103
C Project Specification	105

Figures

2.1 The robot comparison	7
3.1 The fully assembled educational robotic platform	11
3.2 Lower part of the robotic chassis	13
3.3 Upper part of the robotic chassis	14
3.4 Upper part of the robotic chassis with all electronics	15
3.5 Module of BLDC motor with omni wheel	15
3.6 Lower part of the robotic chassis with mounted BLDC motor modules	16
3.7 Motor parameters [11]	16
3.8 Omni directional wheel used in the robotic platform [12]	17
3.9 Capacitive bumper	18
4.1 Disassembled battery module	23
4.2 Battery module	24
4.3 XL4015 5A CC/CV battery charger	25
4.4 4S 30A battery protection and balancing module	26
4.5 Connectors for connecting the battery to the Main board [14]	26
5.1 Lower part of the robotic chassis with mounted BLDC motor modules, capacitive tactile bumper and created PCB boards	28
5.2 The Main board	29
5.3 A schematic of the Main board	33
5.4 Different layers of the Main board PCB	35
5.5 The Driver board	36
5.6 A schematic of the Driver board	39
5.7 Different layers of the Driver board PCB	40
5.8 The Battery board	41
5.9 Different layers of the Battery board PCB	42
6.1 Intel NUC with LED diffusor and quickly detachable spring connector	44
6.2 RPlidar A3 mounted on the NUC	45
6.3 The Waveshare 4-inch IPS LCD display mounted on the robot	45
7.1 STM32CubeIDE	50
7.2 Programming connectors of all the microcontrollers located in the robot	51
7.3 Schematic of the programming connector	51
7.4 Robot connection diagram	52
8.1 Driver board pinout	54
9.1 Main board pinout	60
10.1 The robot model displayed in the ROS visualization (RViz) application	74
10.2 Coordinate system of the robot with the respective position of the wheels	76
10.3 The robot visualization in the RViz application	77
10.4 The created Gazebo world for simulation purposes	78
10.5 Close-up view of the robot within the Gazebo world	78

Tables

2.1 A comparison of the existing indoor educational robotic platforms	5
--	---



Chapter 1

Introduction

The SRobot, a versatile indoor mobile robotic platform meticulously designed to serve as an educational tool for university students, forms the core focus of this master's thesis. This platform aims to impart fundamental robotics concepts spanning both hardware and software domains, offering a comprehensive understanding of robotics principles.

The overview of the SRobot underlines its unique advantages over comparable robots. Key features include its modular design, extended operational lifespan, and innovative capacitive tactile bumper, endowing the platform with true omnidirectional capabilities. The integration of Brushless Direct Current (BLDC) motors not only ensures enhanced precision and smoother movement but also contributes to energy efficiency. The cost-effectiveness of the SRobot, with a significant portion of the budget allocated to the Intel NUC and RPLidar A3 components, is an additional highlight.

The construction process provides a step-by-step guide for replicating the educational platform, elucidating the selection and functionality of each component. The purpose and structure of the capacitive bumper, along with battery selection, balancing, and charging processes, are meticulously explained. The detailed Printed Circuit Board (PCB) design with all integrated components is visually presented, offering a clear roadmap for assembly.

The firmware section dives into the roles of STM32 microcontrollers, providing clear instructions for code upload to each board. The robot connection diagram enhances the understanding of the communication infrastructure. BLDC motor control, including the use of built-in encoders, is thoroughly explained, along with the Proportional–Integral–Derivative (PID) controller's role in ensuring smooth motor control. The firmware discussion delves into various interrupts managed by the ISR, communication protocols between each board, and the value of multiple analog measurements. The purpose of the momentary switch and the front panel connector is clarified, alongside the communication between the robot and the onboard computer. The protective

features offered by the board are highlighted, as well as the high flexibility and wide array of connectors available to accommodate diverse modules and sensors.

The software segment provides insight into the Robot Operating System (ROS) packages employed by the robot, explaining topics and messages within each package. The startup sequence and launched terminals offer a comprehensive perspective on the robot's functioning. The creation and utilization of the robot model for simulation and visualization purposes are explained, along with the processing and publishing of received messages from the robot. The implementation of odometry is thoroughly explained, complete with the relevant equations and the established robot coordinate system. The ROS visualization (RViz) application, running alongside the demonstration script, is showcased, and the importance of Gazebo simulation, depicted through the simulation world, is emphasized.

The final part of the master's thesis emphasizes the suitability of the robot as an educational platform. It showcases a range of educational use cases and topics that could be taught using the platform, offering a comprehensive robotics curriculum focused on the robot as an educational tool. Additionally, the provided example lessons, covering topics from firmware implementation for each board to the utilization of SLAM algorithms for exploration purposes, solidify the value of the robotic platform as a versatile and effective educational resource.



Chapter 2

Overview of the existing indoor educational robotic platforms

There is a diverse range of educational robotic platforms available worldwide, catering to various age groups and educational levels. This thesis primarily concentrates on the development of an omnidirectional indoor mobile robot platform. As such, the review will compare robotic platforms that share similar construction, purpose, and affordability. The comparison will not include building kit robots, humanoid robots, unmanned aerial vehicles, robotic manipulators, or outdoor-oriented platforms. Instead, the focus will be on robots that align closely with the design principles and goals of the robotic platform.

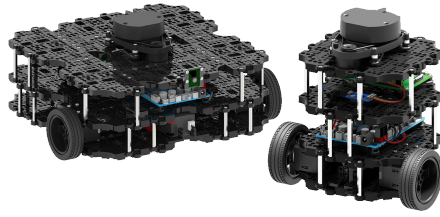
The following table 2.1 provides an overview of key information about the selected educational robotic platforms. These platforms are compared across various categories to gain a better understanding of their capabilities and features. Additionally, a visual representation 2.1 offers an enlarged view of these robots, providing a quick glimpse of their physical attributes.

2. Overview of the existing indoor educational robotic platforms

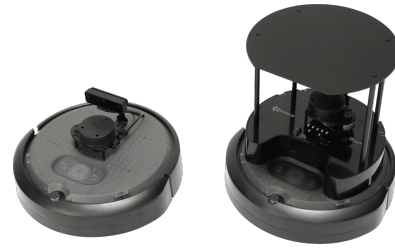
					
Clearpath Robotics	KUKA	Botsync	Misty Robotics	Segway Robotics	Open-source project
DINGO	YouBot	VOLTA	Misti II	RMP Lite 220	SRobot
686 mm x 517 mm	580 mm x 376 mm	520 mm x 365 mm	254 mm x 203 mm	730 mm x 499 mm	306 mm in diameter
13 kg	20 kg	Not mentioned	2.7 kg	33 kg	5.45 kg
20 kg	20 kg	30 kg	Not mentioned	50 kg	30 kg
Differential and Omnidirectional	Omnidirectional	Differential	Differential	Differential	Omnidirectional
2 hours	1.5 hours	2 hours	2 hours	Up to 10 hours	4.5 - 9 hours
410 Wh	120 Wh	Not mentioned	77 Wh	1152 Wh	222 Wh
Moderate	Poor	Poor	Extensive	Poor	Extensive - all information provided
Almost no support	Almost no support	Almost no support	Active support - strong community	Almost no support	No support
Relatively easy	Requires special knowledge	Medium	Easy	Easy	Medium
Highly customizable	Moderate	Moderate	Moderate	Highly customizable	Highly customizable
Mini-ITX Computer i3	Mini-ITX Computer Intel Atom	Odriod XU4	Qualcomm Snapdragon 410 and 820 mobile processor	No computer on-board	Intel NUC 12 Pro i5
IMU	IMU, Robotic manipulator	RPLidar A2, IMU, Astra Orbec Pro depth camera	4K Camera, Depth sensor, capacitive sensors, microphones, obstacle avoidance sensors, speakers, LCD screen	IMU	RPLidar A3, IMU, obstacle avoidance sensors, LCD display
Great for large industrial applications	Great for research and development	Excellent - well-suited for educational purposes	Excellent - well-suited for educational purposes	Great robotic mobile platform that can be built upon	Excellent - well-suited for educational purposes
\$ 23 111	\$ 30 861	Not mentioned	\$ 4 295	\$ 2 999	\$ 2999

Table 2.1: A comparison of the existing indoor educational robotic platforms

2. Overview of the existing indoor educational robotic platforms



(a) : TurtleBot 3 - Waffle Pi and Burger version [1]



(b) : TurtleBot 4 - Lite and Standard version [2]



(c) : Husarion - ROSbot 2 PRO [3]



(d) : Husarion - ROSbot XL [4]



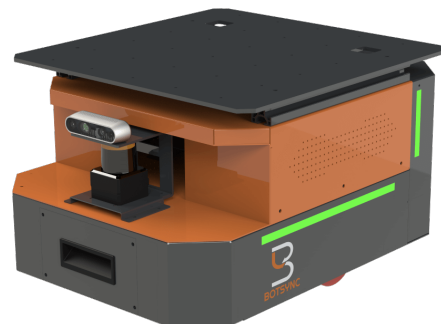
(e) : Robotnik Summit XL [5]



(f) : ClearPath - DINGO [6]



(g) : YouBot KUKA [7]



(h) : VOLTA [8]



(i) : Misti II [9]



(j) : RMP Lite 220 [10]



(k) : SRobot - top view



(l) : SRobot - side view

Figure 2.1: The robot comparison

In terms of educational suitability, all of the mentioned robots can be used for educational purposes, but some are purposefully designed to excel in that role. Particularly, the Turtlebots by Open Robotics and the ROSbots by Husarion stand out as highly suitable platforms for teaching SLAM algorithms and similar methodologies, making them ideal for indoor mobile robotics education. On the other hand, the Misti II robot from Misti Robotics is specifically tailored for teaching image processing algorithms.

In comparison to these selected platforms, the SRobot offers several distinctive advantages. Firstly, it boasts an extended operational lifetime, providing longevity for educational use. Additionally, it provides a modular design, comprehensive hardware documentation, and superior computational power. The SRobot is equipped with a diverse range of sensors, including the innovative capacitive tactile bumper, which grants the platform true omnidirectional capabilities. Furthermore, the cost of the SRobot is notably lower compared to other options, with a significant portion of the price allocated to the Intel NUC and RPLidar A3.

Furthermore, it's noteworthy that only three of the compared robots utilize BLDC motors. The Summit-XL from Robotnik, the YouBot from KUKA, and the SRobot are the platforms featuring these motors, setting the SRobot apart as a distinct and unique robotic platform in this regard.



Part I

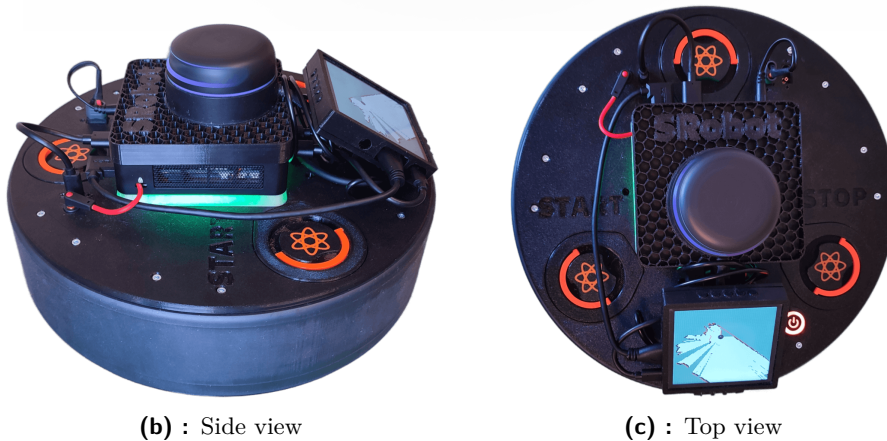
Construction

Chapter 3

Robot chassis



(a) : Model of the educational robotic platform



(b) : Side view

(c) : Top view

Figure 3.1: The fully assembled educational robotic platform

The robot chassis is made only from 3D-printed parts. The primary emphasis during the design phase was placed on the platform's robustness, user-friendly attributes, ease of maintenance, and swift reparability. The

body of the robot has a circular shape. This shape was selected mainly due to the fact, that the robot should serve as an educational platform for students. This design enables effortless orientation and smooth navigation in complex environments, substantially facilitating path-planning tasks.

Due to the circular shape and the requirement for omnidirectional capabilities, the robot uses three omni wheels, directly linked to three BLDC motors. Further details regarding this configuration can be found in section 3.2.

To enable the robot's operation without relying on additional sensors, the robot uses a capacitive bumper to detect any obstacles, more about that in section 3.3.

■ 3.1 Body

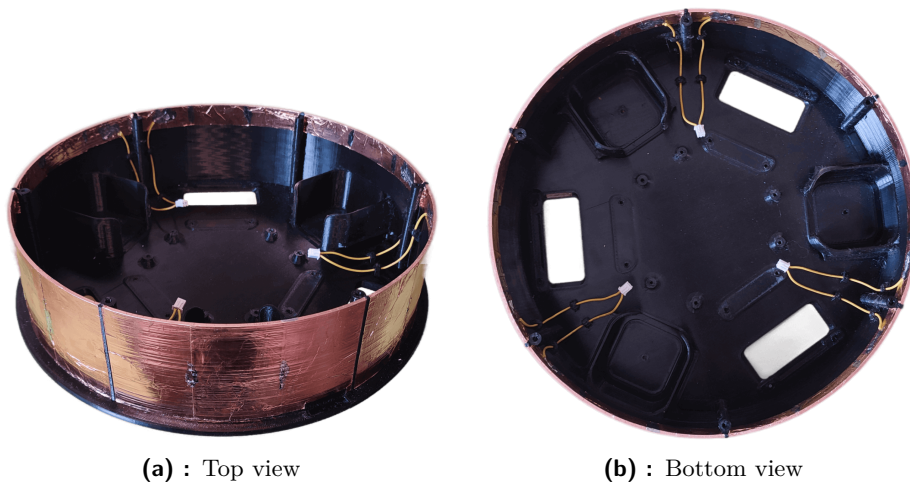
The robot body is divided into two sections. The lower part houses all essential components, including electronics, motors, wheels, and capacitive sensors. The upper part serves as an intermediary linking the robot to a single-board computer for high-level control, particularly connecting it to the Intel NUC in this instance. Additionally, the upper section serves as an interface between the user and the robot, providing real-time status updates and enabling seamless user control over its operations.

■ 3.1.1 Lower part

Figure 3.2 depicts the lower part of the robot, consisting of a single 3D-printed component. This part is equipped with capacitive electrodes affixed onto its surface, as elaborated in section 3.3. Subsequent figures will showcase the integration of additional components such as electronics, motors, and wheels onto this lower part, demonstrating its gradual assembly.

■ 3.1.2 Upper part

Figure 3.3 captures the upper part of the robot, showcasing its various connectors. The XT60 connectors serve a dual purpose. One of the XT60 connectors serves as the power supply for the Intel NUC, while the other functions as the charging port for the robot. The USB-C connector is designated for seamless communication with the robotic platform and can even power up the robot when sufficient power is available. The 3.5 mm jack connector enables the NUC to connect with its front panel connector. A momentary switch is also present, responsible for toggling the robot's power, as elaborated in section 9.1.1. Furthermore, the yellow cables are linked to



(a) : Top view

(b) : Bottom view

Figure 3.2: Lower part of the robotic chassis

the START and STOP capacitive buttons. The upper part is equipped with a WS2812B LED ring that signals the robot's state through various visual signals.

Figure 3.4 shows the side and top view of the upper part, fully equipped with all electronics. An Intel NUC is securely connected to the upper part through a user-friendly, easily detachable spring connector. The LED ring's light signalization is effectively diffused through a diffuser, situated between the upper part and the NUC. Additionally, positioned above the NUC is the RPLidar A3, a crucial component utilized for localization purposes. Furthermore, there is a 4-inch display mounted on the upper part. The display is featuring a capacitive touchscreen and is providing an interactive user interface for controlling and monitoring the robot's state.

■ 3.2 Motors and wheels

Figure 3.5 represents the BLDC motor module with the omnidirectional wheel. This module can be seamlessly installed between the lower and upper parts of the robot using 3 mm screws. In the event of any issues with the module, it can be conveniently detached from the robot and promptly replaced. The robot houses a total of three of these modules. Each module possesses three connectors, with one designated for motor connection to the robot, another for linking the AS5048A magnetic encoder to the robot, and the last one for an NTC thermistor that measures the motor's temperature.

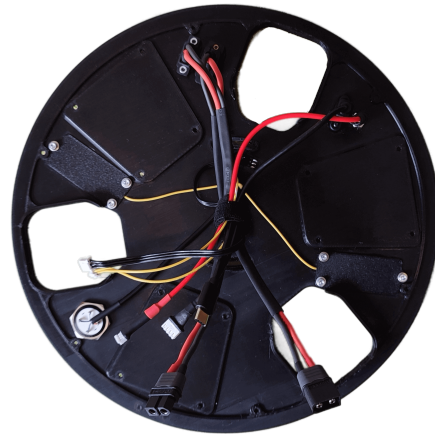
Figure 3.6 showcases the BLDC motor module with the omnidirectional wheels directly mounted onto the lower part of the robotic chassis.



(a) : Side view



(b) : Top view



(c) : Bottom view

Figure 3.3: Upper part of the robotic chassis

3.2.1 Motor

Figure 3.7 shows the parameters and description of the BLDC motor used in this educational robotic platform. This motor is commonly utilized as a gimbal motor, thus featuring a low motor velocity constant ($K_V = 55$). With a lower K_V constant the motor is slower, but usually stronger. Due to its inherent characteristics, this motor does not necessitate a gearbox, allowing for direct mounting to the wheel. Furthermore, the motor is equipped with an integrated AS5048A 14-bit magnetic rotary encoder, which enables precise control of the motor position and velocity.

The choice of the BLDC motor was primarily driven by its effectiveness, particularly when controlled using Field-Oriented Control (FOC). This control technique allows the motor to deliver maximum strength with optimal efficiency, making it especially advantageous for battery-driven systems. Thanks

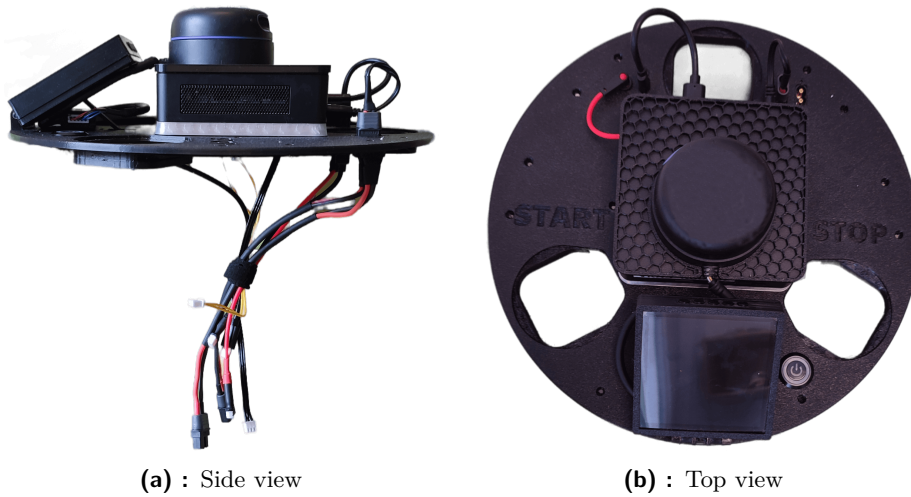


Figure 3.4: Upper part of the robotic chassis with all electronics

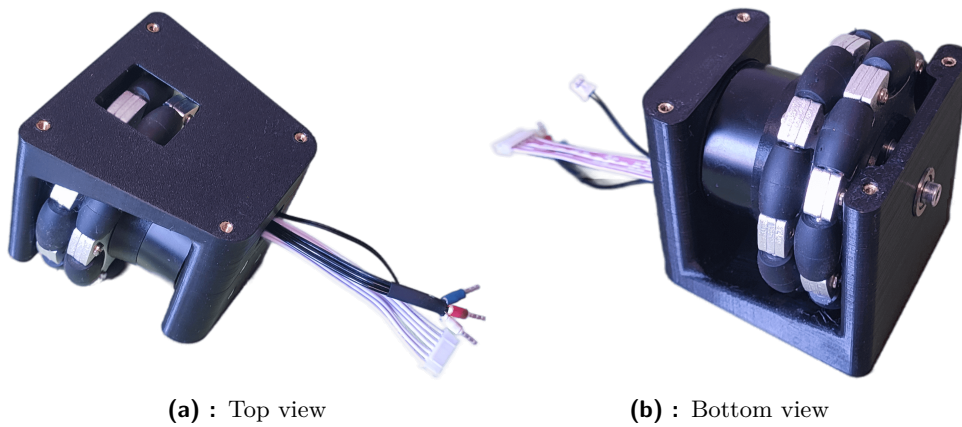


Figure 3.5: Module of BLDC motor with omni wheel

to this type of control, the robot's operational time can be significantly extended, ensuring longer-lasting and more efficient performance.

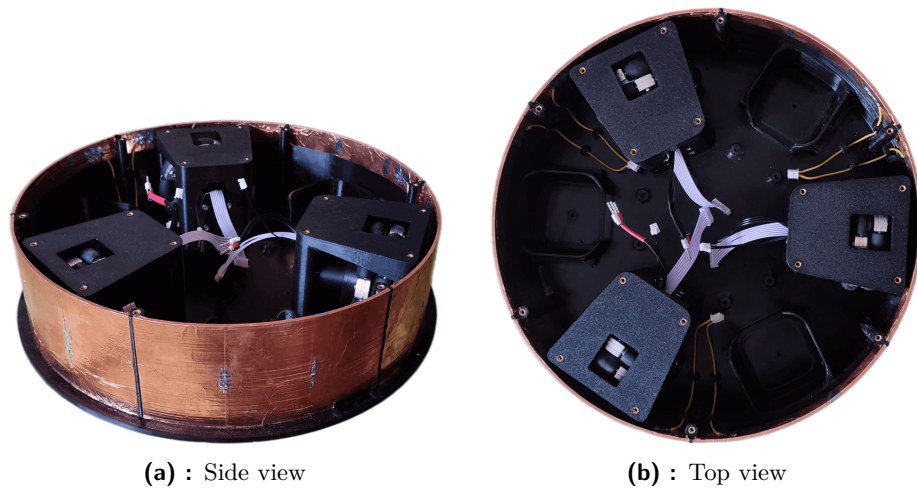


Figure 3.6: Lower part of the robotic chassis with mounted BLDC motor modules

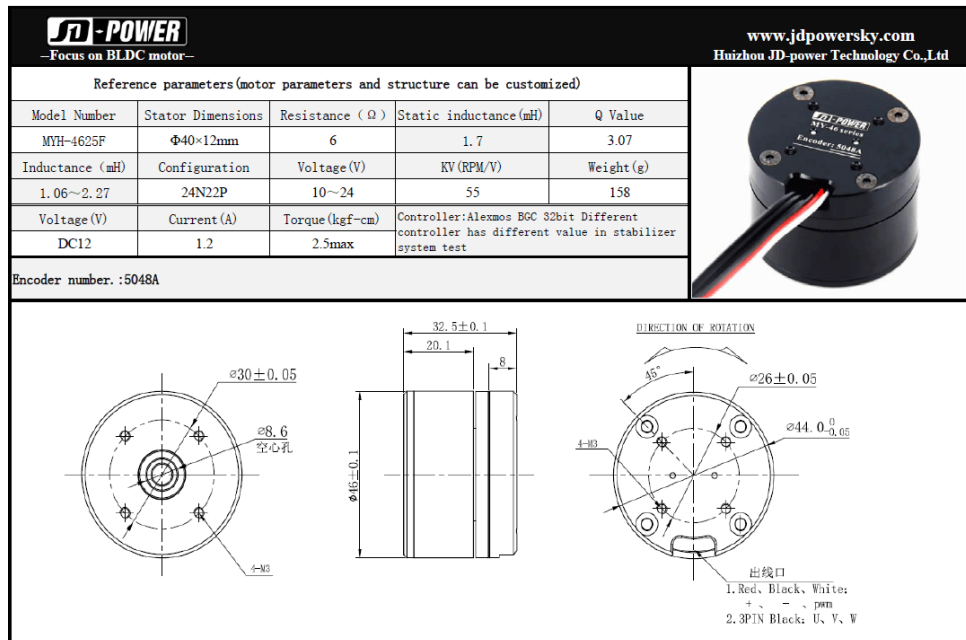


Figure 3.7: Motor parameters [11]

3.2.2 Wheel

The omnidirectional robotic platform can be equipped with either Mecanum wheels or Omni wheels. Robots equipped with Mecanum wheels typically adopt a rectangular shape as they require a minimum of four wheels. Conversely, robots utilizing Omni wheels can function with a minimum of three wheels, which often leads to a circular configuration. The circular shape holds distinct advantages for students, providing seamless orientation and smooth navigation in complex environments, significantly simplifying path-planning

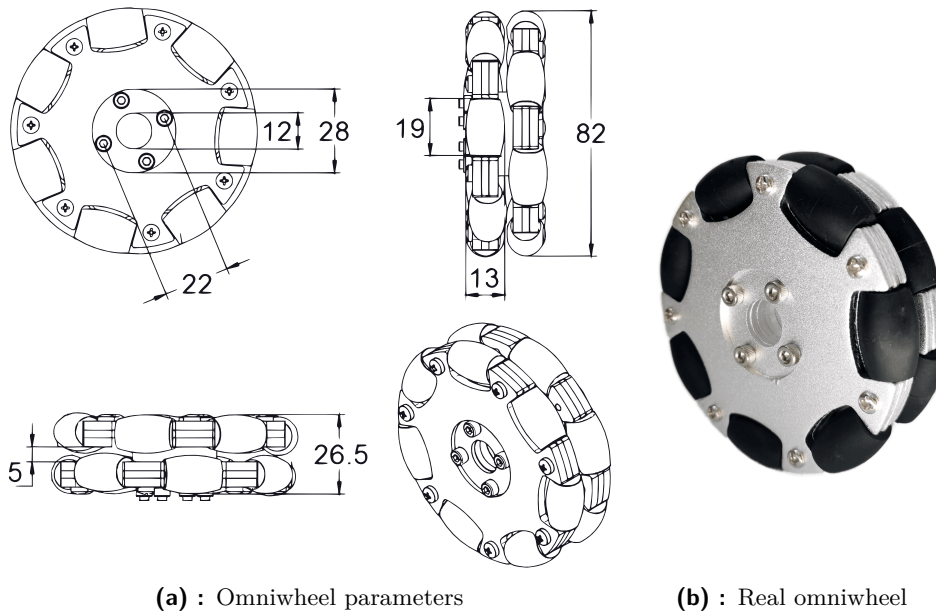


Figure 3.8: Omni directional wheel used in the robotic platform [12]

tasks. An additional benefit of the circular design is that it necessitates fewer motors, reducing the robot's current consumption, a critical aspect for battery-driven systems. Moreover, with fewer motors, the robot's internal space is larger, and the robot is lighter and cheaper. However, it is important to acknowledge certain drawbacks associated with this design. The robot's stability may be compromised due to only three contact points with the ground, requiring careful consideration of its center of gravity, symmetry, and balance. Furthermore, controlling a three-wheel-based robot proves more challenging compared to four-wheel-based counterparts, and the calculation of robot odometry becomes significantly more complex. Despite these challenges, the circular-shaped Omni-wheeled robot offers several compelling advantages for educational platforms, making it a favorable choice for this specific application.

The Omni wheel used in this design is displayed in figure 3.8. It is constructed from a combination of aluminum alloy and rubber materials. The wheel has a diameter of 82 millimeters and is designed to withstand a load weight of up to 15 kilograms.

3.3 Capacitive bumper

The robot employs tactile sensors based on capacitance change to detect obstacles in its surrounding environment. Each tactile sensor consists of two conductive plates separated by an elastomeric separator, ensuring compliance. When pressure is exerted on the plates, there is a change in capacitance,

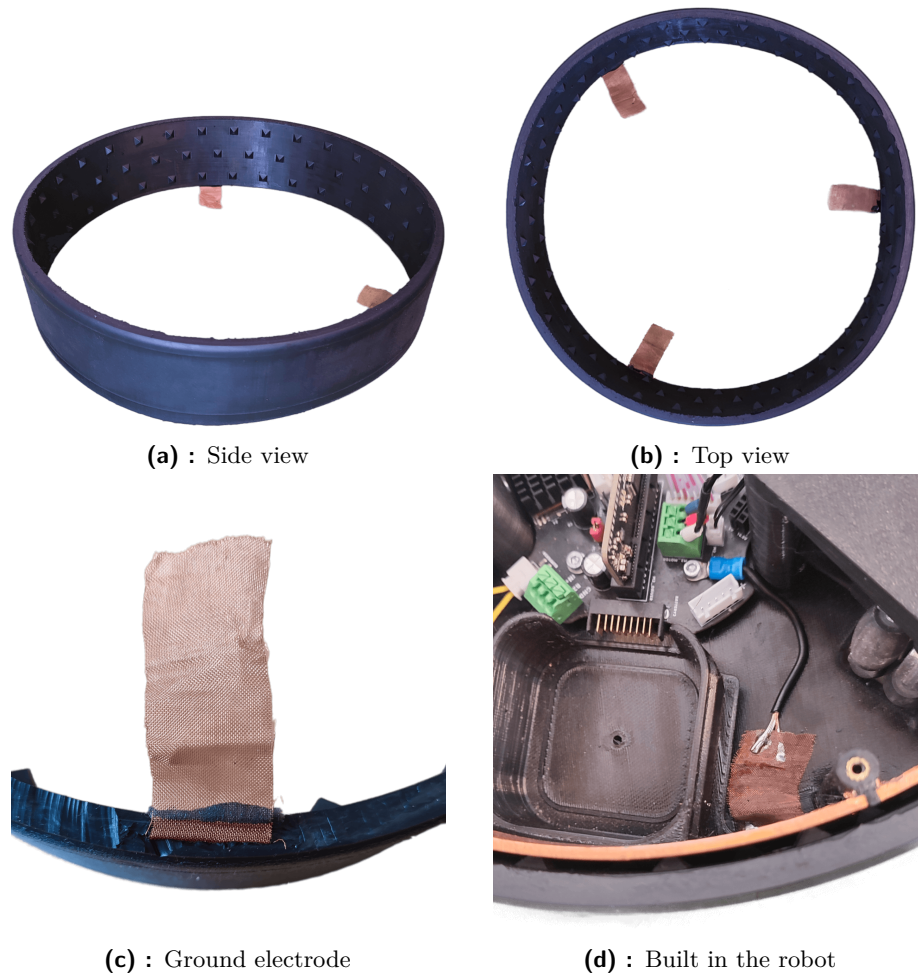


Figure 3.9: Capacitive bumper

which is proportional to the force applied. The robot is equipped with six tactile sensors in total, enabling it to perceive both the presence of obstacles and the direction of impact.

Figure 3.2 displays all six capacitive directional electrodes, directly connected to the robot. The outer electrode is embedded within the silicone cover, as shown in figure 3.9, and it is connected to the ground of the robot.

The robot measures the capacitance between these electrodes using its built-in Touch Sensing Controller (TSC) that utilizes a charge transfer acquisition technique [13]. The connection of the outer electrode to the ground ensures that the capacitances of the directional electrodes remain unaffected when the robot is near people or other grounded objects. As a result, any change in capacitance is solely dependent on the pressure exerted on the capacitive bumper. This allows the robot to detect and respond to its surroundings, enhancing its ability to navigate safely in various environments.

The capacitive bumper is made from silicone using two mold forms. Between the molds is inserted copper fabric serving as the ground electrode. The bumper's inner wall features small pentahedrons, which serve both as springs and spacers. This component is then directly mounted onto the robot's body, providing an effective and resilient tactile sensing system.

Chapter 4

Batteries

As a consequence of opting for the three-wheel platform as discussed in subsection 3.2.2, the robot's stability becomes a critical consideration, necessitating careful attention to its center of gravity, symmetry, and balance. Given this concern, the battery must be positioned either at the robot's center or evenly distributed across its structure. However, due to the center space being reserved for the Intel NUC and Light Detection and Ranging (LiDAR) sensor, as mentioned in chapter 6, the battery had to be distributed to ensure optimal weight distribution and maintain the robot's stability.

4.1 Battery

The robot's larger internal free space allowed for the batteries to be placed inside the robot. The primary considerations when selecting the batteries were easy maintenance, user-friendliness, robot safety, and overall lifetime. For meeting these requirements, the lithium-ion battery was chosen, particularly the 26650 battery, as depicted in figure 4.1a. These batteries were preferred over the 18650 and 21700 options due to their extended lifetime and higher capacity, all while maintaining the same output current capabilities.

Specifically, this battery offers a nominal capacity of 5000mAh, is capable of enduring over 1000 charging cycles, and has a rated current of 30 amperes, making it a well-suited choice for the robot's power requirements.

The selection of a 4-cell battery was made to ensure the minimal operational voltage for the Intel NUC while adhering to the recommended maximum voltage for the motor's drivers. This choice precisely meets the minimum 12V requirement for powering the Intel NUC, while the maximal battery voltage of 16.8V remains comfortably below the recommended 20V for the motor driver, ensuring safe and efficient operation. With this setup, the battery achieves an energy capacity of 74 watt-hours. To further enhance the robot's

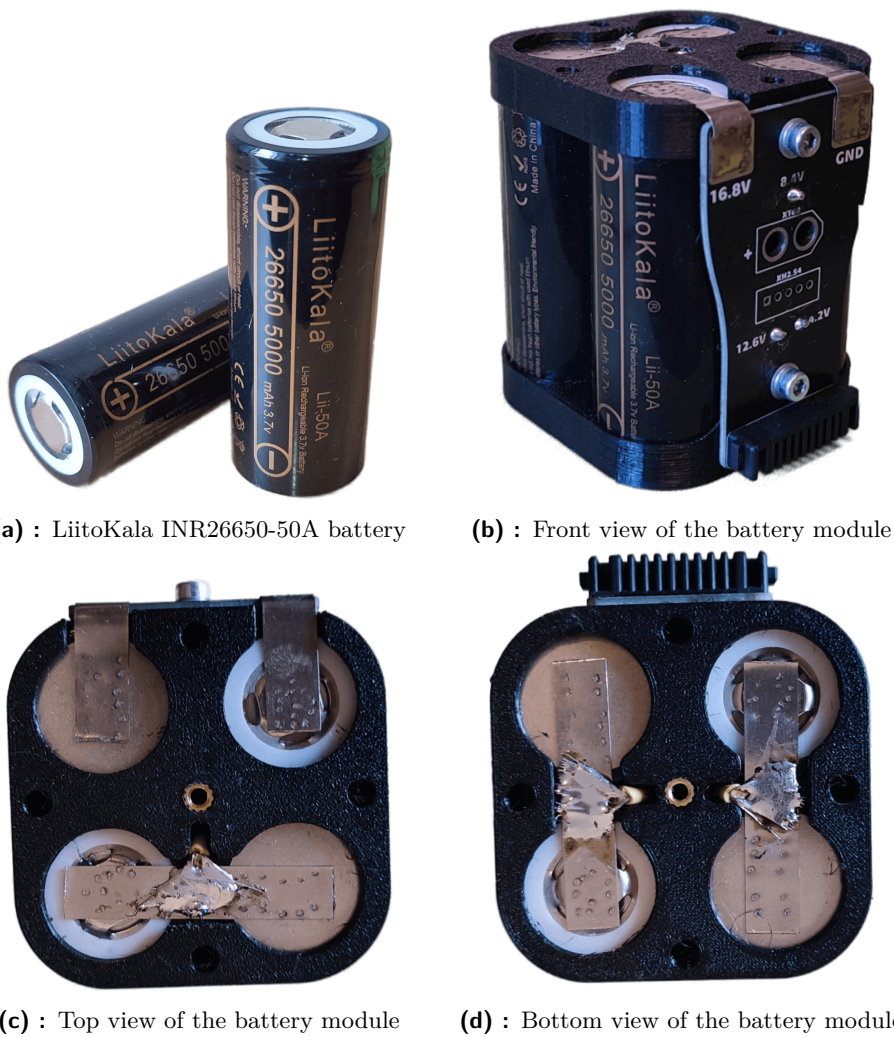


Figure 4.1: Disassembled battery module

module can be readily replaced with a new one, preserving the robot's operability. Conversely, with an onboard solution, a malfunctioning charger could render the entire robot unusable.

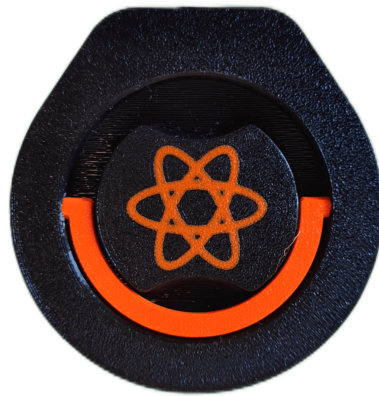
However, using external charger modules led to an unexpected drawback. When directly connected to the battery, the charger draws some energy from it, causing battery discharge over time. This problem was solved by adding a Schottky diode between the charger and the battery. Another issue that was encountered involved ground connections. Although the charger appeared to share the input and output grounds when connected to the power supply, the grounds disconnected. It was determined that the charger required separate ground connections for the input voltage and the output voltage to work properly.

While those issues will be addressed in future versions, subsection 4.1.2



(a) : Battery module without heat shrink sleeve

(b) : Front view



(c) : Top view



(d) : Bottom view

Figure 4.2: Battery module

provides details on how to manage this concern in this current configuration, ensuring the safe and optimal use of the robot and its batteries.

4.1.2 Balancer

This robot was initially designed to feature an active cell balancing circuit, which effectively balances the cells by directly transferring energy from one battery to another using various capacitors. However, due to the problematic issue with the battery charger mentioned in subsection 4.1.1, the active cell balancing circuit had to be replaced with a battery protection and balancing module depicted in figure 4.4.

In the new setup, the charger output ground is directly connected to

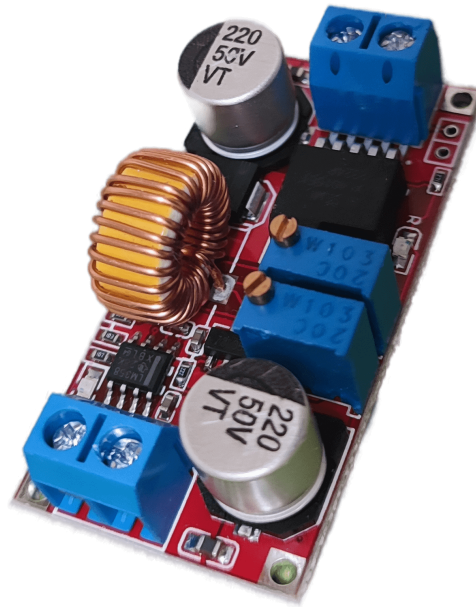


Figure 4.3: XL4015 5A CC/CV battery charger

the P- pad, which is shown in the figure of the battery protection module. This connection creates a virtual ground for the charger and also allows the protection circuit to disconnect the charger from the battery once it's no longer charging. The input ground of the charger is connected directly to the input power supply.

Moreover, the balancing part of the module only operates during charging and not continuously like the active cell balancing circuit would. While the replacement module serves its purpose by ensuring battery protection and balancing during charging, there are certain drawbacks compared to the original design.

■ 4.2 Battery connectors

For the connection between the battery and the main board, the robot uses the connector shown in figure 4.5. Each pin of this connector provides a current rating of 7 amperes. To efficiently power the robot, the battery employs three pins for ground connections, three pins for supplying power to the robot, and the remaining four pins for balancing the battery cells. Based on this configuration, it is crucial to ensure that the average current drawn from any single battery does not exceed 21 amperes, with the possibility of occasional peaks reaching a maximum of 30 amperes. By adhering to these current limits, the robot can operate safely and effectively, preventing any potential overload situations that may compromise its performance or damage the components.

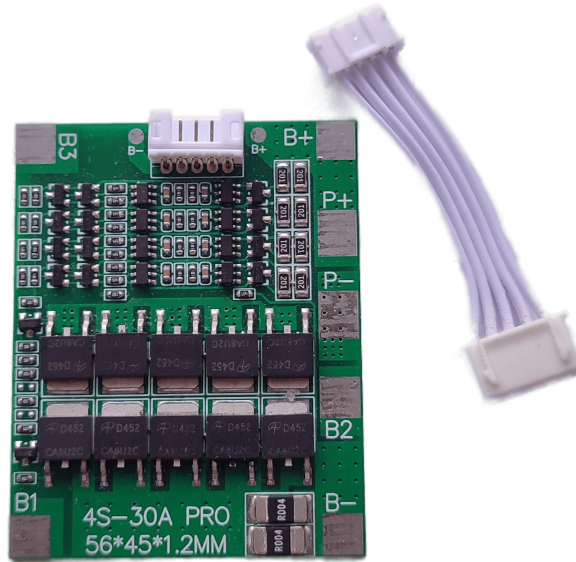
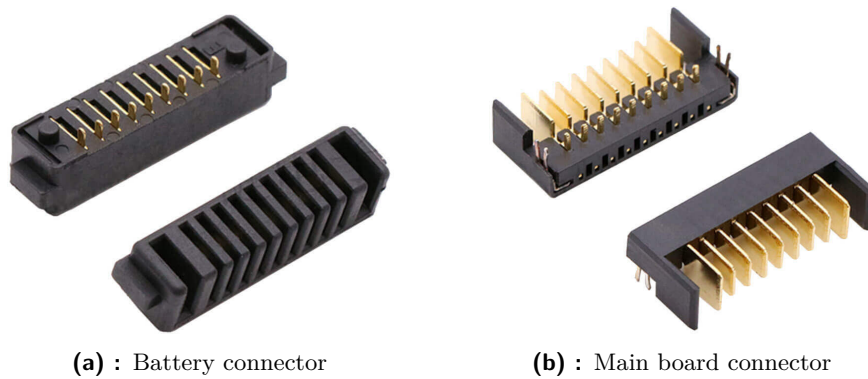


Figure 4.4: 4S 30A battery protection and balancing module



(a) : Battery connector

(b) : Main board connector

Figure 4.5: Connectors for connecting the battery to the Main board [14]

Chapter 5

Printed Circuit Boards

The PCBs were designed using Altium Designer. The primary objective was to create a unified solution on a single board, eliminating the need for multiple modules connected by cables. This design approach resulted in a more robust, stable, and higher-frequency electronics system, while also simplifying maintenance procedures. By minimizing the number of cables, the robot's potential failure points were significantly reduced, enhancing overall reliability.

There are four custom-made PCBs in total, but only three will be highlighted here. The central and most crucial PCB is called the Main board. It contains all the essential electronics for the robot's operation and is responsible for the communication between the NUC and the motor drivers. The Driver board is the second PCB, with three of them directly inserted into the Main board. The Driver boards are responsible for the precise control of the BLDC motors. The third board is the Battery board, which acts as the connection interface between the batteries and the robot. The last PCB, not mentioned in this section, is available in the git repository of this master's thesis. Its purpose is to facilitate testing of the Driver board with motors outside of the robot, providing a valuable tool for development and evaluation.

Since the robot is intended as an educational platform, special attention was given to facilitating easier debugging, signal processing, and signal visualization. Most of the pins on all the boards are accessible as outputs, making it simpler to visualize signals during debugging and analysis. Additionally, the PCBs offer convenient access to all of their peripherals, allowing for the insertion of a wide range of sensors to enhance the robot's capabilities.

Figure 5.1 showcases the created PCB boards mounted onto the lower part of the robotic chassis.

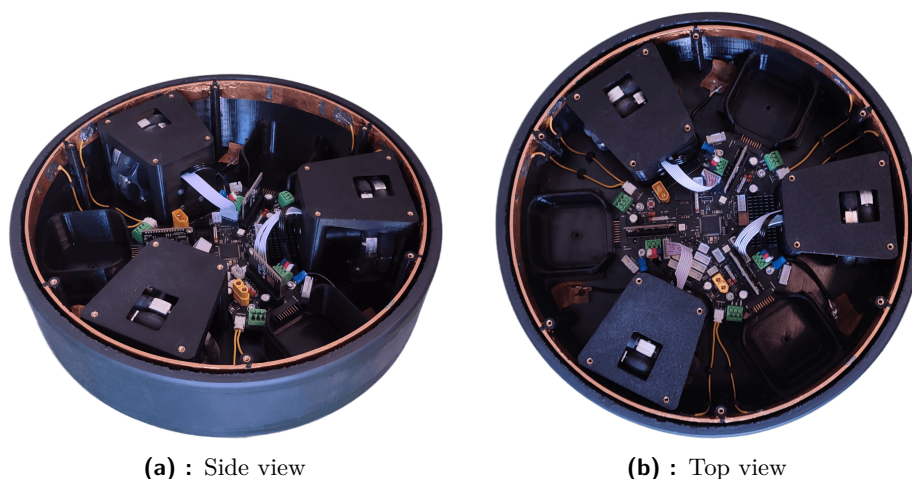


Figure 5.1: Lower part of the robotic chassis with mounted BLDC motor modules, capacitive tactile bumper and created PCB boards

5.1 Main board PCB

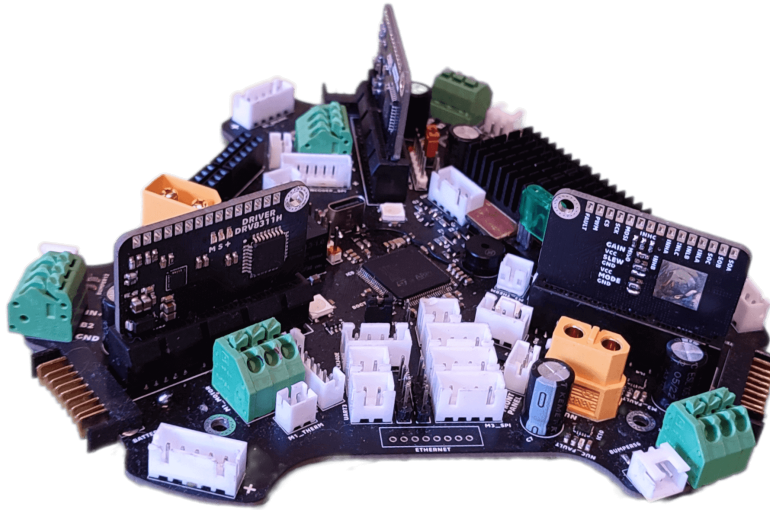
5.1.1 Analog measurements

The Main board performs multiple analog measurements to monitor crucial parameters of the robot. One of the crucial measurements includes monitoring the robot's voltage and battery voltages. These values are measured using a voltage divider, protected with a Zener diode.

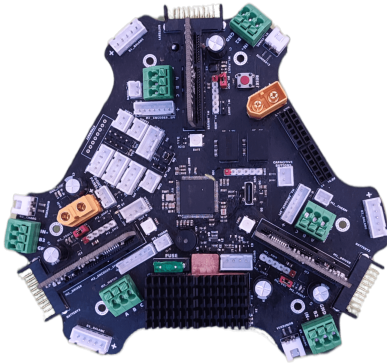
Additionally, the Main board is responsible for measuring seven current values. This includes current flowing from and to the batteries, current flowing to the Driver boards and motors, and the overall robot's current consumption. The overall consumption is measured with the help of the Protected MOSFET (PROFET™), which is outputting current proportional to the actual consumption. The remaining currents are measured using Hall-Effect Linear Current Sensor ACS711, capable of measuring currents in the range of ± 15.5 amperes.

Temperature monitoring is another essential function of the Main board. It measures the temperatures of the motors, Driver boards, and the Main board itself. The temperatures of the motors are measured using NTC thermistors, while the temperature of the Main board is determined via a built-in thermometer inside the STM32 microcontroller. As for the Driver board temperatures, they are measured similarly but transmitted to the Main board using the Serial Peripheral Interface (SPI) protocol.

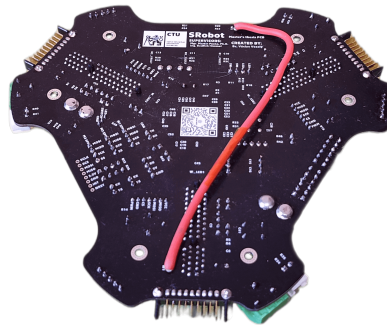
Furthermore, the Main board handles the measurement of eight capacitive tactile sensors. Six of these sensors are used for the capacitive bumper, allow-



(a) : Side view



(b) : Top view



(c) : Bottom view

Figure 5.2: The Main board

ing the robot to detect obstacles and respond to its surroundings effectively as discussed in section 3.3. The remaining two sensors are utilized for the START and STOP buttons as mentioned in subsection 3.1.2.

■ 5.1.2 Power management

As mentioned in the section 4.1, the batteries are connected using Schottky diodes. Each Schottky diode can handle a current of up to 30 amperes. However, due to the significant heat generated at such high currents, the Schottky diodes require heatsinks for effective heat dissipation. To address this issue, the diodes are strategically placed close to each other and connected with a large common heatsink. This arrangement ensures efficient heat dissipation and prevents overheating. One route from the batteries could not be routed on the board and is instead connected with a cable, as shown in figure 5.2c.

The Driver boards are connected to the Main board through Peripheral Component Interconnect (PCI) Express connectors. Each Driver board communicates with the Main board via its own SPI. Additionally, the boards share a common clock source in the form of a precise 8 MHz crystal located on the Main board. The Main board provides this clock source to the Driver boards with its clock output capabilities [15].

For signaling purposes, the Main board features a passive electromagnetic buzzer with a sound pressure level of 85 decibels, allowing the robot to audibly communicate its states. Furthermore, The Main board is also equipped with three WS2812B LEDs, strategically placed next to the Driver boards. This provides a visual indication of the states of each board. The Main board also includes a connector for adding more LEDs to the robot, as mentioned via the LED color ring in subsection 3.1.2.

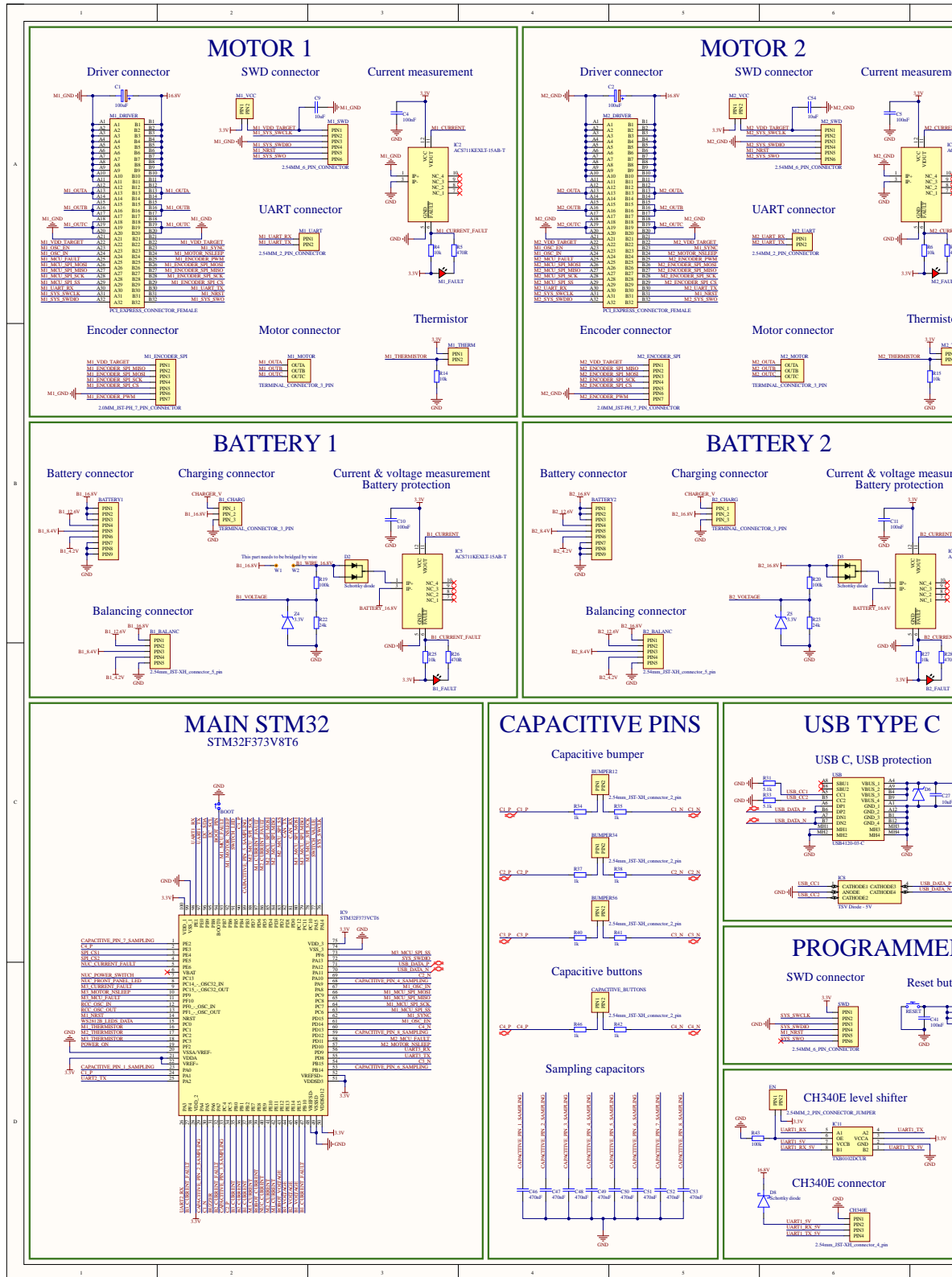
In addition to the above features, the Main board offers multiple connectors for various peripherals, including three unique Universal Asynchronous Receiver-Transmitters (UARTs), one Inter-Integrated Circuit (I2C), one Controller Area Network (CAN) bus, and three SPIs used by the Driver boards. This versatility enables the direct connection of a wide range of sensors and modules to the robot board, further expanding its capabilities and potential applications.

■ 5.1.4 PCB schematic

Figure 5.3 presents the schematic of the Main board PCB. Due to its size and complexity, it is recommended, for a more detailed and clearer view, to refer to the digital version of this document, available in the git repository. The digital version provides a more comprehensive view of the schematic, facilitating a better understanding of the Main board's design and functionality.

■ 5.1.5 PCB design

The Main board is constructed as a four-layer PCB, with each layer serving specific purposes. In figure 5.4, you can observe the different layers of the board. The top and middle-top layers are dedicated to the robot's ground connection, ensuring a robust and stable electrical grounding throughout the board. This ground connection helps minimize noise and interference, promoting smoother and more reliable operation. The middle-bottom layer is poured with 3.3 volts, providing a consistent power supply to the relevant components. Lastly, the bottom layer is poured with the battery power voltage coming from the fuse, serving as the primary power distribution layer for the board. This layer supplies power to the critical components such as the motors, Intel NUC, and other essential elements of the Main board.



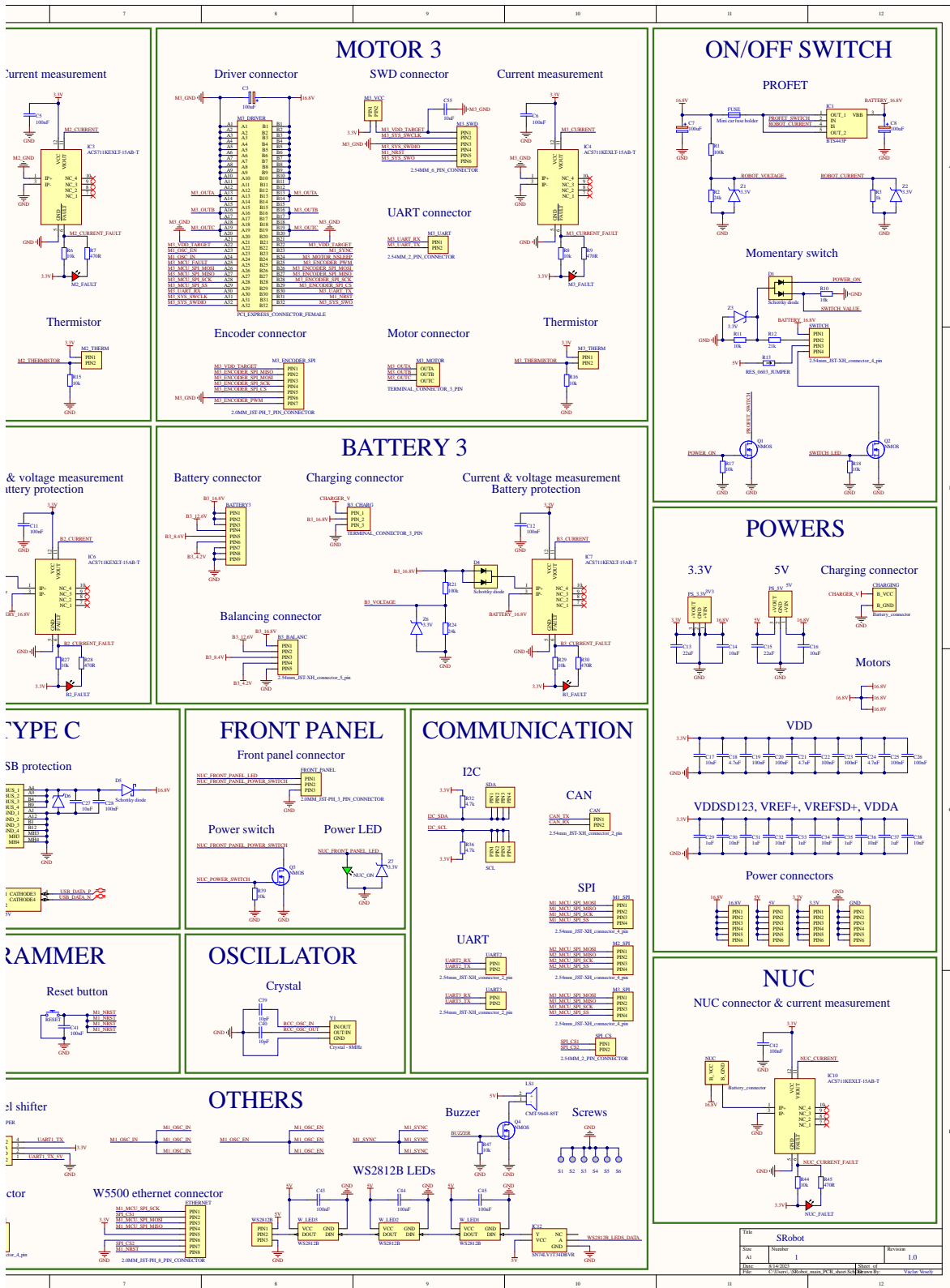
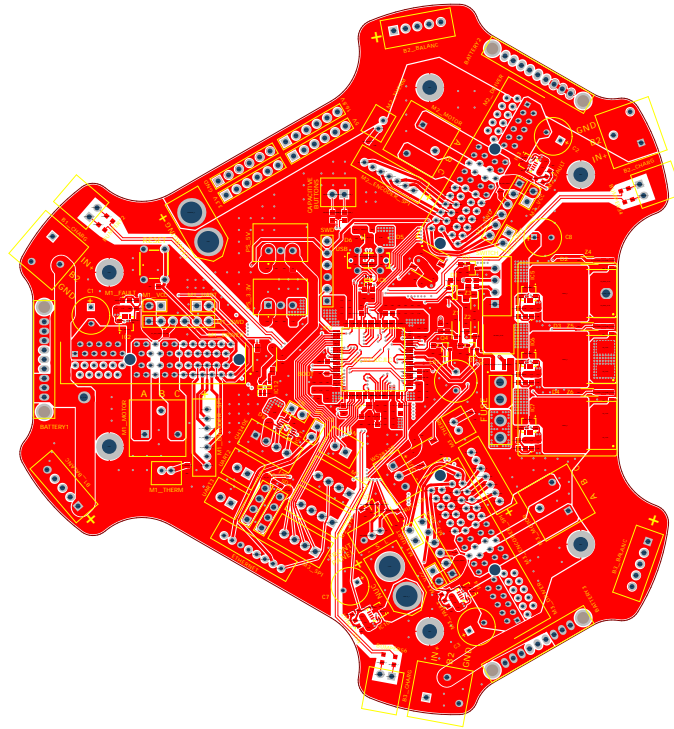
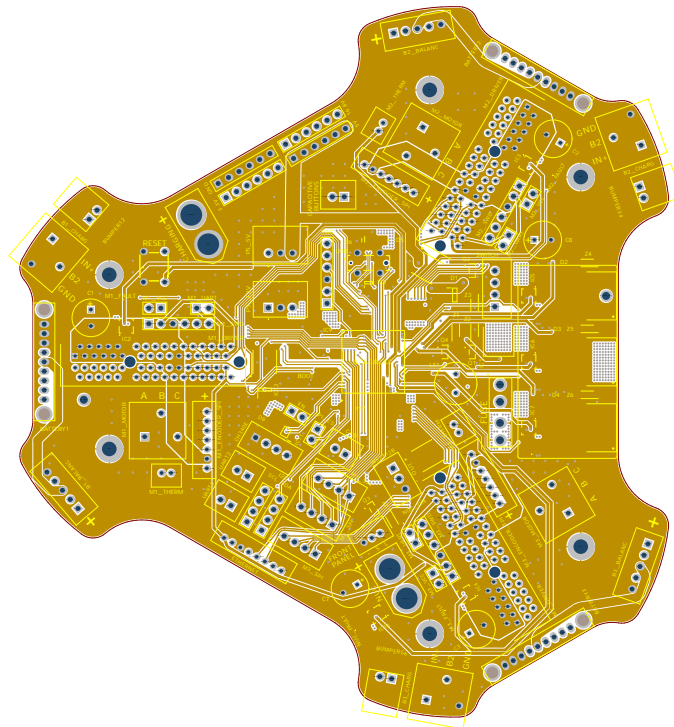


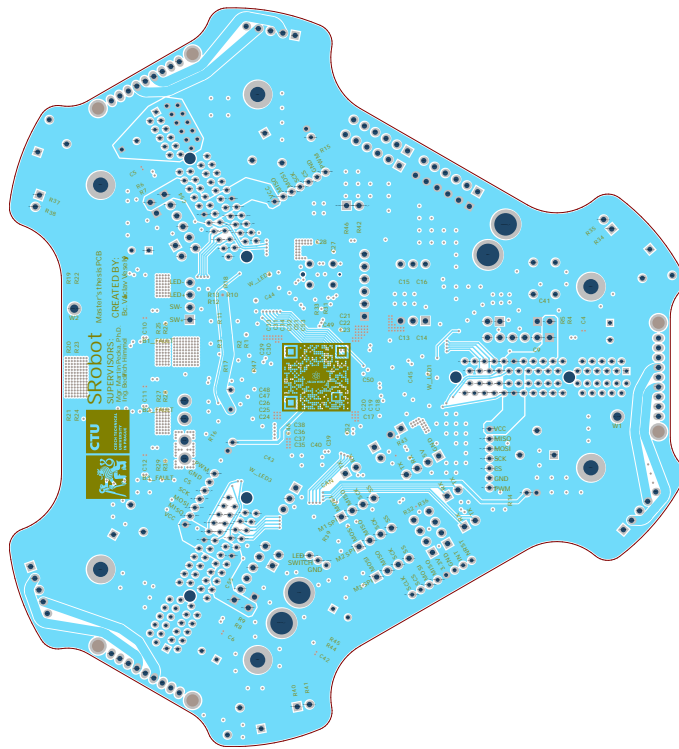
Figure 5.3: A schematic of the Main board



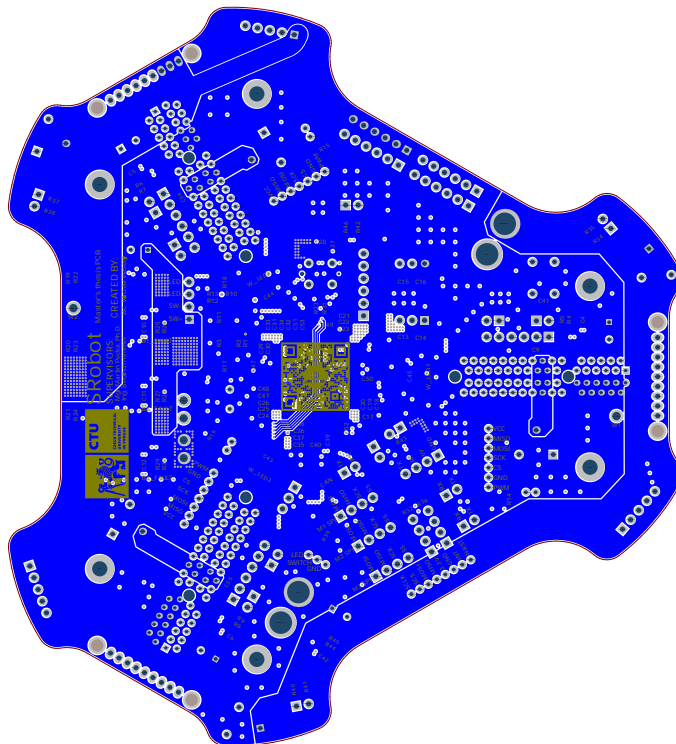
(a) : Top layer



(b) : Middle-top layer



(c) : Middle-bottom layer



(d) : Bottom layer

Figure 5.4: Different layers of the Main board PCB

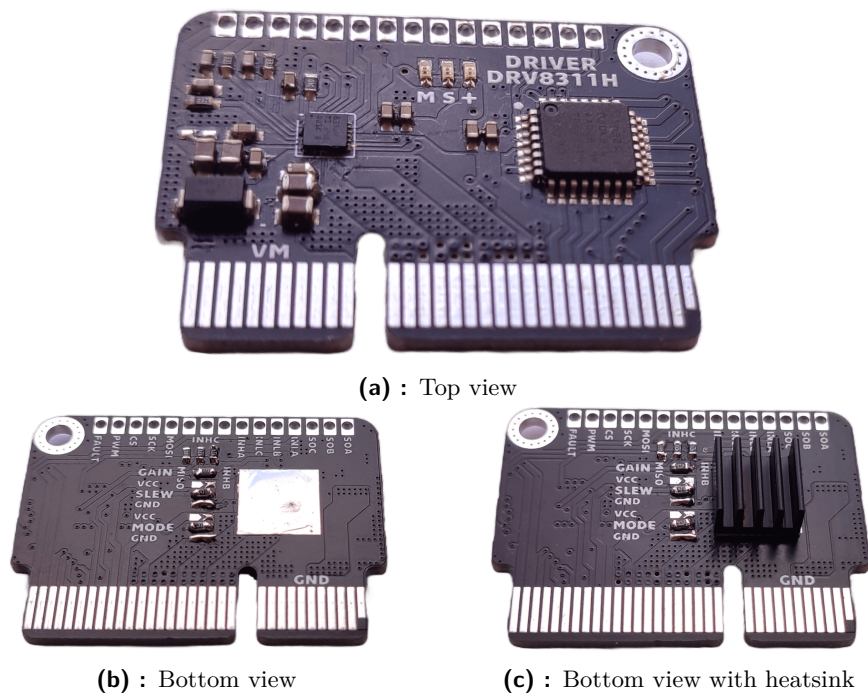


Figure 5.5: The Driver board

5.2 Driver board PCB

The Driver board utilizes a DRV8311H three-phase Pulse-Width Modulation (PWM) motor driver for controlling the BLDC motor. This motor driver features three integrated MOSFET half-H-bridges specifically designed for driving a three-phase brushless DC motor. The DRV8311H is equipped with three current-sense amplifiers with integrated current sensing, enabling accurate sensing of the three phase currents of the BLDC motors. This feature ensures optimal FOC and accurate current-control system implementation [16]. The DRV8311H motor driver operates within a recommended voltage range of 3V to 20V, with an absolute maximum voltage of 24V. It is capable of delivering high output current, with a peak current drive of up to 5 amperes. Additionally, the DRV8311H driver offers various built-in protection features, among the most important belongs overcurrent protection and thermal warning and shutdown protection. These protection mechanisms enhance the safety and reliability of the driver during operation, safeguarding the motor from potential overloads and overheating.

Moreover, the functionality of the DRV8311H motor driver can be modified by the resistors located at the bottom of the Driver board visible in the figure 5.5b. These settings include modifying the gain of the current-sense amplifier, adjusting the slew rate of the integrated MOSFET half-H-bridges, and selecting different control modes of the motor driver.

The Driver board is designed as a 32-pin PCI Express card, which can be directly inserted into the corresponding PCI Express slot on the Main board. This vertical insertion layout not only saves space but also offers convenient repairability. If any unexpected issues arise the Driver board can be easily unplugged and replaced with another, minimizing downtime and simplifying maintenance procedures. Additionally, this design choice contributes to cost-effectiveness, as using the PCI Express slot as the connection interface is more economical compared to using other dedicated PCB edge connectors. However, it is essential to acknowledge one potential drawback, which occurs when a large and heavy card is inserted. In such cases, the connector may become slightly unstable, leading to occasional disconnection of some pins. Fortunately, this is not an issue for the Driver board since it has a similar size to the PCI Express connector and is lightweight, which ensures a stable and reliable connection.

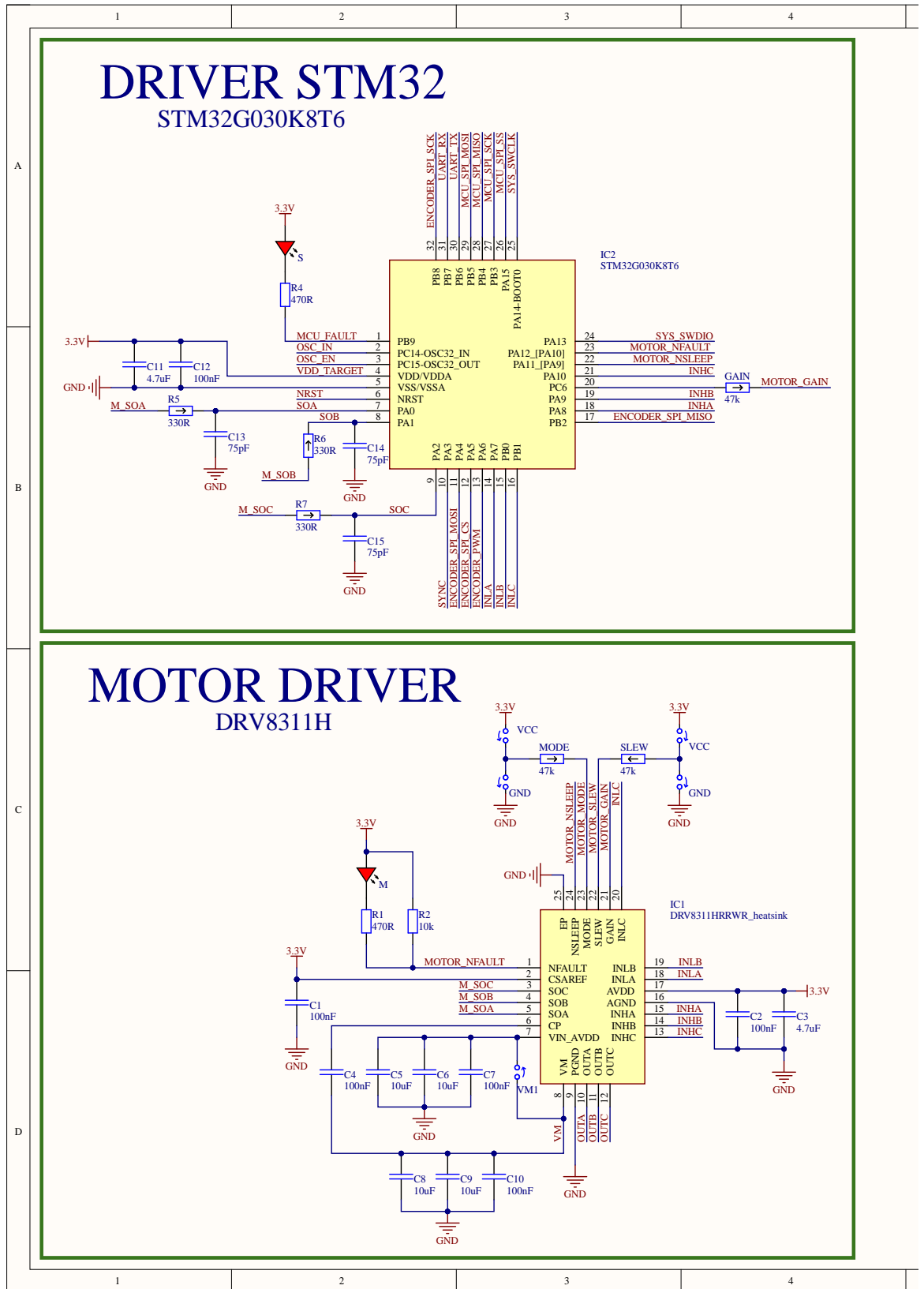
As discussed in section 5.1, communication between each Driver board and the Main board is achieved through individual SPI connections. Additionally, the boards share several common pins for synchronization and control purposes. One of these shared pins provides the Driver boards with a precise 8 MHz clock source from the Main board. This clock source ensures accurate and synchronized timing for their operations. Furthermore, two pins are dedicated to synchronization between the Driver boards. The first pin facilitates velocity synchronization, ensuring that all the motors change the speed at the same time. The second pin enables synchronized brake functionality, allowing all the motors to stop simultaneously. In addition, the boards share a reset pin, allowing for resetting of all boards simultaneously. This simplifies the reset process and enhances the overall functionality of the system. Lastly, each board shares a unique sleep pin with the Main board. This sleep pin allows the Main board to control the power supply of the motors, enabling the robot to turn the motor power on and off as needed.

■ 5.2.1 PCB schematic

Figure 5.6 presents the schematic of the Driver board PCB. Due to its size and complexity, it is recommended, for a more detailed and clearer view, to refer to the digital version of this document, available in the git repository. The digital version provides a more comprehensive view of the schematic, facilitating a better understanding of the Main board's design and functionality.

■ 5.2.2 PCB design

To enhance the cooling efficiency of the DRV8311 motor driver, the thermal pad of the driver is connected to the ground plane of the PCB, which is poured over all four layers of the board visible in figure 5.9. This ground plane acts as



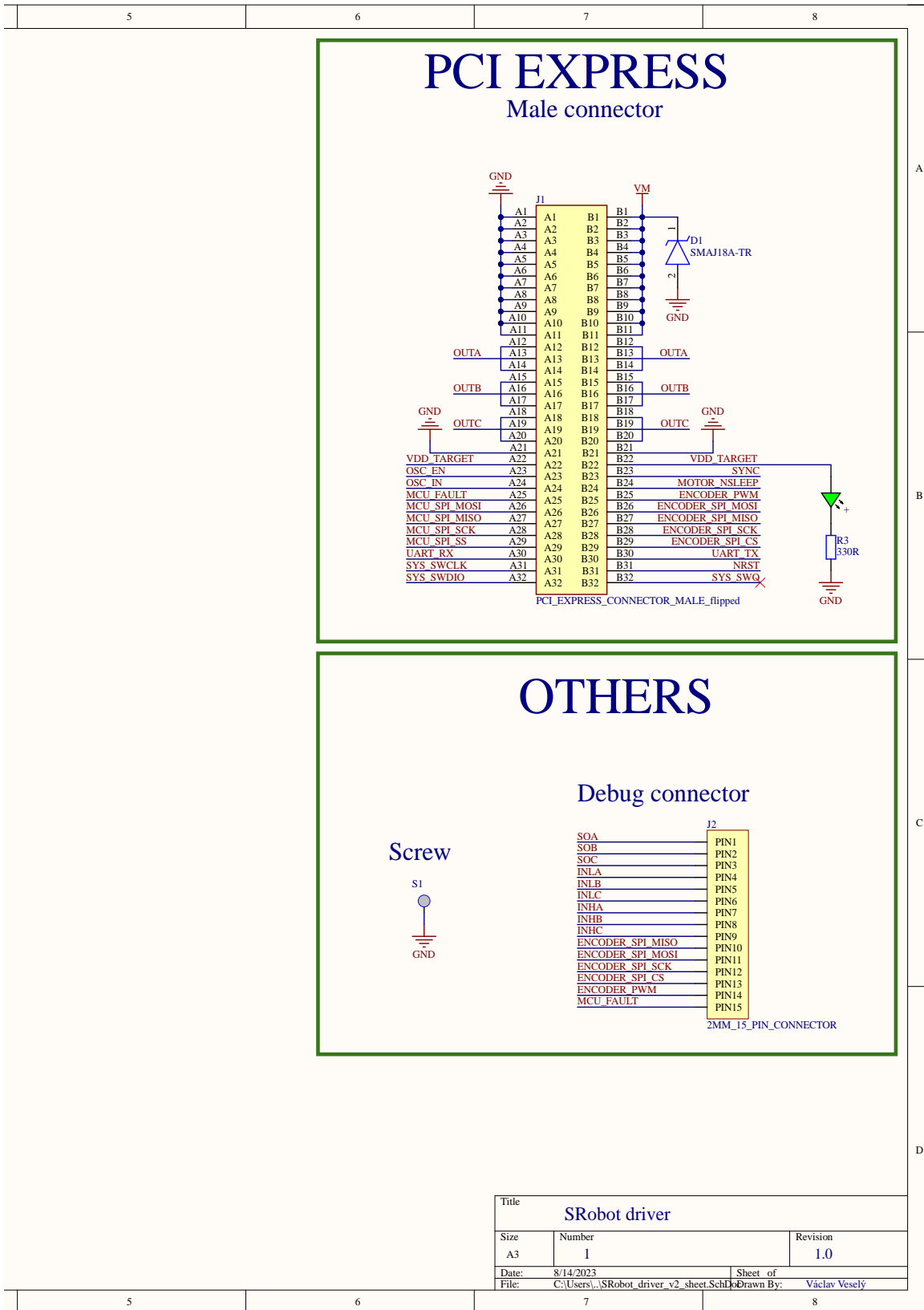


Figure 5.6: A schematic of the Driver board

Title		
SRobot driver		
Size	Number	Revision
A3	1	1.0
Date:	8/14/2023	Sheet of
File:	C:\Users\...SRobot_driver_v2_sheet.SchDoc	Drawn By: Václav Veselý

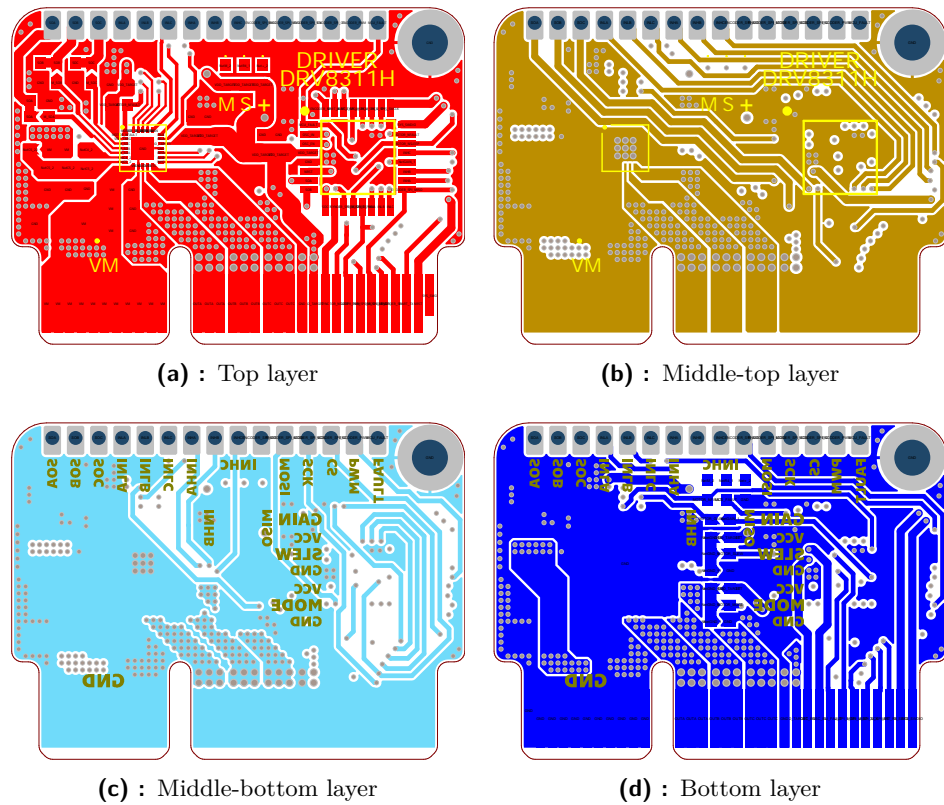


Figure 5.7: Different layers of the Driver board PCB

a large thermal conductor, aiding in the dissipation of heat generated by the motor driver. Additionally, the board has a dedicated space reserved for an 8.8 mm heatsink, positioned directly beneath the DRV8311 driver, as seen in figure 5.5. Although the current setup may not require the use of the heatsink, it becomes crucial when dealing with input voltages above 20V, as it helps further dissipate heat and maintain optimal performance. The connection of the ground plane across all layers ensures efficient thermal distribution, which enables the integrated STM32 to monitor the temperature of the DRV8311 motor driver using its built-in thermometer. This temperature monitoring mechanism helps maintain the stability and reliability of the motor driver during operation.



Figure 5.8: The Battery board

5.3 Battery board PCB

The Battery board, shown in figure 5.8, serves as a crucial link between the battery and the Main board. The board features a single connector that allows the battery to connect to the Main board. As discussed in section 4.2, this connector is capable of delivering an average current of 21 amperes, with occasional peaks reaching a maximum of 30 amperes, ensuring stable and sufficient power delivery for the robot's operation.

Once connected, the Battery board serves as an intermediary, linking the battery to both the battery balancer and the battery charger, as mentioned in section 4.1. The Battery board is securely mounted onto the battery holder using two 3 mm screws. The battery power and ground connections are welded into the Battery board using a nickel sheet plate, ensuring robust and reliable connections. Additionally, the remaining battery cells are soldered with cables to the board, completing the electrical circuit.

Apart from the main connector, there are also XT60 and XH2.54 connectors visible on the board. Although they do not have a current use in this specific configuration, they were intended for potential future expansions where the battery may serve different purposes or be connected to other components.

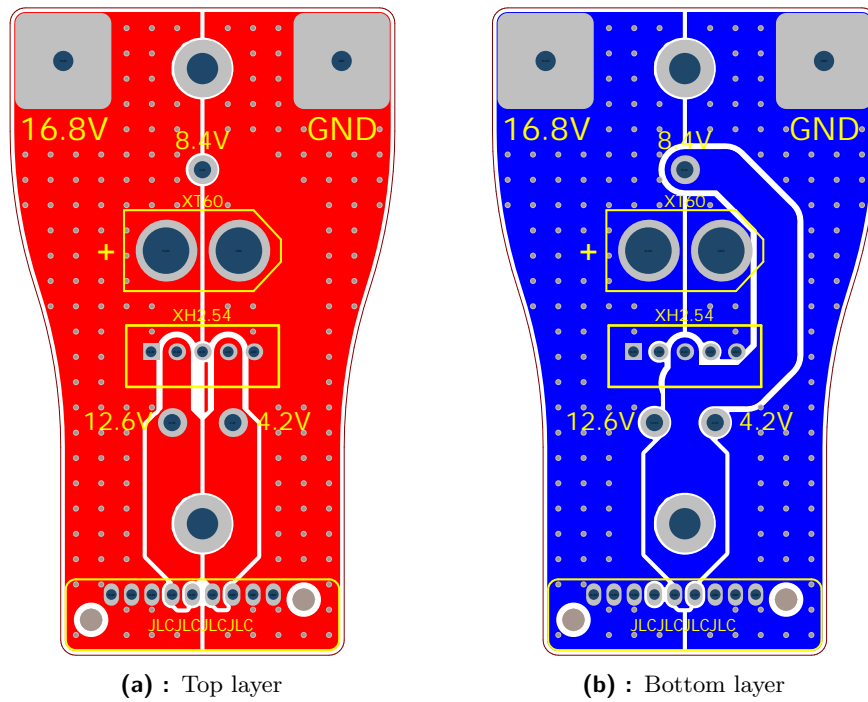


Figure 5.9: Different layers of the Battery board PCB

5.3.1 PCB design

As mentioned, the Battery board serves a straightforward purpose of connecting the battery cells to the connector that supplies power to the robot. Due to its simple functionality, the PCB design for the Battery board is kept relatively uncomplicated. The board consists of only two layers, as depicted in figure 5.9. One side of the board is dedicated to power, while the other side is dedicated to ground

Chapter 6

Electronics and control systems

6.1 Onboard computer

The robot is equipped with an Intel NUC 12 Pro Kit NUC12WSKi7 for high-level control. To ensure a secure and user-friendly connection, the NUC is attached to the robot using a self-created, easily detachable spring connector depicted in figure 6.1.

To secure the NUC to the robot, the connector is inserted into the designated space in the middle of the robot and rotated in an anti-clockwise direction until the diffuser snaps into place. This ensures a secure and stable attachment of the NUC to the robot. Conversely, to remove the NUC from the robot, the NUC needs to be raised slightly to release the diffuser from its position, followed by a careful rotation in a clockwise direction.

The diffuser is constructed from transparent thermoplastic polyester and features a plano-concave diverging lens. Both the top and bottom sides of the diffuser are covered in highly reflective acrylic mirror paint. This unique design enables effective diffusion of the LEDs light located under the bottom of the NUC to the sides, allowing it to function as a light signalization. Additionally, a mirror sticker is placed under the WS2812B LED ring, further enhancing the visibility and signaling capabilities.

For powering, the NUC is connected to the robot through an XT60 connector located at the front of the robot. This connector can provide sufficient power to energize all the connected electronics, with current limited by the 30 Ampere fuse in the robot. This setup ensures a reliable power source for the NUC and other components while protecting against excessive current.

For seamless communication between the NUC and the robot, a USB-C connector is utilized. This connection enables seamless data exchange and control between the NUC and the robot, allowing for effective high-level

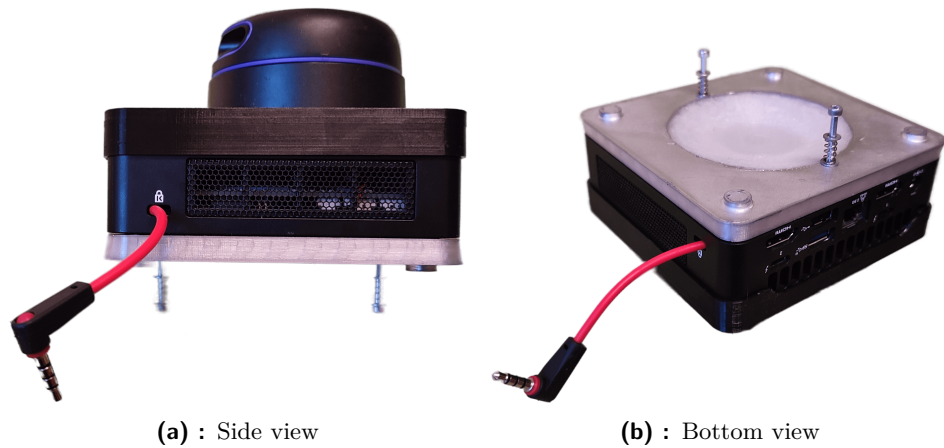


Figure 6.1: Intel NUC with LED diffusor and quickly detachable spring connector

control and operation. Moreover, in situations where no battery is inserted into the robot, and the NUC is powered by an external power adapter, the USB-C connector serves as the power supply for the robot, ensuring continuous operation and functionality.

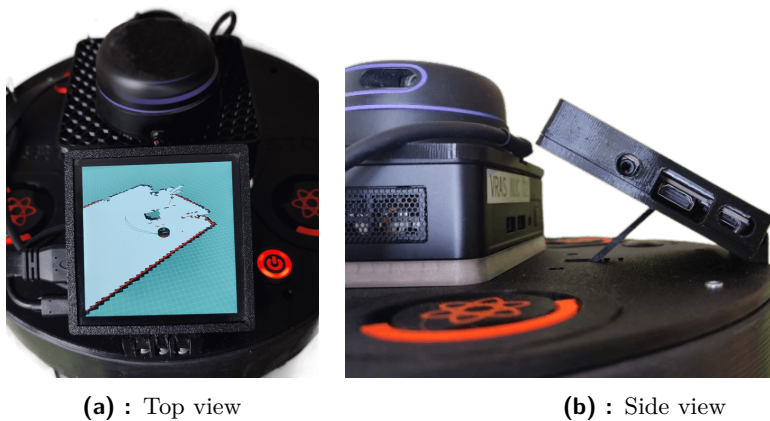
To manage the power state of the NUC, the robot utilizes the NUC's front panel connector as a control interface. The connector is integrated into the robot in the form of a 3.5 mm jack. A momentary switch, as discussed in section 9.1, is utilized to initiate the on and off commands for the NUC. This setup offers a convenient and straightforward method to manage the NUC's power status, providing the robot with efficient control over the high-level processing unit.

6.2 LiDAR sensor

The strategic positioning of the electrical components within the robot was carefully chosen to enable the placement of the LiDAR in the exact center. This layout offers several advantages to the robot's overall performance, including symmetrical measurements around the center point, balanced weight distribution and maximized sensing range in all directions. In this configuration, the robot is equipped with the RPlidar A3 depicted in figure 6.2. The LiDAR sensor frame of reference is oriented in the same direction as the robot frame of reference. This alignment ensures that the LiDAR's measurements correspond accurately with the robot's movement and orientation. To establish communication with the NUC, the LiDAR is connected through a micro USB cable.



Figure 6.2: RPlidar A3 mounted on the NUC



(a) : Top view

(b) : Side view

Figure 6.3: The Waveshare 4-inch IPS LDC display mounted on the robot

6.3 Display

The configuration includes a 4-inch IPS LCD display with a resolution of 720x720 pixels. This display offers a wide 170-degree viewing angle, ensuring optimal visibility from various perspectives. Moreover, the display features a capacitive touch screen capable of detecting up to 5 touch points simultaneously, enabling responsive and intuitive interactions. The display is securely mounted directly on the robot, as depicted in figure 6.3. The display holder allows for easy adjustment in 5-degree increments, with a maximum tilt angle of 45 degrees. This flexibility enables users to position the display according to their preference for optimal viewing. Additionally, the holder includes buttons on top, facilitating convenient control of the display's functions. To establish a connection with the NUC, the display is linked using an HDMI cable, while the capacitive touchscreen functionality is connected with USB-C.



Part II

Firmware

Chapter 7

Programming instructions

The robot features a total of four STM32 microcontrollers, with three of them being integrated into the Driver boards, and one STM32 on the Main board. These microcontrollers were programmed using the STM32CubeIDE integrated development environment, which is depicted in figure 7.1. All the codes used for programming are accessible in the attached git repository and are managed under the Berkeley Software Distribution (BSD) license.

To initiate the programming process, the user needs to connect the programmer to one of the designated connectors, as shown in figure 7.2. The programming connector highlighted in green is used for programming the Main board, while the connectors in orange are for the Driver boards. After connecting the programmer, the user can proceed with uploading the code by pressing the green run button depicted in figure 7.1. To provide a clear visual representation of the programming connector, figure 7.3 showcases its schematic. The programming pins indicated in red color represent a voltage of 3.3 volts. This setup ensures a seamless and user-friendly approach to programming the STM32 microcontrollers within the robot's architecture.

7. Programming instructions

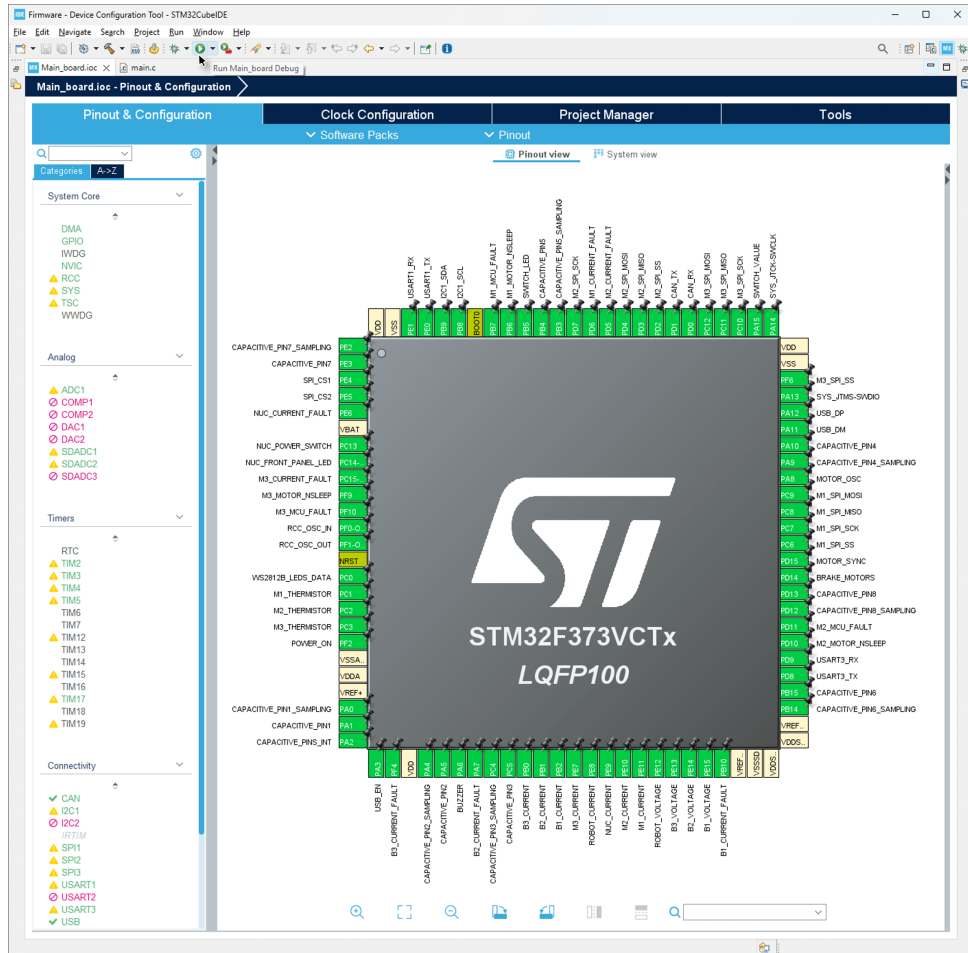


Figure 7.1: STM32CubeIDE

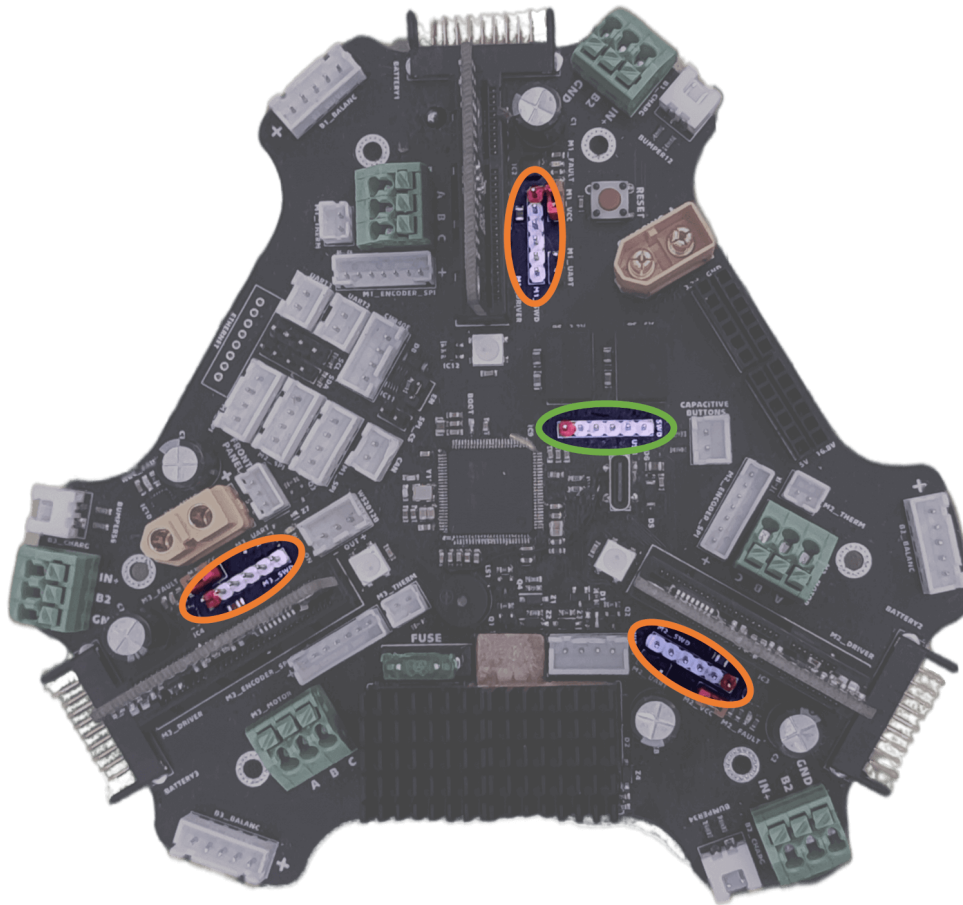


Figure 7.2: Programming connectors of all the microcontrollers located in the robot

SWD connector

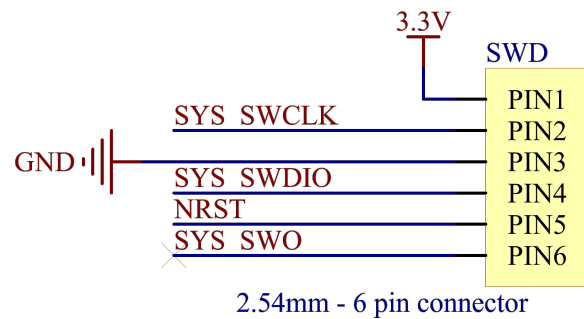


Figure 7.3: Schematic of the programming connector

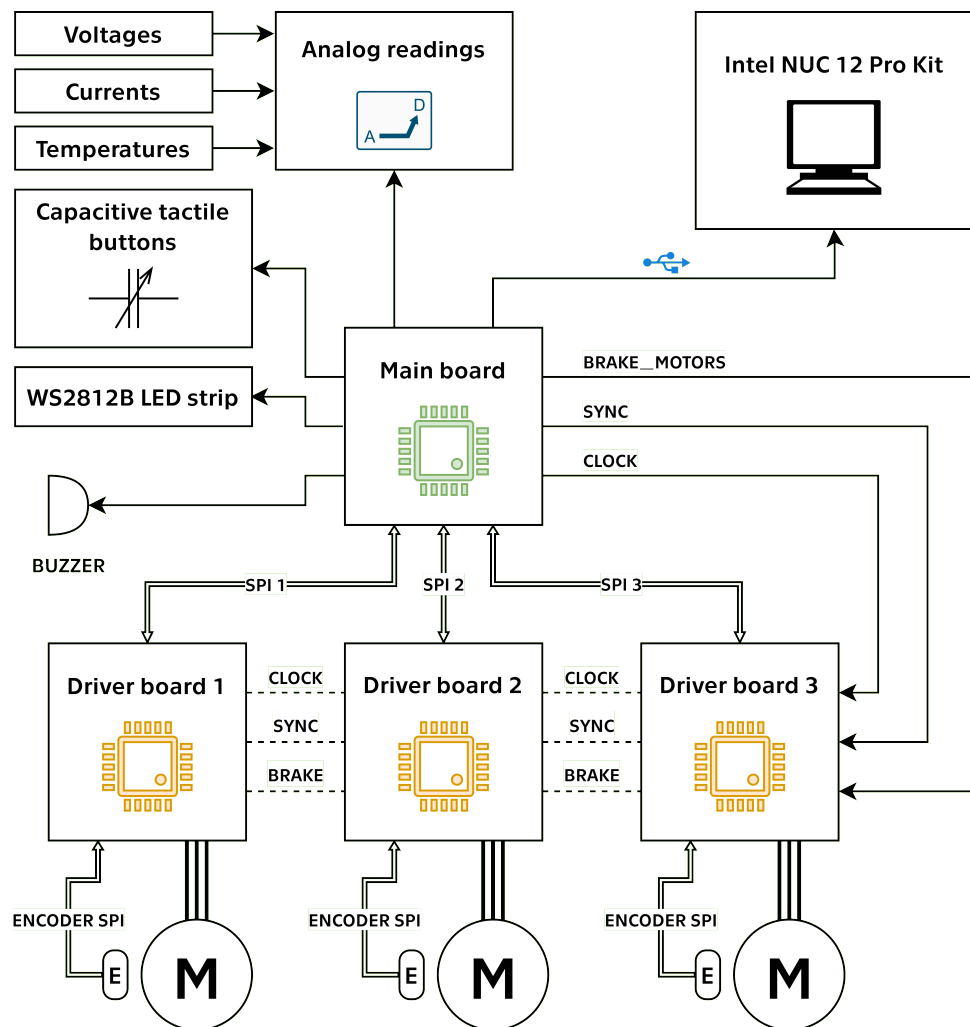


Figure 7.4: Robot connection diagram

7.1 Robot diagram

Figure 7.4 illustrates the robot's connection diagram. The diagram provides an overview of the primary communication channels, significant synchronization pins, and the various sensors integrated within the robot system. This diagram serves as a useful reference for understanding the interconnections within the robot's architecture. More information about the specific connections and the roles of individual components and pins can be found in the following sections.

Chapter 8

Driver board

8.1 Motor control

The STM32 microcontroller on the Driver board is connected to the DRV8311 motor driver in a way that enables the control of both the high-side and low-side integrated MOSFETs of the half-H-bridges. This configuration results in a 6-channel PWM control interface. The PWM channels are controlled at a frequency of 32 kHz. Although the DRV8311 motor driver is capable of sensing the current of different phases of the motor and is customized to achieve optimum FOC and current-control system implementation, these features were not fully implemented in the current setup. The DRV8311 driver measures the current on the low-side of the half-H-bridges. Therefore, the STM32 must measure the current only at the moment when the low-side MOSFET is on and the current flows through it. This type of synchronization may result in the integrated STM32 on the Driver board not being fast enough to measure the current at the right moment, leading to inaccuracies in current sensing.

As a result, the motor control is implemented using sinusoidal control with voltage torque control. This control method assumes that the torque, proportional to the set currents, is also proportional to the set voltage. While this assumption may not be entirely accurate, it is suitable for small current applications [17].

To address the current measurement issue and improve motor control accuracy and efficiency, several potential solutions may be available. One approach is to measure the current directly on the motor phases, which eliminates the need for synchronized measurements. Another option is to use a faster microcontroller or one with a higher number of Analog-to-Digital Converters (ADCs) to improve the measurement timing. Alternatively, the control frequency can be decreased to increase the synchronization window and enhance accuracy.

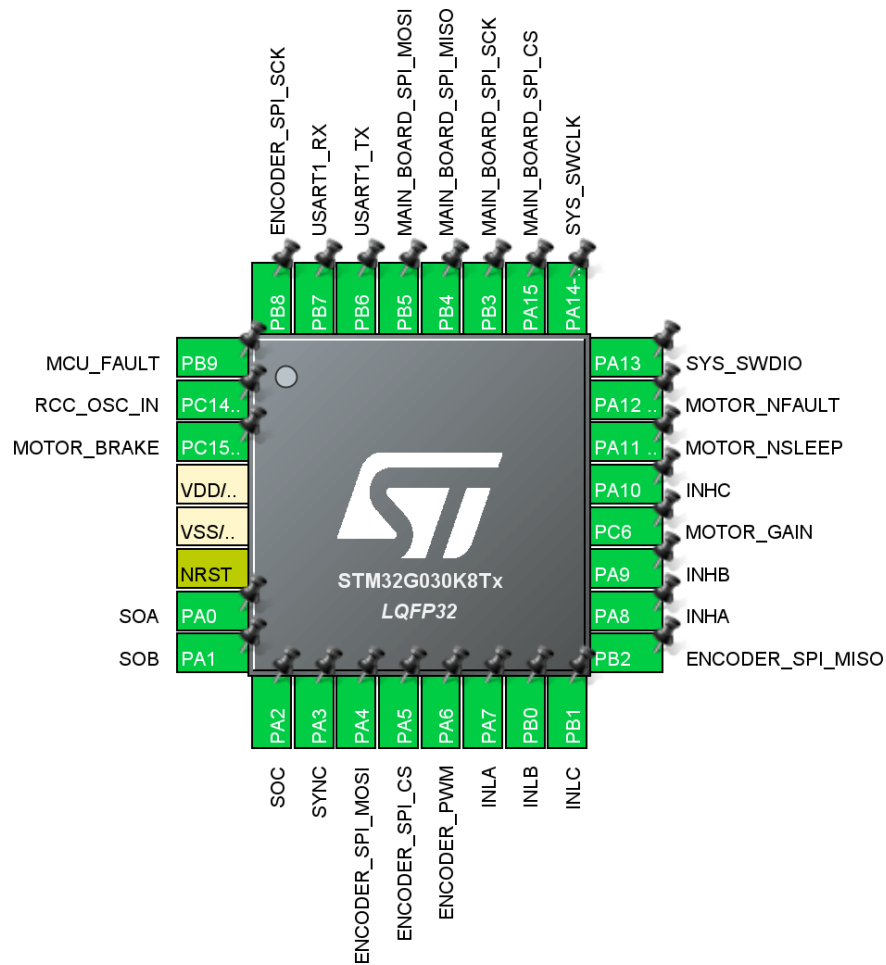


Figure 8.1: Driver board pinout

8.2 Encoder

The motor is equipped with an integrated AS5048A 14-bit magnetic rotary encoder. This encoder provides the robot with precise and accurate absolute angle information. The encoder can output this angle data either through SPI or by PWM. While both options are available on the Driver board, the SPI interface has proven to be faster and more reliable in practice. Therefore, the Driver board communicates with the magnetic encoder using the SPI protocol, with a baud rate of 16 Mbits per second. The microcontroller then uses a low-pass filter to process the raw data from the encoder and calculate the motor velocity. This process helps smooth out any noise or fluctuations in the encoder readings, resulting in more accurate velocity calculations.

The encoder readings, along with the PID controller for motor control, are performed in the main loop of the microcontroller. This implementation has proven to be the most effective and efficient solution for achieving precise

motor control based on the encoder feedback.

8.3 PID controllers

The PID controller used for the BLDC motor differs from that used for DC motors. This difference arises because the velocity of the BLDC motor is not solely dependent on the applied voltage, but also on its relative position. In the BLDC motor, the magnetic force applied to the rotor should have a 90-degree offset from its permanent magnetic field to ensure maximum torque.

As a result, the feedback loop for the PID controller is not run periodically by a timer. Instead, it runs each time the encoder reads new data, as mentioned in section 8.2. This approach allows the controller to take into account the motor's position and adjust the control signals accordingly.

Since the sinusoidal voltage control requires trigonometric functions, a lookup table is employed to speed up the calculations. By utilizing this lookup table, the main loop with all the required calculations runs at an average frequency of 35 kHz. This high-frequency operation ensures smooth and precise control of the BLDC motor, resulting in optimal performance and efficiency.

The Driver board provides two types of controllers for different operational scenarios. During normal operations, the board employs the velocity PID controller to regulate the motor's velocity and ensure smooth and accurate movements. In situations where the robot needs to brake the motors or hold a specific position, the board switches to using the position PID controller. This controller is specifically designed to provide precise control over the motor's position, enabling the robot to brake instantly and remain stationary when required.

Furthermore, the Driver board allows for dynamic adjustment of the PID controller's constants, as mentioned in the section 8.6. This feature enables users to fine-tune the motor control parameters to suit specific requirements or optimize the robot's performance under varying conditions. The ability to modify the PID constants provides a higher level of flexibility and adaptability, making the robot more versatile and capable of handling different tasks and environments.

8.4 Interrupt service routine

The Driver board effectively controls the motor by utilizing an Interrupt Service Routine (ISR) to handle various interrupts. The most critical interrupt

By providing visual feedback through the `MCU_FAULT` pin, the Driver board enhances the robot's diagnostics capabilities, allowing for better monitoring of the motor's state and the overall performance of the motor driver. This information is critical for identifying and addressing potential issues promptly, ensuring the robot operates efficiently and safely.

8.6 Communication

The communication between the Driver board and the Main board is accomplished through SPI connections, with a baud rate set to 9 Mbits per second. The message data size for this communication is 16 bits. The Driver board utilizes Direct Memory Access (DMA) for SPI communication, significantly reducing CPU overhead [18]. The DMA is configured in circular mode, enabling continuous data transfer without CPU intervention.

The Driver board operates as a full-duplex slave in this communication setup, meaning it only sends and receives data when the Main board initiates communication with it. To optimize communication efficiency, the Driver board sends and receives data in 12-byte packets. The data sent by the Driver board consists of three float values, representing the motor's velocity, position, and the Driver board's temperature.

On the other hand, the Driver board receives two float values and one `uint32_t` value from the Main board. The float values are used to set the mode of the motor and the requested motor velocity. The mode value serves to control various states of the motor, such as enabling or disabling it, setting motor power, applying brakes or setting the motor position to zero. Additionally, the `uint32_t` value sets the acceleration time of the motor, determining how many milliseconds it takes to reach the required velocity.

To configure the PID constants on the Driver board, special values are used. When the acceleration time is set to 3000000000, the P and I constants are set using the mode and velocity float values, in respective order. When the acceleration time is set to 4000000000, the mode value is used to set the D constant, while the velocity value sets the anti-windup value of the PID controller.

This communication strategy was designed to minimize the size of messages exchanged between the boards, resulting in faster communication and more efficient robot operation. By properly initializing the DMA, the Driver boards are relieved from handling the communication process actively. They only need to update the position, velocity, and temperature values periodically, while the DMA takes care of the rest. When the Main board wishes to read data from the Driver board, it simply sends a command message and receives the values accordingly. For setting data, the Main board sends a command

message with the **SYNC** pin connected to logic level voltage. After setting the pin back to the ground, the Driver boards execute the command message simultaneously, thanks to the ISR. This efficient communication mechanism streamlines the control and operation of the robot, enabling precise motor control and smooth interaction between the boards.

Chapter 9

Main board

9.1 Power management

The robot's ability to measure the current consumption of the Intel NUC provides valuable information about the NUC's power state. This feature allows the robot to detect whether the NUC is turned on, off, or in sleep mode, enhancing its overall control and autonomy.

With this knowledge, the robot can take various actions based on the NUC's power state. For example, if the NUC is turned off or in sleep mode, the robot may initiate a shutdown sequence to conserve power and prevent unnecessary operation. On the other hand, if the robot detects some unexpected movement or external stimuli through its sensors, it can wake up the NUC to respond to the situation appropriately.

By effectively monitoring the NUC's power consumption and incorporating it into the decision-making process, the robot gains the ability to respond to changing conditions and improve its efficiency and responsiveness. This feature enhances the robot's overall intelligence and adaptability, making it a more sophisticated and capable platform for various tasks and applications.

9.1.1 Momentary switch

The momentary switch on the robot serves as the primary control for turning the robot on and off. However, it is designed to prevent unintended shutdowns and provides different functions depending on how long it is pressed.

To turn the robot on, the momentary switch requires the battery voltage to exceed 12.4 volts as mentioned in the subsection 5.1.2. Once this condition is met, the microcontroller on the Main board takes control and holds the

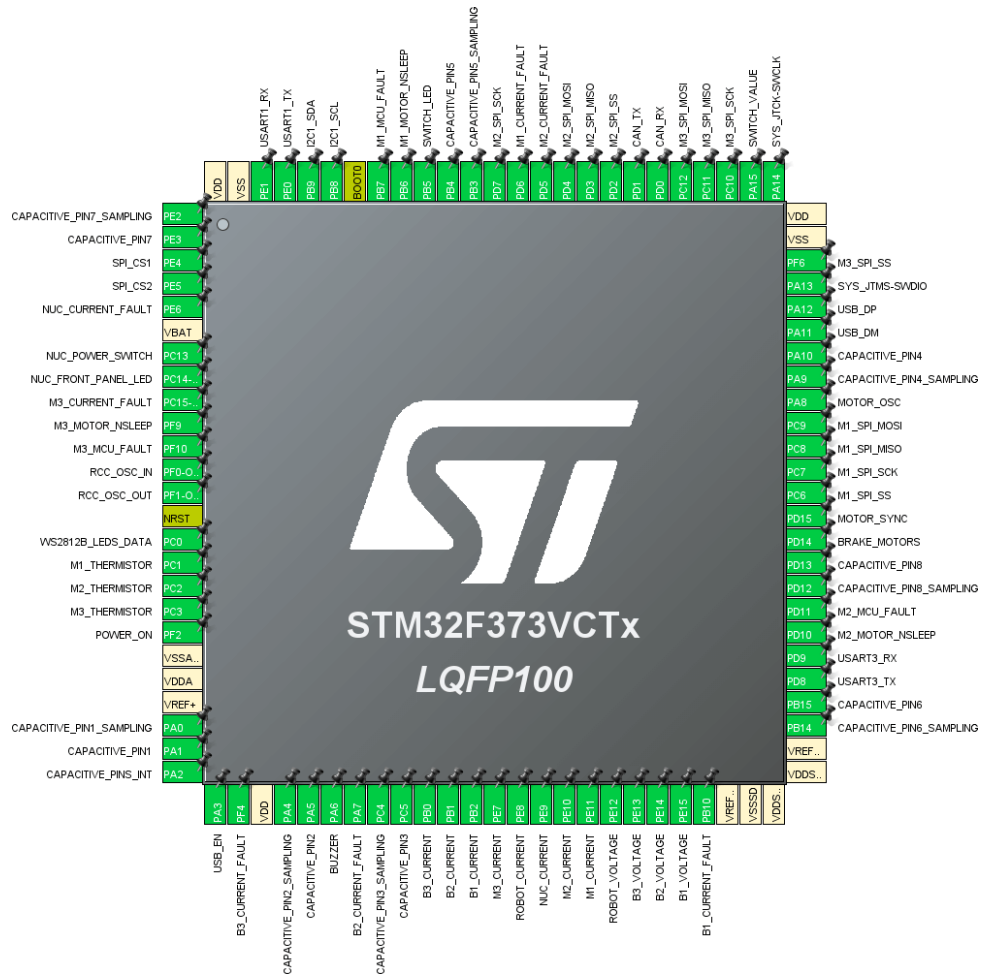


Figure 9.1: Main board pinout

robot turned on until the user initiates the power-off sequence or until the battery voltage drops below 12 volts.

To turn the robot off, the same momentary switch offers three different states to prevent unintended shutdowns. Each state corresponds to a different duration of pressing the switch. When the switch is pressed briefly, the robot initiates the reset sequence. In this state, the Intel NUC will terminate all existing nodes and reset the ROS with all the starting nodes.

If the switch is pressed for a longer duration, between one and three seconds, the robot will attempt to turn off the Intel NUC using the front panel connector. It will wait for the NUC to completely turn off by monitoring the consumed current. Once the current of the Intel NUC drops, the robot will proceed to turn itself off. If the NUC does not turn off within two minutes, the robot will automatically turn off itself.

The last state of the momentary switch occurs when it is pressed for

more than three seconds. In this case, the robot will directly turn itself off without waiting for the NUC to shut down. To provide feedback to the user, the momentary switch will indicate the duration of pressing through colored signalization and the sound produced by the robot. This multi-state functionality of the momentary switch provides a user-friendly and intuitive way to control the robot's power status.

■ 9.1.2 Front panel connector

As mentioned in section 5.1.3, the robot is connected to the Intel NUC front panel connector. This enables the robot to monitor the robot's power status and provide the robot with the ability to turn the NUC on and off as needed. When the momentary switch is pressed, the robot uses this connection to turn the Intel NUC on.

To monitor the power status of the Intel NUC, the robot reads the power status LED of the NUC. If the front panel connector is properly connected to the NUC, the robot can directly check the power status from the LED. This method provides an accurate and reliable way to determine whether the Intel NUC is turned on or off.

However, in situations where the front panel connector is not connected to the NUC, the robot needs an alternative method to detect the NUC's power status. In such cases, the robot utilizes the current consumption readings from the XT60 connector mentioned in section 3.1.2. By monitoring the current flow through this connector, the robot can determine if the Intel NUC or any other electronics connected to it are powered on, and only in such cases it will initiate the startup process.

■ 9.2 Analog data processing

The STM32 on board measures the current and voltage values every 10 microseconds. The temperature measurements are done every 100 microseconds. All these measurements are performed by a circular DMA which is started by an external trigger, in this case, a timer. This ensures efficient and continuous data acquisition without putting additional strain on the microcontroller's resources.

Once the measurements are completed, the ADC conversion complete callback is triggered, allowing the STM32 to further process the measured data. During this callback, the raw data are converted into meaningful representations of current, voltage, and temperature values.

For the capacitive tactile sensors, which cannot be measured using DMA,

the STM32 uses an ISR for data acquisition. The measurement of capacitive sensors is initialized at the start of the robot. After the measurement is finished, the TSC conversion complete callback occurs. In this callback, the measured raw data from the capacitive sensors are processed and evaluated to determine if the sensors are pressed or not. Since the capacitive sensors work by detecting changes in capacitance when pressed, an applied force to the sensors can be directly measured. After processing the data, the measurement readings are initialized again, and the whole process repeats.

To ensure accurate and stable data, a low-pass filter is applied to the measured values. The low-pass filter is designed to smooth out noise while maintaining responsiveness to changes. This filtering technique is essential for obtaining reliable and steady readings, especially in scenarios where noise or fluctuations in measurements may occur [19].

By implementing these techniques, the STM32 can efficiently and accurately monitor current consumption, robot voltages, temperatures, and capacitive tactile sensor inputs. This information is crucial for the robot to make informed decisions and respond effectively to its surroundings and user interactions. The combination of these features contributes to the robot's overall performance and enhances its ability to operate reliably and safely in various situations.

■ 9.3 Communication

■ 9.3.1 Driver boards

The communication between the Main board and the Driver boards is established through SPI. Each Driver board is connected to a dedicated SPI channel on the Main board. In this setup, the Main board acts as the master, and the Driver boards act as slaves. The communication occurs only when the Main board needs to read data from the Driver boards or change their velocity settings. The communication occurs in 12-byte packets and is described in more detail in section 8.6.

To ensure that the Driver boards are operating correctly, the Main board actively monitors their status using ISR. If something unexpected or erroneous happens with any of the Driver boards, the Main board can detect it through the respective `Mx_MCU_FAULT` pins. This allows the Main board to react to any potential issues promptly and take appropriate actions if needed.

Moreover, the Main board can synchronize velocity changes of the motors through a dedicated `SYNC` pin. This synchronization ensures that all motors change their velocity simultaneously, enhancing the robot's coordinated movement. Additionally, in case of any unforeseen circumstances or emergencies,

the Main board can initiate synchronized braking of all motors using the `BRAKE_MOTORS` pin. This feature, fully described in section 8.4, allows the robot to halt quickly and safely in response to unexpected events.

By using SPI communication, interrupts for monitoring Driver board status, and dedicated synchronization pins, the Main board maintains effective and reliable control over the motor drivers and the overall robot's motion. This level of control ensures precise and coordinated movements, enhanced safety, and efficient operation, making the robot a reliable and intelligent autonomous system.

■ 9.3.2 Intel NUC

The communication between the robot and the Intel NUC is facilitated through a USB-C port. The robot has full control over this USB connection and can handle any interruptions that may occur. If communication is interrupted, the robot can disconnect and reconnect the USB to re-establish communication with the NUC. This reconnection process prompts the NUC to restart its communication scripts, ensuring a reliable and consistent connection.

Since the USB communication is directly handled by the STM32 on the Main board without the need for any additional UART to USB module, there is no requirement to set specific baud rates. The communication between the robot and the NUC occurs at the maximum USB bitrate of 12Mbps for Full-speed mode [20]. Therefore, the set baud rate in any script will not affect the communication speed between the robot and the NUC.

To ensure effective and readable communication, the messages sent by the robot are in a specific format with the following structure " $M; x; y; z$ ". The " M " represents the type of message, and the values " $x; y; z$ " represent the data sent in that message. Different types of messages are used to send various data, including current, voltage, temperature, capacitive tactile button status, motor position, motor velocity, and reset messages. Each message type contains a specific number of values and is sent at different frequencies. The frequency at which each message is sent can be adjusted to optimize the communication.

The messages coming from the Intel NUC are in a similar format " $M x; y; z$ ". The Intel NUC has significant control over the robot's functionalities and components. It can synchronize the stopping of all motors simultaneously or stop them individually. The NUC has full control over motor velocity, acceleration, and power for each motor, as well as the ability to disable the power to the Driver boards, effectively disconnecting the motors. It can set different PID constant values for each motor to optimize its performance. Furthermore, the NUC has control over the robot's signaling elements, such as the buzzer frequency and the color of each LED located on the robot. The

robot is equipped with 27 LEDs, three of them are on the Main board and the rest under the NUC in the diffuser.

The communication between the robot and the NUC enables the NUC to initiate resets of various components, including the capacitive tactile buttons, motor positions, or encoder positions relative to the motors. The NUC has full control over the power state of the robot and can initiate its own reset or a complete shutdown sequence when necessary. This level of communication and control empowers the Intel NUC to effectively manage the robot's operations and adapt to different scenarios and tasks.

9.4 Robot protection

The Main board's direct control over the power of the Driver board provides an added layer of safety and protection for the robot. If the Main board detects increased current consumption, potentially caused by a stuck wheel or other issues with the motors, it can disconnect the DRV8311 motor driver from the power supply, preventing any further damage or overheating. This functionality can also be transferred to the Driver board itself since it shares the same `MOTOR_NSLEEP` pin, providing additional redundancy and reliability in motor control.

The robot's ability to measure the current coming from each battery and the power consumption of the Intel NUC offers valuable insights into overall power usage and system health. By continuously monitoring these parameters, the robot can detect anomalies or potential hazards, such as excessive power consumption similar to short circuits, that could lead to critical failures. In such situations, the robot can take precautionary actions, such as turning itself off, to prevent any damage to the batteries or other components.

Moreover, the robot's temperature measurements play a crucial role in its safety and protection mechanisms. By monitoring the temperature of each component, the robot can detect overheating issues. If the robot detects overheating of the Driver board or the motors, it can take preventive measures such as disconnecting the motors from power to allow them to cool down. In more extreme cases, if the internal temperature of the robot reaches unsafe levels, the robot can initiate a shutdown sequence to prevent any further heat-related issues.

Additionally, the robot continuously monitors the battery voltage to ensure it stays within safe operating limits. If the battery voltage drops below 12.3 volts, the robot will notify the user through LED signalization and sound alerts. This serves as a warning to the user that the battery is reaching a low state and may need recharging.

However, if the robot detects a critical voltage level lower than 12 volts,

it will automatically initiate a turn-off sequence. During this sequence, the robot will utilize the front panel connector to safely shut down the Intel NUC before completely powering off. This mechanism ensures that the robot will not operate in a state where the battery voltage is too low, preventing potential damage to the batteries and to the robot itself.

By implementing these safety measures and intelligent control mechanisms, the robot can proactively respond to various situations and ensure the safety and reliability of its operations. These features provide an added level of protection and peace of mind when operating the robot in various environments and scenarios.

9.5 Extensions

The high flexibility and extensive range of connectors provided by the robot's Main board make it a highly versatile and customizable platform. With three unique UARTs, one I2C, one CAN bus, and three SPIs, the robot can easily interface with various sensors and modules, allowing for seamless integration of new functionalities and expanding its capabilities. This adaptability enables users to tailor the robot to their specific needs and applications, making it suitable for a wide range of tasks and projects.

Additionally, the availability of a dedicated port for an ethernet SPI module enhances the robot's communication capabilities. With pre-connected ethernet interrupts and ground and logic voltage pins, integrating an ethernet module becomes straightforward and efficient. This allows the robot to communicate over an Ethernet network, opening up possibilities for more sophisticated control and data exchange with other devices or systems.

Furthermore, the provision of a dedicated connector for a CH340E USB to Serial adapter offers a backup solution in case the USB-C connector on the robot becomes non-functional. This redundancy ensures that the robot can maintain its communication capabilities even in the face of potential hardware issues, enhancing its overall reliability and resilience.

Overall, the Main board's extensive connector options and flexibility in accepting various modules and sensors empower the robot with the adaptability needed to meet diverse requirements and applications in research, education, or any other field where a customizable robotic platform is advantageous.



Part III

Software

Chapter 10

Robot Operating System

10.1 ROS packages

10.1.1 Robot packages

`cras_srobot_bringup`

This package includes the complete model of the robot, providing a comprehensive representation of its components and behaviors. The launch file, named `start.launch`, can initiate the robot's simulation within the custom-built Gazebo world or activate the real robot application. By default, the launch file initiates the real robot application. Additionally, this launch file is also responsible for initializing other packages referenced in this section, ensuring a coordinated and coherent startup process for the robot's functionality.

`cras_srobot_driver`

This package assumes the critical role of facilitating effective communication with the robot. It takes on the responsibility of transmitting control commands to the robot. Additionally, it publishes the received information about the robot state by a structured message called `Robot_state.msg`.

`cras_srobot_control`

This package is responsible for computing the robot's odometry by processing the received motor velocities. Moreover, it provides the crucial ability to control the movement of the robot. The package achieves this functionality by

subscribing to `PoseStamped` messages, which are obtained through the topics `/move_base_simple/goal` and `/cras_robot/goal`. Upon the reception of a goal message, this package generates the requisite control commands for the robot to navigate toward the specified destination. This capability with the help of other packages enables the robot to autonomously navigate and traverse the desired paths, enhancing its overall mobility and versatility.

■ `cras_srobot_frontier`

The primary role of this package centers on frontier exploration, with the ability to identify either the closest frontier in proximity to the robot or select a random frontier. This exploration data is shared visually through the use of `OccupancyGrid` messages, which are published via the `/cras_robot/occupied_space` and `/cras_robot/findeed_frontiers` topics. This functionality enhances the robot's ability to explore and navigate its surroundings, making informed decisions based on the availability of frontiers and the occupancy status of the environment.

■ `cras_srobot_exploration`

The main purpose of this package is to facilitate the autonomous navigation and exploration of uncharted territory by the robot. It leverages the functionality of the `cras_srobot_frontier` package to identify potential frontiers within the environment and subsequently generates a path leading to the selected frontier. The package collaborates with the `cras_srobot_control` package to ensure effective navigation along the generated path.

It's important to note that, as of the submission date, this package remains a work in progress and is not fully implemented. This indicates that further development is required to realize the complete functionality intended by this package.

■ 10.1.2 Other used packages

■ `rplidar_ros`

The purpose of this package is to establish communication with the RPLidar A3 sensor that is mounted on the robot. This LiDAR sensor offers a comprehensive 360-degree scanning field, allowing for rapid distance measurements at a rate exceeding 16,000 samples per second. The primary function of this package is to facilitate the transmission of the acquired LiDAR data [21].

The package achieves this by publishing the LiDAR scan data in the form of `LaserScan` messages, which are shared through the designated ROS topic `/cras_robot/scan`. By publishing these messages, the package enables other components of the robot's software to access and utilize the LiDAR data for various purposes, such as obstacle detection, environment mapping, and navigation.

■ `hector_mapping`

This package takes on the critical role of constructing an environmental map utilizing SLAM algorithms. Its core function is to build a representation of the surroundings, which is essential for the robot's spatial awareness and navigation capabilities [22].

To accomplish this, the package subscribes to the `/cras_robot/scan` and `/cras_robot/odom` topics. These topics are published in respective order by the `rplidar_ros` and `cras_srobot_control` packages. The package combines these two sets of data to generate a comprehensive global map of the environment. The map is represented using the `OccupancyGrid` message type and shared via the `/cras_robot/map` topic.

■ 10.2 Start up sequence

The Intel NUC is designed to power up only when it receives the appropriate signal from the robot's front panel connector. It does not automatically power on when connected to the power supply. Therefore, after the momentary switch is pressed, bringing the robot to life, the NUC remains powered off until the robot determines that the batteries have sufficient power.

After the robot decide that the batteries are charged enough, it initiates the NUC's startup process by activating the front panel connector. Once the NUC boots up, it automatically starts a robot start-up script. This script is responsible for initiating the functionality of the entire robotic platform. The start-up script is designed to demonstrate the capabilities of the robot and its various components, showcasing its potential applications and features.

■ 10.2.1 Demonstration

The startup script, as detailed in section 3.2, initiates the operational phase of the robot. It accomplishes this by launching the `start.launch` launch file, which is provided by the `cras_srobot_bringup` package. This package

After generating the map, the robot starts searching the environment by identifying the nearest frontier and creating a path for the robot to follow toward that frontier. As mentioned in the subsection , as of the submission date, the `cras_srobot_exploration` package remains a work in progress and is not fully implemented yet. Nevertheless, if the robot receives the correct path, it autonomously navigates along the path to that frontier using its odometry and the map generated by the SLAM algorithms.

The map search process can be stopped and started again using the capacitive buttons located on the robot. This provides the user with control over the robot's exploration behavior. Additionally, if the robot collides with an obstacle while searching the map, it triggers the braking mechanism to stop and avoid potential collisions. To continue the map search after encountering an obstacle, the user needs to press the START button on the robot. Once the robot completes the map search, it stops and signals this event with sound and light signalization.

Overall, the demonstration showcases the robot's autonomous exploration capabilities, map creation, and ability to navigate toward unexplored areas within its environment. The start-up script and the integrated ROS packages make it possible to execute these complex operations and demonstrate the robot's functionality in a user-friendly and controlled manner.

10.3 Robot model

For simulation and visualization purposes, a complete model of the robot was developed using the XML macro (Xacro) language. This model, depicted in figure 10.1, is designed to replicate both the kinematics and dynamics of the actual robot. It incorporates accurate values for inertia and friction to closely mimic the behavior of the real robot. Each wheel of the simulation model can be actuated using the `JointGroupVelocityController`, enabling control over their rotational speeds. Additionally, the model allows for setting the rotational speed of the LiDAR through the `JointVelocityController`, which is used primarily for visualization purposes. Furthermore, the orientation of the virtual robot and its joints corresponds precisely to the real-world robot. The velocities of the motors within the simulation are set in the same units and directions as those in the physical robot. This level of fidelity ensures that the simulation provides an accurate representation of the robot's behavior.

The simulation model provides an invaluable tool for testing and validating the robot's behaviors and algorithms in a controlled and virtual environment before implementing them on the physical robot. This level of consistency between the simulation and the real-world robot streamlines the development process and contributes to the overall success of the educational robotic platform.

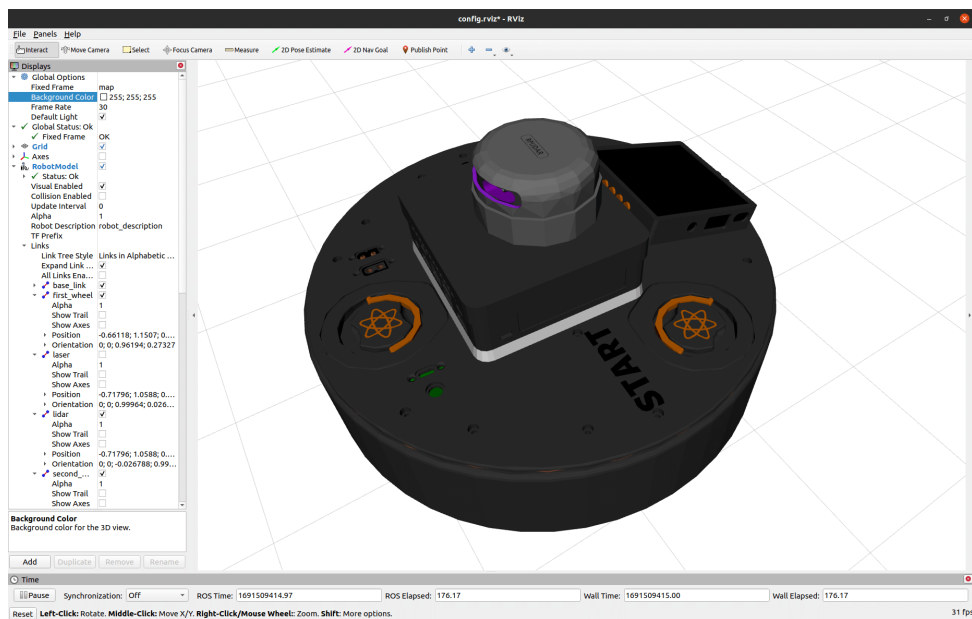


Figure 10.1: The robot model displayed in the RViz application

10.4 Robot messages

The communication between the Intel NUC and the robot involves a specific set of messages outlined in subsection 9.3.2. Upon reception, these messages are processed and subsequently published to the `/cras_robot/robot_state` topic. This topic employs a custom ROS message format called `/cras_srobot_driver/Robot_state.msg`, which comprehensively encompasses all the information the robot shares. This message format incorporates a wide range of data, including details about the robot's current consumption, battery and robot voltages, various temperature readings, capacitive tactile button states, and motor positions and velocities.

The `frame_id` field within the message header offers insight into the content of the most recent message that the NUC received. This approach allows for the integration of real-time data from the robot into the ROS ecosystem, enabling effective control, monitoring, and analysis of the robot's status and performance.

10.5 Odometry

The odometry system implemented in this robot is built upon the work conducted by Jordi Palacín from the University of Lleida, as detailed in [23]. The kinematics of the omnidirectional robotic platform is represented in figure 10.2 and mirrors the configuration of the omnidirectional mobile

robot as described in the referenced article, with the only alteration being the arrangement of the motors. In essence, the robot calculates the angular rotational velocity for each wheel based on the motion vector $M = (v, \alpha, \omega)$. In this context, v signifies the translational velocity of the displacement (in m/s), α represents the angular orientation of the displacement (in radians) with respect to the robot's reference frame, and ω indicates the angular rotational speed (in rad/s) applied to the center of the omnidirectional mobile robot. This calculation process is depicted in the equation 10.3, where the parameters $(\delta_a, \delta_b, \delta_c)$ represent the angular orientation of each omnidirectional wheel, relative to the robot reference frame. In this configuration, the angles are assigned as $(60^\circ, 300^\circ, 180^\circ)$. Parameter $R = 0.108$ represents the radial distance of the wheels relative to the center of the robot and parameter $r = 0.041$ corresponds to the radius of the wheel.

$$\begin{aligned} v_x &= v \cdot \cos(\alpha) \\ v_y &= v \cdot \sin(\alpha) \end{aligned} \tag{10.1}$$

$$IK = \begin{bmatrix} -\sin(\delta_a) & \cos(\delta_a) & R \\ -\sin(\delta_b) & \cos(\delta_b) & R \\ -\sin(\delta_c) & \cos(\delta_c) & R \end{bmatrix} \tag{10.2}$$

$$\begin{bmatrix} \omega_a \\ \omega_b \\ \omega_c \end{bmatrix} = IK \cdot \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \cdot \frac{1}{r} \tag{10.3}$$

The robot then employs the received motor angular velocities $(\omega_a, \omega_b, \omega_c)$ to compute the coordinates (x, y, θ) of the robot within the global world frame. This calculation process is described by the equation 10.6, where parameters (x_i, y_i, θ_i) represent the actual position of the robot in the world reference frame and parameter ΔT symbolizes the time lapse between the calculation measurements. This odometry system enables an accurate estimate of the robot's position and orientation, enhancing its navigational capabilities and contributing to its overall control and movement precision.

$$\begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix}_{robot} = IK^{-1} \cdot \begin{bmatrix} \omega_a \\ \omega_b \\ \omega_c \end{bmatrix} \cdot r \tag{10.4}$$

$$\begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix}_{world} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix}_{robot} \tag{10.5}$$

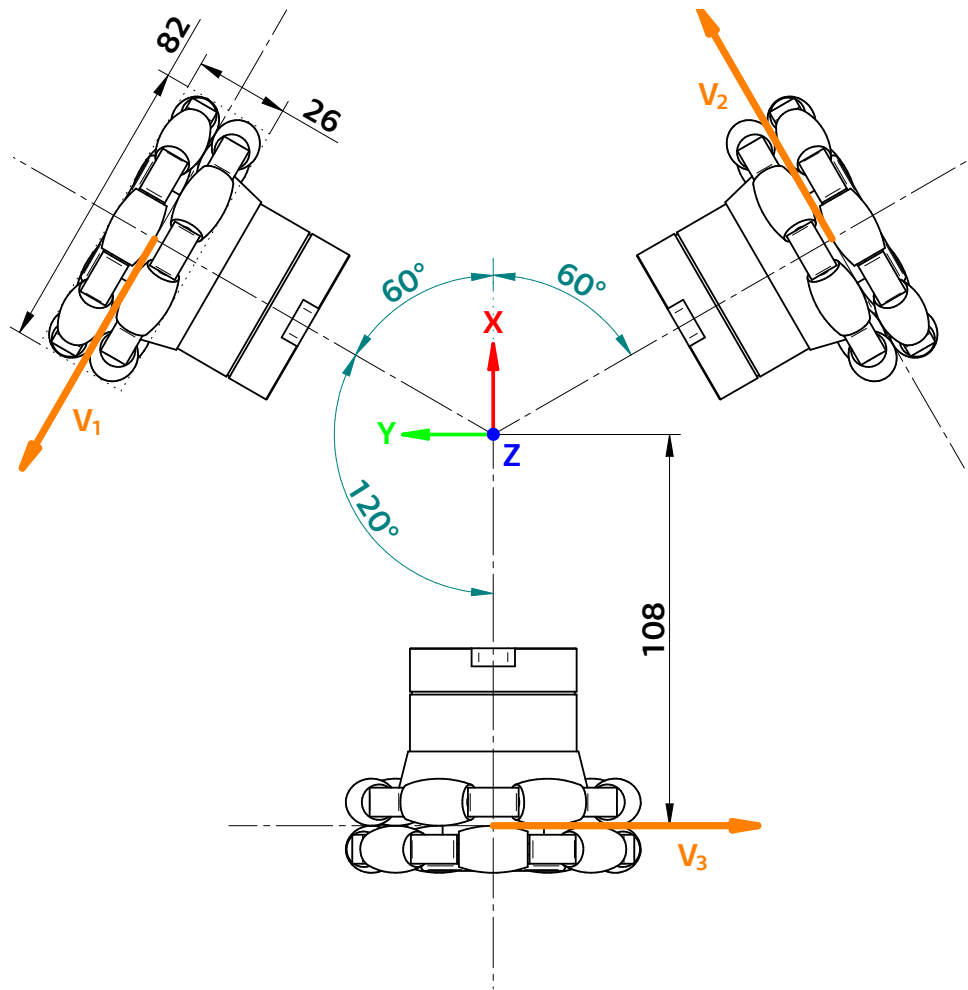


Figure 10.2: Coordinate system of the robot with the respective position of the wheels

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_{world} = \begin{bmatrix} x_i \\ y_i \\ \theta_i \end{bmatrix}_{world} + \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix}_{world} \cdot \Delta T \quad (10.6)$$

For a more comprehensive understanding of this methodology, in-depth information can be sourced from the original articles [23] and [24].

10.6 RViz

The figure 10.3 showcases the RViz application interfaced with the robot's demonstration script running. Within the image, the robot model, along with its calculated odometry, is displayed. Furthermore, the figure incorporates

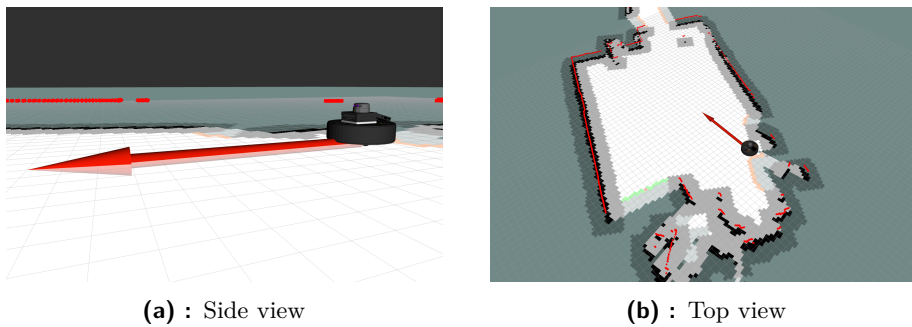


Figure 10.3: The robot visualization in the RViz application

the map data, which has been processed via the SLAM algorithms. The figure also integrates the output of the `cras_srobot_frontier` package, where the grey area denotes potential collision zones, the orange area signifies identified frontiers, and the green area represents the selected frontiers, with the central orange frontier being the one chosen for further exploration. Lastly, the visualization features real-time LiDAR data obtained from the robot sensor.

10.7 Gazebo simulator

In the development process outlined in section 10.3, a comprehensive model of the robot was constructed specifically for simulation applications. This simulated model accurately replicates both the kinematics and dynamics of the real robot. By utilizing this model, developers can first create and test their code within the Gazebo simulation environment before deploying it onto the physical robot. This approach offers several advantages, including streamlined development and the ability to identify and address potential issues in a controlled virtual setting.

To facilitate this simulation process, an entire Gazebo world was designed. Figure 10.4 showcases this Gazebo world environment, while figure 10.5 provides a close-up view of the robot within this simulated environment. The blue region visible in the figure represents the Gazebo laser sensor plugin, which simulates the behavior of the LiDAR sensor on the robot. By using Gazebo in conjunction with the robot model, developers gain the advantage of thoroughly validating their code in diverse scenarios, ensuring it operates as expected before being applied to the physical robot.

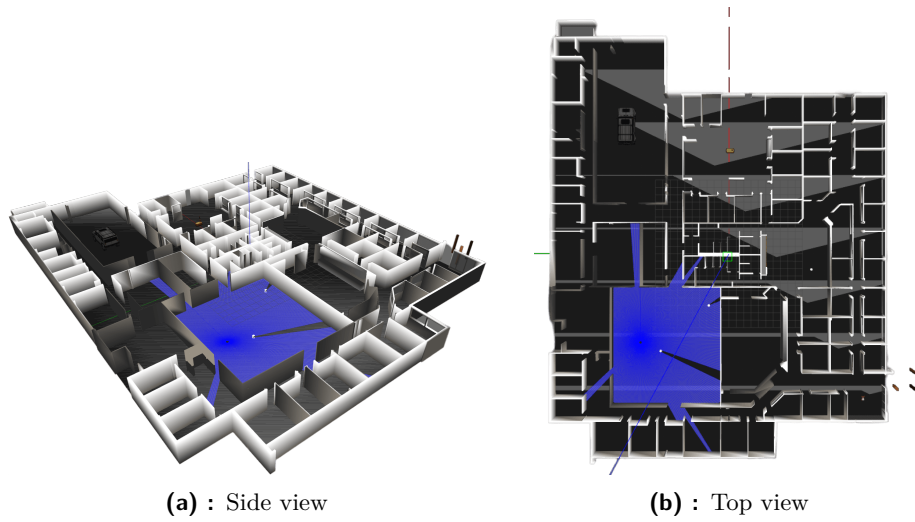


Figure 10.4: The created Gazebo world for simulation purposes

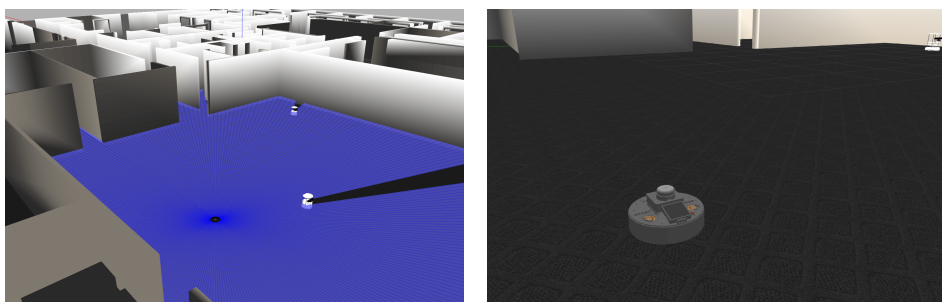


Figure 10.5: Close-up view of the robot within the Gazebo world



Part IV

Possible usage of the educational robotics platform

Chapter 11

Robotics curriculum

Annotation: This comprehensive curriculum takes students on a journey from understanding hardware and firmware basics to advanced robotic control, navigation, and perception. By providing a hands-on experience with the robotic platform, students develop a solid foundation in robotics engineering, preparing them for real-world challenges in the field.

■ Exploring Robotic Control and Perception with the Robotic Platform

1. Introduction to Robotic Hardware and Firmware

- **Module 1:** Understanding Robotic Components and Sensors
 - Introduction to the robotic platform and its hardware components
 - Understanding the role of sensors like LiDAR, encoders, and current sensors
 - Basic overview of motor drivers and power management
- **Module 2:** Basic Firmware Development for STM32 Microcontroller
 - Introduction to STM32Cube IDE and its features
 - Writing and uploading a simple "Hello World" program
 - Configuring GPIO pins for LED control
- **Module 3:** Hardware Communication and Data Exchange
 - Introduction to SPI communication between Main board and Driver boards

- Writing firmware to exchange data between boards
- Utilizing DMA for efficient data transfer
- **Module 4: BLDC Motor Control Concepts**
 - Introduction to BLDC motor operation and principles
 - Exploring different control methods: voltage control, current control, FOC
 - Programming STM32 to control BLDC motor using different methods

2. Advanced Robotic Control and Navigation

- **Module 5: PID Controller Implementation**
 - In-depth understanding of PID control theory
 - Writing firmware to implement a PID controller for motor velocity
 - Tuning PID gains for optimal performance
- **Module 6: Advanced Motor Control Techniques**
 - Implementing (FOC) for precise motor control
 - Understanding Clarke and Park transformations
 - Comparing FOC with other control methods
- **Module 7: Robot Motion and Odometry**
 - Introduction to robot kinematics and odometry calculation
 - Writing firmware to compute robot's position using wheel encoders
 - Visualizing robot's trajectory in RViz
- **Module 8: SLAM Algorithms and Navigation**
 - Understanding Simultaneous Localization and Mapping (SLAM)
 - Exploring SLAM algorithms: GMapping, Hector SLAM
 - Implementing SLAM in the robotic platform

3. Robotics Perception and Advanced Applications

- **Module 9: Frontiers Detection and Exploration**

- Understanding the concept of frontiers in exploration
- Programming the robot to identify and navigate towards frontiers
- Visualizing frontiers on the map using RViz
- **Module 10:** Path Planning and Following
 - Introduction to path planning algorithms: A* and Dijkstra
 - Writing firmware to plan paths and follow them using motor control
 - Simulating path planning in Gazebo
- **Module 11:** Robot Communication and Integration
 - Implementing communication protocols for robot-NUC interaction
 - Programming the Intel NUC to control the robot's actions
 - Integrating ROS topics and services for seamless communication

4. Final Projects and Applications

- **Module 12:** Project - Autonomous Navigation Challenge
 - Students implement autonomous robot navigation through a complex environment
 - Utilizing SLAM, frontiers detection, and path planning algorithms
- **Module 13:** Project - ROS-based Human-Robot Interaction
 - Students develop a human-robot interaction scenario
 - Programming the robot to respond to user commands and gestures
- **Module 14:** Project - Advanced Perception and Manipulation
 - Exploring advanced topics like object detection using neural networks
 - Integrating robotic arm control and manipulation

Assessment and Evaluation:

- Quizzes and assignments to evaluate theoretical understanding
- Hands-on programming tasks to assess practical skills
- Project-based assessments to test the application of concepts
- Final projects showcasing a comprehensive understanding of the robotic platform

Chapter 12

Example lessons

12.1 Introduction to BLDC Motor Control Using the Robotic Platform and STM32Cube IDE

Objective: The objective of this lesson is to introduce students to the basic concepts of BLDC motor control using the robotic platform and STM32Cube IDE. By the end of the lesson, students should be able to upload a code to the STM32 microcontroller, implement a PID controller, and control the BLDC motor's velocity.

Prerequisites:

- Basic understanding of microcontrollers
- Familiarity with C programming language

Duration: 2 hours 20 minutes

Materials Needed:

- Robotic platform with BLDC motor
- STM32Cube IDE installed on a computer
- USB cable for connecting the Robotic platform to the computer
- STM32 debugger for programming the microcontroller

Lesson Plan:

- Introduction to BLDC Motors and Motor Control (15 minutes)

■ ■ 12.1. Introduction to BLDC Motor Control Using the Robotic Platform and STM32Cube IDE

3. Show how to calculate error, proportional, integral, and derivative terms.
 4. Explain how to adjust PID constants for optimal performance.
- Testing Motor Control with PID (20 minutes)
 1. Connect the robotic platform to the computer using the USB cable.
 2. Demonstrate how to establish communication between the computer and the STM32 microcontroller.
 3. Showcase the process of sending velocity setpoints to the microcontroller.
 4. Run the uploaded code and observe the motor's response to different setpoints.
 - Hands-On Experiment (30 minutes)
 1. Divide students into pairs or small groups.
 2. Provide each group with a task to accomplish using the robotic platform.
 3. Tasks could include setting specific velocities, implementing velocity ramps, or demonstrating the effect of different PID constants.
 4. Encourage students to modify the code and experiment with different scenarios.
 - Recap and Discussion (10 minutes)
 1. Gather the students for a recap of the lesson's key concepts and activities.
 2. Facilitate a discussion on the challenges faced, insights gained, and the significance of understanding motor control.

Homework Assignment: Assign students a task to modify the PID constants and observe how it affects the motor's behavior. Encourage them to document their observations and insights.

Assessment: Evaluate students based on their ability to upload code to the STM32 microcontroller, implement a PID controller, and successfully control the BLDC motor's velocity. Assess their understanding through discussions and responses to questions related to motor control concepts.

12.2 Introduction to Odometry and Robot Motion Control using a Mobile Robot Platform

Objective: To familiarize bachelor students with the concept of odometry and how it is used for robot motion control through practical experimentation with a mobile robot platform.

Prerequisites:

- Basic understanding of robotics, kinematics, and programming.

Duration: 2 hours

Materials Needed:

- Mobile robot platform with omnidirectional wheels
- Laptop with ROS installed
- Gazebo simulation environment
- RViz visualization tool
- Sample code for odometry and motion control

Lesson Plan:

- Introduction (15 minutes):
 1. Briefly introduce the concept of mobile robotics and its applications.
 2. Explain the importance of odometry in estimating the position and orientation of a robot in an environment.
 3. Discuss the significance of motion control in enabling robots to navigate and perform tasks.
- Theory of Odometry (20 minutes):
 1. Present the mathematical foundation of odometry and its components (linear displacement, angular orientation, and rotational speed).
 2. Discuss the challenges of odometry, including errors and drift.
 3. Explain how odometry can be calculated using wheel encoders and the motion model of the robot.

- Hands-On Experiment (60 minutes):
 1. Divide the students into pairs or small groups.
 2. Provide each group with the mobile robot platform and laptops.
 3. Guide the students through the steps of launching the Gazebo simulation environment and RViz visualization tool.
 4. Assist the students in loading the pre-configured robot model and environment into Gazebo.
 5. Help the students run the sample odometry and motion control code on the robot platform. The code should simulate the robot's movement and display the estimated position and orientation in RViz.

- Experiment Steps:
 1. Initialize the robot's starting position in the simulation.
 2. Implement code to control the robot's wheels and calculate odometry.
 3. Drive the robot in different directions and distances using appropriate commands.
 4. Observe how the robot's estimated position changes based on the calculated odometry.
 5. Discuss the impact of errors and inaccuracies in odometry on the robot's trajectory.

- Discussion (20 minutes):
 1. Gather the students and encourage them to share their observations and experiences during the experiment.
 2. Facilitate a discussion on the challenges encountered and the differences between calculated odometry and ground truth.
 3. Emphasize the importance of sensor calibration and error correction in accurate odometry estimation.
 4. Highlight the relevance of odometry in real-world robot navigation and localization tasks.

- Wrap-up (5 minutes):
 1. Summarize the key concepts learned during the lesson.

2. Reinforce the connection between odometry, robot motion control, and navigation.
3. Encourage students to explore more advanced concepts in mobile robotics and to conduct independent experiments using the provided robot platform.

Homework Assignment: Ask students to research and prepare a short presentation on methods for improving odometry accuracy, such as sensor fusion, SLAM algorithms, and loop closure detection.

Assessment: Evaluate students based on their engagement during the hands-on experiment, their ability to articulate the concepts of odometry and motion control and their contributions to the group discussion.

■ 12.3 Exploring SLAM Algorithms and Frontiers Detection using the Robotic Platform

Objective: The goal of this lesson is to introduce students to the fundamental concepts of SLAM (Simultaneous Localization and Mapping) algorithms and frontiers detection. By the end of the lesson, students should be able to understand how SLAM works, identify frontiers on a map, and simulate the process in Gazebo.

Prerequisites:

- Basic understanding of robotics and sensor concepts
- Familiarity with ROS (Robot Operating System) basics
- Familiarity with Gazebo simulation environment

Duration: 2 hours

Materials Needed:

- Robotic platform with a LiDAR sensor
- Gazebo simulation environment
- Computer with ROS and Gazebo installed

Lesson Plan:

- Introduction to SLAM and Frontiers Detection (15 minutes)
 1. Provide an overview of the SLAM concept, explaining its significance in robot navigation and mapping.
 2. Introduce the concept of frontiers as unexplored areas that a robot can identify and navigate towards.
- Overview of Robotic Platform and LiDAR Sensor (10 minutes)
 1. Present the robotic platform and its key components, including the LiDAR sensor.
 2. Explain how LiDAR sensors work and their role in environment perception and mapping.
- Understanding SLAM Algorithms (30 minutes)

Homework Assignment: Assign students a task to create a ROS package for map exploration that will plan the shortest path to the selected frontier and enable them to search the entire map.

Assessment: Evaluate students based on their understanding of SLAM algorithms, ability to identify frontiers and successful navigation in Gazebo simulation. Assess their participation in group discussions and their ability to analyze simulation results.



Chapter 13

Conclusions

The master's thesis has comprehensively explored the design, construction, firmware development, and software implementation of the SRobot, a versatile indoor mobile robotic platform explicitly designed as an educational tool for university students. The platform aims to impart fundamental robotics concepts encompassing both hardware and software aspects.

The robot's overview highlighted numerous distinct advantages over comparable robots. Foremost among these are the robot's modular design, extended operational lifetime, and innovative capacitive tactile bumper, which endows the platform with true omnidirectional capabilities. The integration of BLDC motors not only ensures enhanced precision and smoother movement but also contributes to energy efficiency. Remarkably, the SRobot stands out for its affordability, with a significant portion of the cost attributed to the Intel NUC and RPLidar A3 components.

The construction process provides a comprehensive guide for replicating the educational platform. It meticulously outlines the selection and functionality of each component. It elucidates the purpose and structure of the capacitive bumper and delves into battery selection, balancing, and charging processes. The detailed PCB design with all integrated components is fully described and visually presented, providing a clear blueprint for assembly.

The firmware part offers insights into the roles of the STM32 microcontrollers, providing clear instructions for code upload to each board. The robot connection diagram is elucidated, enhancing the comprehension of the communication infrastructure. BLDC motor control, including the use of built-in encoders, is thoroughly explained, as is the PID controller's role in ensuring smooth motor control. The firmware part delves into the various interrupts managed by the ISR, elucidates communication protocols between each board and underscores the value of multiple analog measurements. The purpose of the momentary switch and the front panel connector is clarified, along with the communication between the robot and the onboard computer.

The thesis also highlights the protective features offered by the board and underscores the high flexibility and wide array of connectors available to accommodate diverse modules and sensors.

The software part provides insight into the ROS packages employed by the robot, explaining topics and messages within each package. It outlines the startup sequence and launched terminals, presenting a comprehensive perspective on the robot's functioning. The creation and utilization of the robot model for simulation and visualization purposes were explained, along with the processing and publishing of received messages from the robot. The implementation of odometry is thoroughly explained, complete with the relevant equations and the established robot coordinate system. The RViz application, running alongside the demonstration script was demonstrated and the importance of Gazebo simulation, depicted through the simulation world, was emphasized.

The last part emphasizes the suitability of the robot as an educational platform. It showcases a range of educational use cases and topics that could be taught using the platform and offers a comprehensive robotics curriculum focused on the robot as an educational tool. Additionally, the provided example lessons, covering topics from firmware implementation for each board to the utilization of SLAM algorithms for exploration purposes, solidify the value of the robotic platform as a versatile and effective educational platform.

In conclusion, the successful fulfillment of this master's thesis assignment has resulted in the creation of the SRobot, an exemplary indoor mobile robotic platform. Its modular design, hardware capabilities, sophisticated firmware, and versatile software make it an ideal educational tool. The inclusion of example lessons further enhances its value, making the SRobot a valuable resource for teaching robotics concepts to university students.

13.1 Future improvements

13.1.1 Hardware

The current design of the system exhibits certain imperfections, which require focused attention in future iterations. One prominent issue pertains to the battery charging and balancing mechanism, necessitating a redesign to ensure proper functionality. The placement of the Schottky diode directly on the board is crucial, and a separation of the power supply ground from the robot ground connection is essential. Implementing low voltage and current protection for each battery is also a recommended enhancement.

Furthermore, the STM32 on the Main board, responsible for USB communication with the onboard computer, necessitates a $1.5\text{ k}\Omega$ pull-up resistor

connected to the DM channel to operate properly. Additionally, the DM channel requires hardware pull-up post the STM32 initialization. This challenge could potentially be addressed by adopting different STM32 microcontroller versions that manage this autonomously, eliminating the need for specific enable pins.

Furthermore, enabling the TSC necessitates the use of ISR. Regrettably, the current interrupt allocation conflicts with one of the UART peripherals. This problem would need to be put into consideration for the next design.

To streamline the design, the connector for the balancing circuit will be replaced with a 2 mm JST-PH connector, minimizing unnecessary cable crimping and enhancing reliability.

One significant enhancement planned for future designs is the inclusion of an onboard inertial measurement unit on the Main board, addressing the current lack of this feature. This addition will significantly improve the robot's capabilities, particularly improving the measured odometry accuracy.

Another pivotal modification in future designs involves replacing the PCI Express connector with a more robust edge card connector. The current connector lacks secure connectivity and may result in pin disconnection due to deflection, primarily affecting larger boards. This issue does not impact the current lightweight Driver board setup.

Consideration is also given to potentially upgrading the STM32 microcontroller on the Driver boards or introducing current measurement sensors between the motor phases. This enhancement would enable FOC control for the motors, enhancing their performance.

Lastly, relocating the battery current measurement sensors between the charger and the Schottky diode is envisioned. This alteration would enable the robot to distinguish between charging and non-charging states. During the charging process, the robot could measure charging current and battery voltages, leveraging LED visualization to provide insights into the ongoing charging process. This process would contribute to the system's overall usability and functionality.

■ 13.1.2 Software

The software aspect of the robot also requires further attention to unlock its full potential. Key areas of focus for future development include enhancing the `rosserial` package, which will enable the robot to communicate over a standardized ROS protocol. This enhancement is essential for seamless integration with ROS, facilitating more efficient data exchange and interaction.

An essential improvement will involve the creation of a ROS `move_base` for

the robot. This addition would grant the robot access to various existing ROS navigation and exploration packages, significantly expanding its capabilities.

Additionally, efforts will be directed toward enabling ssh communication and the option to attach to the tmux terminal sessions. This feature will provide a convenient means to monitor the robot's state and the status of each package, enhancing the debugging and management process.

The final planned update for the robot involves moving the computation of odometry directly onto the robot itself. Integrating onboard odometry calculations, combined with data from the IMU, will result in more precise measurements and reduced drift. This enhancement ensures the robot's navigational accuracy and enhances its overall performance.

13.2 Resources

All the resources, including CAD models, PCB designs, firmware, and software codes related to this project, are readily accessible within the attached Git repository [25]. These valuable assets are made available under the BSD and MIT licenses, ensuring openness and flexibility in their usage. The PCB designs are covered by the Creative Commons license.

For those seeking to explore the repository further or access any specific details, please follow this link:

<https://gitlab.com/WenceslawCZ/design-and-construction-of-an-omnidirectional-base-for-student-robotics-platform>

For any additional information or potential inquiries regarding the SRobot, please feel free to contact me directly via email at vaclav@vesely.pro. I am more than willing to provide insights and answer any questions about this educational platform, its design, capabilities, and potential applications.



Appendices



Appendix A

Acronyms

- ADC** Analog-to-Digital Converter. 53, 61
- BLDC** Brushless Direct Current. 1, 7, 12–14, 27, 36, 55, 82, 85–87, 95
- CAN** Controller Area Network. 31, 65
- DMA** Direct Memory Access. 57, 61, 82
- FOC** Field-Oriented Control. 14, 36, 53, 82, 97
- I2C** Inter-Integrated Circuit. 31, 65
- ISR** Interrupt Service Routine. 55, 56, 58, 62, 95, 97
- LiDAR** Light Detection and Ranging. 21, 44, 70–73, 77, 81, 91, 92
- NMOS** N-channel metal-oxide-semiconductor. 30
- PCB** Printed Circuit Board. 1, 22, 27, 31, 37, 42, 95, 98
- PCI** Peripheral Component Interconnect. 30, 36, 37, 97
- PID** Proportional–Integral–Derivative. 1, 54–57, 63, 82, 85–87, 95
- PROFET™** Protected MOSFET. 28–30
- PWM** Pulse-Width Modulation. 36, 53, 54, 86
- ROS** Robot Operating System. 2, 60, 71–74, 83, 88, 91–93, 96–98

A. Acronyms

RViz ROS visualization. viii, 2, 74, 76, 82, 83, 88, 89, 96

SPI Serial Peripheral Interface. 28, 31, 37, 54, 57, 62, 63, 65, 81

TSC Touch Sensing Controller. 18, 62, 97

UART Universal Asynchronous Receiver-Transmitter. 31, 63, 65, 97

Xacro XML macro. 73

Appendix B

Bibliography

- [1] INC. ROBOTIS. *TurtlBot 3*. URL: <https://www.robotis.us/turtlebot-3/> (visited on 08/12/2023).
- [2] Reichelt elektronik. *TurtlBot 4*. URL: <https://www.reichelt.com/de/en/turtlebot-4-pro-tb4-pro-p330320.html> (visited on 08/12/2023).
- [3] Husarion. *ROSbot 2 PRO*. URL: <https://husarion.com/manuals/rosbot/> (visited on 08/12/2023).
- [4] Husarion. *ROSbot XL*. URL: <https://husarion.com/manuals/rosbot-xl/> (visited on 08/12/2023).
- [5] The Construct Sim. *Mastering with ROS: SUMMIT XL*. URL: https://www.theconstructsim.com/robotigniteacademy_learnros/ros-courses-library/mastering-ros-summit-xl/ (visited on 08/12/2023).
- [6] Clearpath Robotics Inc. *DINGO - Indoor Robotic Platform*. URL: https://www.generationrobots.com/media/DINGO_Datasheet_2020v2.pdf (visited on 08/12/2023).
- [7] TURBOSQUID. *Kuka Youbot 3D model*. URL: <https://www.turbosquid.com/3d-models/kuka-youbot-3d-model-1158728> (visited on 08/12/2023).
- [8] Botsync. *VOLTA*. URL: <https://www.botsync.co/volta> (visited on 08/12/2023).
- [9] Westminster Technologies. *Misti II*. URL: <https://www.westminstertech.com/products/rai-misty-ii> (visited on 08/12/2023).
- [10] Segway Robotics. *RMP Lite 220 - Robotics Mobility Platform Kit*. URL: <https://robotics.segway.com/> (visited on 08/12/2023).
- [11] Ltd Huizhou JD-power Technology Co. *MYH-4625F servo hollow shaft flat outrunner brushless motor*. URL: <http://www.jdpowersky.com/en/p-info.aspx?cid=67&id=93> (visited on 08/13/2023).

I. Personal and study details

Student's name: **Veselý Václav**

Personal ID number: **474718**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Cybernetics and Robotics**

II. Master's thesis details

Master's thesis title in English:

Design and Construction of an Omnidirectional Base for Student Robotics Platform

Master's thesis title in Czech:

Návrh a realizace všesměrového podvozku pro výukovou robotickou platformu

Guidelines:

The main goal of the thesis is creation of a small omniwheel robot usable as a learning platform for university students of robotics. Basic design of the omniwheel platform has already been done in semestral project, however it was not verified in practice. The student is expected to finish the platform, thoroughly test it and improve the original design so that the resulting platform is robust, simple to use, maintain and repair. Both hardware and software design are part of this thesis. The software created for this thesis should follow principles of writing reliable, well documented and simple code. It is expected the robot will have a microcontroller on board that will control the robot's BLDC motors, and a PC or SBC for high-level control. The high-level control framework will be based on ROS (Robot Operating System) and will provide a standardized interface to the robot, including velocity control of the whole mobile base and wheel-based odometry. The developed robot has to offer an easy interface for connecting various sensors like depth cameras, 2D lidars, RGB cameras, ultrasound bumpers etc.

The student should provide an overview of existing robotic platforms suitable for teaching robotics. These platforms should be evaluated in terms of suitability for teaching, maintainability, what modifications of the software and hardware does the platform allow, and similar.

As a verification of suitability of the platform for teaching, the student will design an example lesson for bachelor students that will explain one of the basic concepts of mobile robotics with the help of this platform. Conducting the lesson with real students is optional.

Bibliography / sources:

- [1] <https://robots.ros.org/tags/#education>
- [2] <https://cw.fel.cvut.cz/b212/courses/b3m33aro/tutorials/start>
- [3] Kabátová, Martina & Pekarova, Janka. (2010). Learning how to teach robotics. Constructionism.
- [4] <https://www.theconstructsim.com/teaching-robotics-from-home/>
- [5] <https://www.cs.stanford.edu/groups/manips/teaching/cs223a/>

Name and workplace of master's thesis supervisor:

Mgr. Martin Pecka, Ph.D. Vision for Robotics and Autonomous Systems FEE

Name and workplace of second master's thesis supervisor or consultant:

Date of master's thesis assignment: **13.01.2023**

Deadline for master's thesis submission: **14.08.2023**

Assignment valid until: **22.09.2024**

Mgr. Martin Pecka, Ph.D.
Supervisor's signature

prof. Ing. Tomáš Svoboda, Ph.D.
Head of department's signature

prof. Mgr. Petr Páta, Ph.D.
Dean's signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

Date of assignment receipt

Student's signature