Requirement and implementation specification and test plan

for

Mobile robot electronics project

Prepared by Kalle Filppa (2208121), Dennis Goyal (2207589)

Faculty of Information Technology and Electrical Engineering
Embedded System Project
Computer Engineering
Röning,Juha J.R.
Kauppinen,Marko M.K
Tikanmäki, Antti A.T

Oulu, 20. August 2023

Contents

Revision History 1

1 Introduction 2

1.1 Purpose . . . 2

1.2 Scope . . . 2

1.3 Human resources . . . 3

1.4 Overview . . . 3

2 Requirements 4

2.1 Functional requirements . . . 4

2.1.1 General functionality . . . 4

2.2 Electrical requirements . . . 5

2.2.1 Operating voltage . . . 5

2.2.2 Power source . . . 5

2.2.3 Motor Driver . . . 5

2.3 Sensing requirements . . . 5

2.3.1 LiDAR Sensor . . . 5

2.3.2 Accelerometer . . . 5

2.4 Mechanical requirements . . . 5

2.4.1 Physical dimensions . . . 5

2.4.2 Durability . . . 6

2.4.3 External connectors . . . 6

2.5 Other requirements . . . 6

2.5.1 Price . . . 6

2.6 Future features . . . 6

3 Implementation 7

3.1 Introduction . . . 7

3.2 Component selection . . . 7

3.2.1 Nucleo board STM32F042K6 . . . 7

3.2.2 L293DD Motor Driver Controller 8
3.2.3 LSM303CTR Accelerometer 8
3.2.4 LM3480 Voltage Regulator 8
3.2.5 HM-11 Bluetooth Module 9
3.2.6 VL53L1X LiDAR Sensors 9
3.3 Hardware Design 10
3.3.1 Assumptions 10
3.3.2 Block diagram of the device 10
3.3.3 MCU 10
3.3.4 PCB 11
3.3.5 Power 11
3.3.6 Casing 12
3.4 Board Design on KiCad 13
3.4.1 PCB Design and Schematic Capture 13
3.4.2 Design Rules 15
3.4.3 Bill of Materials (BOM) 16
3.5 Software 18
3.5.1 System Initialization 18
3.5.2 Architecture and Planning: Interface 19
3.5.3 Bluetooth Communication (HM-11) 19
3.5.4 LSM303CTR Accelerometer and Magnetometer 19
3.5.5 Lidar Sensors (VL53X) 19
3.5.6 L293DD Motor Control 19
3.5.7 Interfaces and Middleware 20
3.5.8 Development Environment and Tools 20
3.5.9 Software implementation 20
3.5.10 Main loop 20
3.6 Cost calculations 21

4 Testing **22**
4.1 Basic hardware tests without power and without ICs on board 22
4.2 Basic hardware tests with power and without ICs on board 22
4.3 Tests after connecting IC:s without power 22
4.4 Tests when connecting power to PCB with IC:s 23
4.5 Unit Tests 23
4.6 System tests 24
4.6.1 Tests for general functionality 24
4.6.2 Tests for mechanical requirements 24

5 Reflection and Feedback **25**
5.1 Improvements and Fixes 25
5.2 Project Progression Timeline 27
5.3 Final Feedback 28

Appendices **29**

A Glossary **30**

B Final Schematic and PCB Layout **31**

Revision History

Revision	**Date**	**Author(s)**	**Description**
1.0	16.2.2023	K.Filppa	Initial document creation
1.1	16.2.2023	K.Filppa	Chapter 1 - Introduction
1.2	9.3.2023	D.Goyal	Added project requirements/scope
1.3	20.3.2023	D.Goyal	Robot change from spider to 2-Wheeled
2.1	5.4.2023	K.Filppa	Added Implementation page and selection of components
2.2	24.5.2023	K.Filppa	Hardware design/PCB
2.3	22.6.2023	D.Goyal	Draft board design - Kicad
2.4	1.8.2023	D.Goyal	Added LSM303CTR to board
3.1	15.8.2023	D.Goyal	Completed Software implementation
3.2	20.8.2023	D.Goyal	Added test plan (yet to be performed)
3.3	21.8.2023	K.Filppa	Ran Unit and System tests
3.4	12.10.2023	D.Goyal	Added chapter 5: Reflections and Feedback

Chapter 1

Introduction

1.1 Purpose

The purpose of this document is to specify what this device is about, what is wanted from it, and how it is implemented and tested. The document also defines the requirements and specifications for the robot with DC Motors and a Nucleo control device with a shield to extend capabilities to control the robot. It specifies the requirements of the 2-wheeled robot controlled by an STM32 microcontroller on a Nucleo board with a Bluetooth module, LiDAR, compass, and accelerometer to detect orientation and location. The robot should be able to move around, track its location, detect obstacles, and be remotely controlled via bluetooth.

Requirement specification: defines the requirements for the mobile robot electronics project and preliminary dimensions.

Implementation specification: defines the implementation of the project based on the requirement specification with a corresponding implementation definition for every requirement.

Test plan: defines the tests performed during and after development. Tests verify that the device meets requirements.

The ever-evolving field of robotics has witnessed significant advancements, especially in autonomous mobile robots. This document provides a comprehensive analysis of the two-wheeled robot equipped with the latest STM32 microcontroller, aiming to achieve a blend of seamless navigation. The mobile robot under design is primarily intended for indoor usage. It is a platform to demonstrate basic robotic control, obstacle detection, and environment mapping through embedded design techniques. While the robot will have several features, managing expectations is important. High-level accuracy, extensive mapping, or outdoor capabilities are reserved for more advanced models. This robot focuses on fundamental concepts.

1.2 Scope

Requirement specification covers the functional, physical and electrical requirements of the device. The mobile robot will be built with sensors like an IMU and LiDAR to move in a variety of directions and to avoid obstacles and approximate distance travelled. The robot will be controlled using a Nucleo device with a shield for wheel motor control. A Bluetooth controller may provide an interface for manual input and control.

Implementation specification covers the implementation techniques, protocols, components, PCB and software.

The test plan covers descriptions of basic hardware, unit and system tests.

- Designing and building the physical robot
- Integrating the STM32 microcontroller on a Nucleo board with the necessary sensors and communication

modules

- Programming the microcontroller to control the motors, read sensor data, and communicate with the Bluetooth module
- Developing a mobile application to control the robot via Bluetooth
- Testing and validating the robot's functionality and performance

1.3 Human resources

The project load is distributed evenly between all members. Responsibility is divided into categories. Due to the nature of the project, each member of the team of 2 will fulfil the roles and responsibilities of a Mechatronics Engineer, Project Management and Software Engineer.

1.4 Overview

The robot has a forward-facing distance sensor pair, which it can detect and avoid obstacles and approximate the angle of attack to the obstacle. It will also have a sensor on the side to estimate the distance to the obstacle while following the side of the obstacle. The robot has an IMU which the robot can use to approximate the distance travelled. The robot has a Bluetooth module for communicating with a user who can send instructions to the robot. The robot has a driver, which will be used to control the motor's speed and angular direction. The robot may use these attributes to find a wall, start following it and navigate.

The capabilities and features of the mobile robot will be as follows:

- Move using motor controllers.
- Utilize sensors like a LiDAR and an IMU to avoid obstacles and follow walls.
- User input through a manual shield and respond to it.
- For portability and ease of use, be small and light.
- The mobile robot will meet the following non-functional requirements:

- Speed: the mobile robot should be able to move quickly and efficiently.
- Accuracy: The mobile robot should be accurate when moving and dodging obstacles.
- Reliability: The mobile robot should continue to function reliably over time.
- Safety: Any potential risks should be avoided as the mobile robot is designed. Due to the size of the robot, the risks are minimal.

Chapter 2

Requirements

2.1 Functional requirements

2.1.1 General functionality

Req-2.1.1	The device must be able to move.
Req-2.1.2	The device must estimate the distance and angle of attack to an obstacle.
Req-2.1.3	The device must be able to estimate the distance travelled using the built-in IMU.
Req-2.1.4	The device must have Bluetooth connectivity. Serial Port Profile (SPP) must be supported to detect commands and deliver sensor data. The device must have at least a 10 m range for Bluetooth connectivity.
Req-2.1.5	The device must detect 3-dimensional acceleration (min 0 G, max 2 G, resolution 0.1 G min).
Req-2.1.6	The device must detect the distance and angle to an obstacle straight ahead and the distance to an object on the side.
Req-2.1.7	The device must have status led for power. The led is on when power is connected, and the device is running.
Req-2.1.8	Device must have status led for bluetooth activity. Led is on when data is sent or received via bluetooth connection.
Req-2.1.9	Device must have a power switch.
Req-2.1.10	Device may have internal error log stored to EEPROM. The error log must have space to at least 10 last error. The log must have human readable error description and timestamp.
Req-2.1.11	The device must have a serial port for communicating with the DC motors.
Req-2.1.12	The device must respond with acceleration sensor data when a message requesting it is sent using bluetooth.
Req-2.1.13	The device must respond with orientation sensor data when a message requesting it is sent using bluetooth.
Req-2.1.14	The device must respond with distance travelled when a message requesting it is sent using bluetooth.
Req-2.1.15	The device must respond with motor status data when a message requesting it is sent using bluetooth.
Req-2.1.16	The device must respond with shape of the mapped room when a message requesting it is sent using bluetooth.
Req-2.1.17	The device must be able to move autonomously to map it's environment if instructed.
Req-2.1.18	The robot should have a fail-safe mechanism in case of sensor failures or loss of connectivity.
Req-2.1.19	The robot should be capable of self-diagnostics to identify malfunctioning hardware components.

2.2 Electrical requirements

2.2.1 Operating voltage

Req-3.1.1	The device must have 5V and 3.3V operating voltage available for different parts of the device.

2.2.2 Power source

Req-3.2.1	The device must use 2 x 3.7V 2400mAh Li-ion battery.
Req-3.3.2	Device's operating time with full charge must be at least 1.5 hours.
Req-3.3.3	A power management system to automatically shut down if the battery voltage drops below a certain threshold to prevent damage.
Req-3.3.4	The robot should go into a low-power mode if idle for more than 10 minutes.

2.2.3 Motor Driver

Req-3.2.3	The motor driver must be capable of driving at least two DC motors with a 1:48 or 1:120 gear ratio, and must be compatible with the STM32 microcontroller on the Nucleo board.

2.3 Sensing requirements

2.3.1 LiDAR Sensor

Req-3.2.3	The LiDAR sensor must be powered by the 5V operating voltage and compatible with the STM32 microcontroller on the Nucleo board.

2.3.2 Accelerometer

Req-3.2.3	The 3.3V operating voltage must power the accelerometer sensor and be compatible with the STM32 microcontroller on the Nucleo board.

2.4 Mechanical requirements

2.4.1 Physical dimensions

Req-4.1.1	The device electronics must fit in the car frame.
Req-4.1.2	The sensors must be placed to fulfil the minimum operating distance requirement.
Req-4.1.3	The sensors must be placed to fulfil their purpose as environment detectors properly.
Req-4.1.4	Leds and switches must be placed on the frame's top or sides for easy visibility and access.
Req-4.1.5	The electronics must be placed in the frame so that the components can withstand minor collisions without damage.

2.4.2 Durability

Req-4.3.1	The robot's exterior must be made of a durable material resistant to minor impacts.
Req-4.3.2	The robot's wheels should be designed for optimal traction on various surfaces, including tiles, wood, and low-pile carpet.

2.4.3 External connectors

Req-4.2.1	USB-C connector for charging the battery.

2.5 Other requirements

2.5.1 Price

Req-3.2	Nucleo board STM32 costs approx 20€, the additional electronics costs approx 20€.
Req-3.3	Material and work for the frame and extension boards costs approx 10€.

2.6 Future features

Req-3.2	The device maps the entire room, including the centre of the room
Req-3.3	The device sends data to the user's phone from which a map of walls and obstacles can be rendered to the user interface.

Chapter 3

Implementation

3.1 Introduction

The modern robotics ecosystem thrives on efficient and versatile embedded systems. This document explores designing and implementing an extension board or shields for the Nucleo board STM32F042K6 tailored for a mobile robot. The shield will serve as a hub, incorporating essential modules like Bluetooth communication, motor control, and sensors.

3.2 Component selection

One of the foundational aspects of this project is the judicious selection of components that align with the project's goals and are simultaneously accessible, given the resources at our disposal. Since all components were sourced from the University Lab, there was a dual focus on optimization and accessibility. Most available components were surface-mount devices, so a reflow oven was used to mount components. Here's an insight into the rationale behind the choice of each component:

3.2.1 Nucleo board STM32F042K6

Key Features: A powerful microcontroller board capable of multitasking various robot operations.

Importance: Serves as the robot's brain, interpreting sensor data and making real-time decisions.

Availability: A few different Nucleo STM32 boards were available for use on this project. STM32F042K6 was chosen due to its compact size and extremely lower power consumption. [1]

Specification	Details
Microcontroller	STM32F042K6T6
Architecture	ARM Cortex-M0
Clock Speed	48 MHz
Flash Memory	32 KB
RAM	6 KB
I/O Pins	32
ADC	10-bit, 1 Msps, up to 16 channels
Communications	I2C, SPI, USART
Power Supply	From 3.3V to 5V

[1] `https://www.st.com/en/microcontrollers-microprocessors/stm32f030k6.html`

Rationale: With its robust processing power, versatility, and compatibility with a wide range of peripherals, the STM32F042K6 is an ideal choice for the robot's microcontroller needs. A drawback we later discovered is its limited IO capacity, which was overcome using software techniques. The STM32F042K6 is a choice that aligns perfectly with the requirements of embedded robotic projects. It has a compact form factor that integrates well within constrained spaces, and its ARM Cortex-M0 architecture offers a powerful yet energy-efficient processing solution. The support for multiple communication interfaces like I2C, SPI, and USART facilitates seamless interfacing with sensors and modules.

3.2.2 L293DD Motor Driver Controller

Working Principle: Converts low-power control signals into higher-power signals to drive the motors.

Importance: Essential to drive the two DC motors of the robot, making motion possible.

Availability: A staple in many robotic projects in the lab, this component was easily accessible.

Specification	**Details**
Type	Quadruple Half-H Driver
Voltage Range	4.5V to 36V
Output Current	600mA per channel (1.2A peak)
Channels	Can control 2 DC motors or one stepper motor
Protection	Thermal Shutdown and High-Noise Immunity Inputs
Interface	TTL Compatible

Rationale: A bridge between the logic and power circuits is required to transform the microcontroller's instructions into actuation. The L293DD, capable of driving two DC motors simultaneously, fits well.

3.2.3 LSM303CTR Accelerometer

Overview: A module measuring acceleration enables the robot to understand it's path/distance travelled.

Use-case: Facilitates functionalities like auto-correction when the robot tilts or is off-course. LSM303CTR 6-axis IMU with accelerometer and magnetometer will be used to estimate the distance travelled and the distance vectors' orientation regarding the earth's magnetic field. The sensor also operates at the specified 3.3V operating voltage. It will communicate with the MCU using the SPI communication interface.

Availability: Previously used in motion-related projects, this component was available in ample quantity.

Specification	**Details**
Type	3D digital linear acceleration sensor and a 3D digital magnetic sensor
Acceleration Range	$\pm 2/\pm 4/\pm 8$ g
Magnetic Field Range	± 16 gauss
Output Data Rates (ODR)	Accelerometer - up to 1kHz; Magnetometer - up to 100Hz
Interface	I2C, SPI

Rationale: The robot is assumed to move only in a 2D plane with a gravity component perpendicular to it, so a gyroscope isn't required. The LSM303CTR, with its precision and reliability, aids in achieving this objective.

3.2.4 LM3480 Voltage Regulator

Need: A voltage regulator ensures consistent 5V output with various components requiring stable power. The regulated 5V from Nucleo can't be used as it has lower current rating than required for DC motors.

Choosing the 5V regulator: Chosen for its reliability, stability, and compatibility with onboard components.

Specification	Details
Output Voltage	5V
Input Voltage Range	Up to 35V
Output Current	Typically up to 100mA
Dropout Voltage	Low, typically less than 1V
Protection	Overheat and Overcurrent Protection

Rationale: Power management in robots is crucial to ensure consistent performance. The LM3480 is adept at providing a stable 5V output, even when input voltages fluctuate. Its low dropout voltage means the robot can function effectively even as battery levels deplete. The in-built overheat and overcurrent protection mechanisms further safeguard the system, ensuring longevity and reliable performance.

3.2.5 HM-11 Bluetooth Module

Overview: A low-energy module aiding wireless communication between the robot and external devices.

Benefits: Offers a seamless link to mobile devices for robot control and status updates. Provides Bluetooth V4.0 HM-11 BLE. It operates at the specified 3.3V, is cheap and supports BLE, which is required for the power consumption requirements. It will communicate with the MCU using UART.

Availability: The lab had a stock of these modules due to their popularity in wireless communication projects.

Specification	Details
Bluetooth Version	Bluetooth 4.0 (Bluetooth Low Energy)
Frequency Range	2.4 GHz
Transmission Range	Up to 100 meters (in open space)
UART Interface	With adjustable baud rate
Power Consumption	Ultra-low sleep current of 400uA

Rationale: The module offers a stable and efficient Bluetooth 4.0 connection, allowing seamless communication with other devices, making it pivotal for remote control operations. In robotic applications where wireless communication is pivotal, the HM-11 module stands out. Bluetooth 4.0 BLE ensures rapid data transmission rates while preserving energy. The module's capability to function at a low sleep current of 400uA guarantees longer operational times, which is vital for autonomous robots. Its UART interface also simplifies communication with the main controller.

3.2.6 VL53L1X LiDAR Sensors

Purpose: Provides distance measurement aiding obstacle detection and navigation.

Specification: High-resolution sensors with a range optimal for indoor robotics navigation. VL53L1X - is a long-range, high-speed Time-of-Flight (ToF) laser-ranging sensor designed for various applications, including robotics, drones, industrial automation, and IoT devices.

Some key features of the VL53L1X sensor include:

- Measuring range up to 4 meters with millimetre-level accuracy
- Fast measurement rate of up to 50 Hz
- Low power consumption in both active and standby modes
- I2C interface for easy integration with microcontrollers and other devices
- Compact package size of 4.9 x 2.5 x 1.56 mm

Availability: These sensors were present due to their inclusion in previous automation and robotics projects. Can also receive help with designing the LiDAR sub-boards to be connected using jumper cables from the Fablab team.

Specification	**Details**
Distance Measuring Range	Up to 4 meters
Accuracy	±25 mm
Measurement Mode	Single, Continuous
Output Interface	I2C
Peak Emission Wavelength	940 nm
Field of View	15° typical
Supply Voltage	2.6V to 3.5V
Power Consumption	Active ranging consumption reduced to 19 mW (typical)

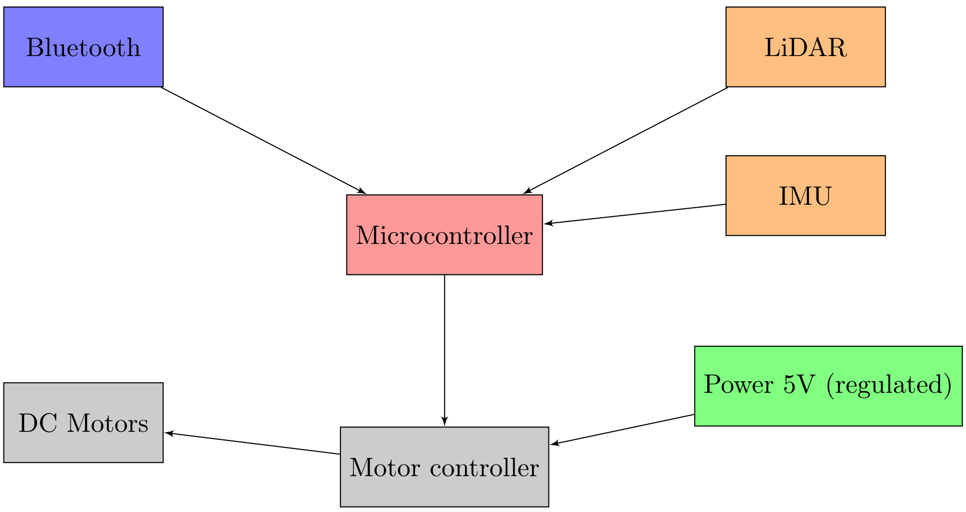
Rationale: The VL53L1X LiDAR sensor offers exceptional accuracy in distance measurement, essential for collision avoidance and navigation in robots. The capability of detecting objects up to 4 meters away and with an accuracy of ±25 mm ensures that the robot can react promptly to obstacles. Its compact size and the ability to deliver high-speed measurements make it ideal for real-time navigation.

3.3 Hardware Design

3.3.1 Assumptions

In this project, a NUCLEO-F042K6 development board and a breadboard are used to test the PCB design while it is in process. After a successful PCB design is verified, a shield is manufactured for the NUCLEO-F042K6 board.

3.3.2 Block diagram of the device



3.3.3 MCU

The STM32F042K6 is a microcontroller (MCU) that's part of STMicroelectronics' STM32 family, which is based on the high-performance ARM® Cortex®-M0 32-bit RISC core. This MCU is optimized for cost-sensitive and space-constrained applications, offering a balanced blend of computing power, energy efficiency, and peripheral integration. With up to 32 KB of Flash memory, 6 KB of RAM, and a rich set of peripherals like UART, SPI, I2C, and more

3.3.4 PCB

The entire robot contains three additional PCBs in addition to the NUCLEO development kit. One main extension board and one for each of the two LiDAR sensors. The Nucleo board has to fit into the main extension on the board.

3.3.5 Power

2pcs of 18650 3.7V Li-Ion battery cells are used to provide power. This is a requirement for the 5V Voltage regulator to operate where $V_s > V_{ss}$.

The 3.3 V requirement is fulfilled by using the regulated pin from the Nucleo board.

Power Consumption Estimation

Component Specifications

The different components consume power as follows:

HM-11 Bluetooth:

- Peak current during transmission: $\approx 20\,\text{mA}$
- Operating voltage: 3.3 V

LSM303CTR Accelerometer and Magnetometer:

- Operating current (combined): $\approx$ 500 ¯A (0.5 mA)
- Operating voltage: 3.3 V

L293DD Motor Driver:

- Quiescent current (no motor load): $\approx 5\,\text{mA}$
- Operating voltage: up to 36 V (but we'll consider 7.4 V as it's powered by 2 Li-ion batteries)
- **DC motor:**
  - Stall current: $\approx 800\,\text{mA}$

2 Lidar sensors VL53X:

- Operating current (each): $\approx 20\,\text{mA}$
- Operating voltage: 3.3 V

LM3480-5.0V regulator:

- Quiescent current: $\approx 100\,^{-}\mathrm{A}$ ($0.1\,\mathrm{mA}$)
- Operating voltage: Assuming input is $7.4\,\mathrm{V}$ and output is $5\,\mathrm{V}$

Note: Passive components like capacitors, diodes, and resistors will not consume power unless there's a current passing through them. For the sake of this estimation, we'll exclude their power consumption as it would be marginal.

Power Calculation

Power using the formula $P = IV$.

$$
\begin{aligned}
\text{HM-11 Bluetooth Power} &= 3.3\,\mathrm{V} \times 20\,\mathrm{mA} = 66\,\mathrm{mW} \\
\text{LSM303CTR Power} &= 3.3\,\mathrm{V} \times 0.5\,\mathrm{mA} = 1.65\,\mathrm{mW} \\
\text{L293DD Quiescent Power} &= 7.4\,\mathrm{V} \times 5\,\mathrm{mA} = 37\,\mathrm{mW} \\
\text{DC Motor Power (each)} &= 7.4\,\mathrm{V} \times 800\,\mathrm{mA} = 5920\,\mathrm{mW} \\
\text{2 Lidar sensors Power} &= 2 \times (3.3\,\mathrm{V} \times 20\,\mathrm{mA}) = 132\,\mathrm{mW} \\
\text{LM3480-5.0V regulator Power} &= (7.4\,\mathrm{V} - 5\,\mathrm{V}) \times 0.1\,\mathrm{mA} = 0.24\,\mathrm{mW}
\end{aligned}
$$

Component	**Power Consumption (mW)**
HM-11 Bluetooth	66
LSM303CTR Accelerometer and Magnetometer	1.65
L293DD (Quiescent)	37
DC Motor (each)	5920
2 Lidar sensors VL53X	132
LM3480-5.0V regulator	0.24
Total	**12076.89**

Table 3.1: Power consumption estimation for each component

Total current consumption:

$$I_{\text{total}} = 20 + 0.5 + 5 + 1600 + 40 + 0.1 = 1665.6\,\mathrm{mA}$$

Estimated Runtime on Full Charge:

Assuming a battery capacity of $2400 * 0.85\,\mathrm{mAh}$:

$$\text{Runtime} = \frac{\text{Battery Capacity}}{I_{\text{total}}}$$

$$\text{Runtime} = \frac{2000\,\mathrm{mAh}}{1665.6\,\mathrm{mA}} \approx 1.2 \text{ hours}$$

3.3.6 Casing

The development board, extension board and LiDAR chips will be placed on the mobile robot frame. They will be placed so that if the robot collides with an object, it will take the impact with the frame instead of the electronics.

3.4 Board Design on KiCad

KiCad is a premier open-source software suite for electronic design automation (EDA). For this project, we tried to leverage existing libraries for most components and validated them against datasheets. For e.g. we used the footprints provided by Fablab or created from the datasheets of respective components.[2] Custom footprints were cross-referenced with component physical dimensions.

- Prioritized component placement for minimized trace lengths for critical signal paths.
- Ensured decoupling capacitors were placed close to active components.
- Utilized a ground plane for better noise immunity and thermal management.

Component Placement

Each component was strategically placed to minimize trace lengths, ensuring optimal performance and reducing possible interference. See final design.

Routing

Priority was given to power and ground connections, followed by high-frequency signals, particularly from the LiDAR and Bluetooth module, ensuring minimal noise and cross-talk.

3.4.1 PCB Design and Schematic Capture

Pin connections on the PCB

To ensure proper functionality and communication between integrated components, the following pin connections were made on the PCB board:

Connection to Nucleo:

Power Management:

Pin	Connection
VIN	Output of the 5V Voltage regulator (VCC)
3v3	3.3v output from Nucleo
GND	digital ground

* 3.3v can be safely used from the Nucleo device due to low current consumption by ICs. According to the datasheet a max of 500mA » $1\tilde{00}mA$.

LSM303CTR Accelerometer & Magnetometer:

Using the SPI protocol for communication:

We realised later that 2 pins were connected to the wrong output ports of Nucleo. **NRST** and **5V** outputs from Nucleo were connected to **INT_XL** and **DRDY** of Accelerometer. As these are optional we disconnected by scraping off the copper connection on the PC board.

[2] `https://gitlab.fabcloud.org/pub/libraries/`

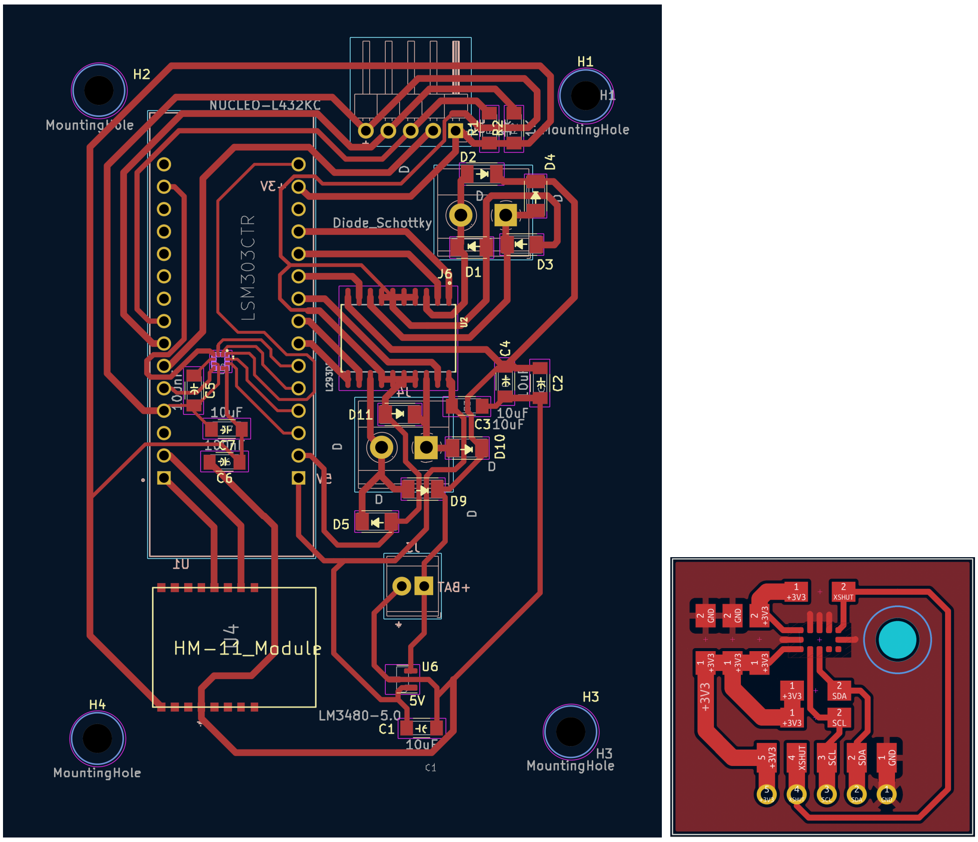


Figure 3.1: Completed PCB layout (copper fill removed from image)

Nucleo Pin	LSM303CTR Pin
PB_3 (SPI1_CLK)	SCL/SPC
PA_7	CS_MAG
PA_2	CS_XL
PA_12	INT_MAG
PB_5 (SPI1_MOSI)	SDO/SDA/SDI

L293DD Motor Driver:

Connections for the L293DD Motor Driver:

Nucleo Pin	L293DD Pin
PA_0	EN_A
PA_1	Input_A_1
PA_3	Input A_2
PA_4	Input B_1
PA_5	Input B_2
PA_6	EN_B

Motor outputs from L293DD are connected through 100V Schottky diodes for protection against kickback. The power is provided through 5V Voltage regulator directly from the power source marked as V_{ss}.

HM-11 Bluetooth Module:

Using the UART protocol for communication:

Nucleo Pin	HM-11 Pin
PA_9	UART_TX
PA_10	UART_RX

VL53L1X LiDAR:

Connections to the optional LiDAR sensors are made through jumper cables:

Nucleo Pin	LiDAR Pin
PB_6 (I2C1_SCL)	SCL
PB_7 (I2C1_SDA)	SDA
PB_0	XSHUT

Additional Note: Capacitors and resistors are placed throughout the board to ensure voltage stability and control current flow, respectively.

3.4.2 Design Rules

Conducted ERC and DRC checks in KiCad. Minor errors were rectified related to ground connections.

Recommended Design Rules

Parameter	Specification (Minimum)	Final Design
Hole (drill) diameter	0.3mm	1mm
Hole pad	1.5 times hole diameter	1.53mm
Line width	0.2mm (200µm, 8mils)	0.3mm
Clearance	0.2mm	0.2mm

Note: Absolute minimum line width (to connectors or component footprints), line to line, and clearance between pads is 0.1mm (100µm, 4mils).

PCB Specifications

Parameter	Specification	Final Design
PCB size	250mm x 180mm (max)	82mm x 66mm
Standard PCB material	FR4	FR4
Thickness options	0.8mm insulation/18µm copper both sides, 1.0mm/18µm, 1.6mm/35µm	–

Id	Designator	Footprint	Quantity	Designation
1	Diodes D1-11	SOD-123T	8	Diode_Schottky
2	U4	HM-11_Module	1	HM-11_Module
3	C6,C5	C_1206	2	100nF
4	U6	SOT-23	1	LM3480-5.0
5	R2	R_1206	1	1K
6	U2	SOIC127P1032X265-20N	1	L293DD
7	C1,C4,C7,C3,C2	C_1206	5	10uF
8	H3,H1,H2,H4	MountingHole_3.2mm_M3_ISO7380	4	MountingHole
9	R1	R_1206	1	1k
10	U5	LGA12R50P_200X200X100	1	LSM303CTR
11	J4,J6	MKDS-1,5-2-5.08_1x02_P5.08mm	2	MOTOR2 OUT
12	U1	STM32F042K6	1	NUCLEO-L432KC
13	J3	282834-2_1x02_P2.54mm	1	Conn_01x02_Socket
14	J5	PinHeader_1x05_P2.54mm_Horizontal	1	LiDAR_conn

Table 3.2: Detailed Bill of Materials for the Nucleo Extension Board Design

3.4.3 Bill of Materials (BOM)

PCB Assembly

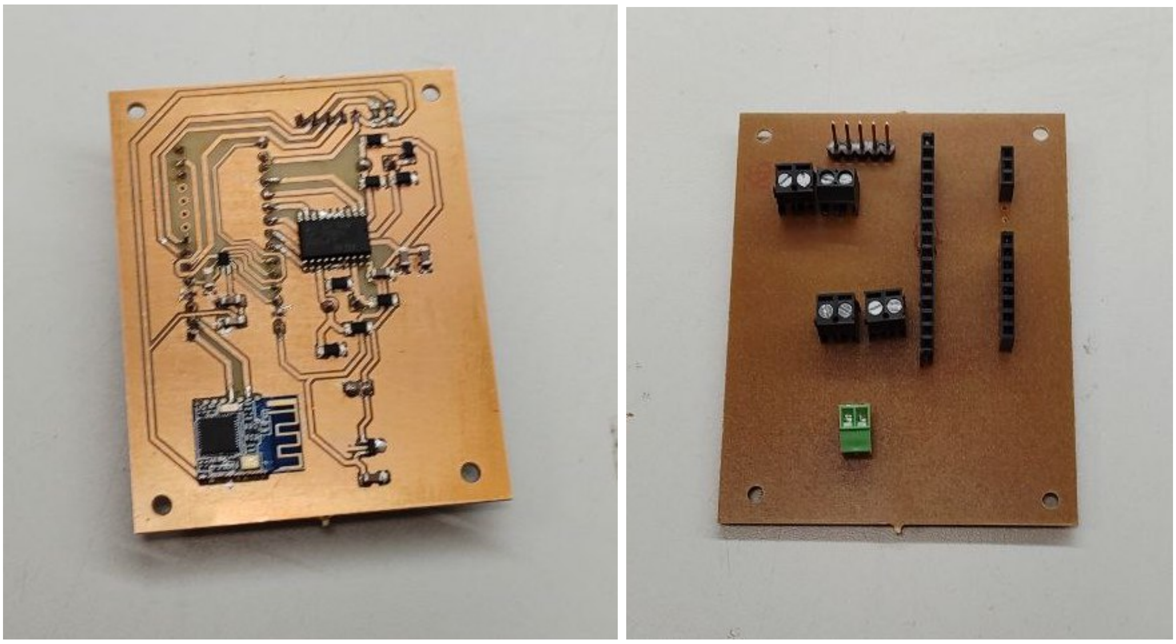


Figure 3.2: Assembled extension board

PCB milling was done in the Fablab facility at the University of Oulu. The process was straightforward and automatic using LPKF Laser and Electronics -machine. The quality of drill trace was good enough for prototyping although few of used drill bits were worn out and needed replacement. Quality of milling was confirmed with microscope inspection and with multimeter. No short circuits were detected.

The surface of the PCB was sanded with steel wool to remove clear couting. It was also washed with isoprophyl alcohol to remove dust, dirt and grease. Board was prepared to oven by adding low temperature mounting paste to board. Paste was added with sharp needle to specific spots of the board. For LSM303CTR Accelerometer and Magnetometer paste adding and component positioning had to be done under microscope. It revealed to be a hard task because LSM303CTR is very tiny component and legs of the component are positioned underneath the component so those can not be seen from the top view.

First ovening went well altought there were few bad spots on the board, like Accelerometer and Bluetooth. Those two components were removed and new layer of paste was added to Accelerometer pins on board. The component was attached again with another ovening process. Bluetooth chip was added this time with hand soldering. Also few capacitors were taken off and put back to board using hand soldering. The quality of hand soldering and oven soldering was confirmed with microscope insection and also with multimeter measurements using resistance measurement. The oven used for manufacturing is LPKF Laser and Electronics ProtoFlow.

After adding all the surface mounting components we added all of the connectors. Rails for Nucleo and input connector for operating power was added, and also output connectors for motor controlling and for lidar.

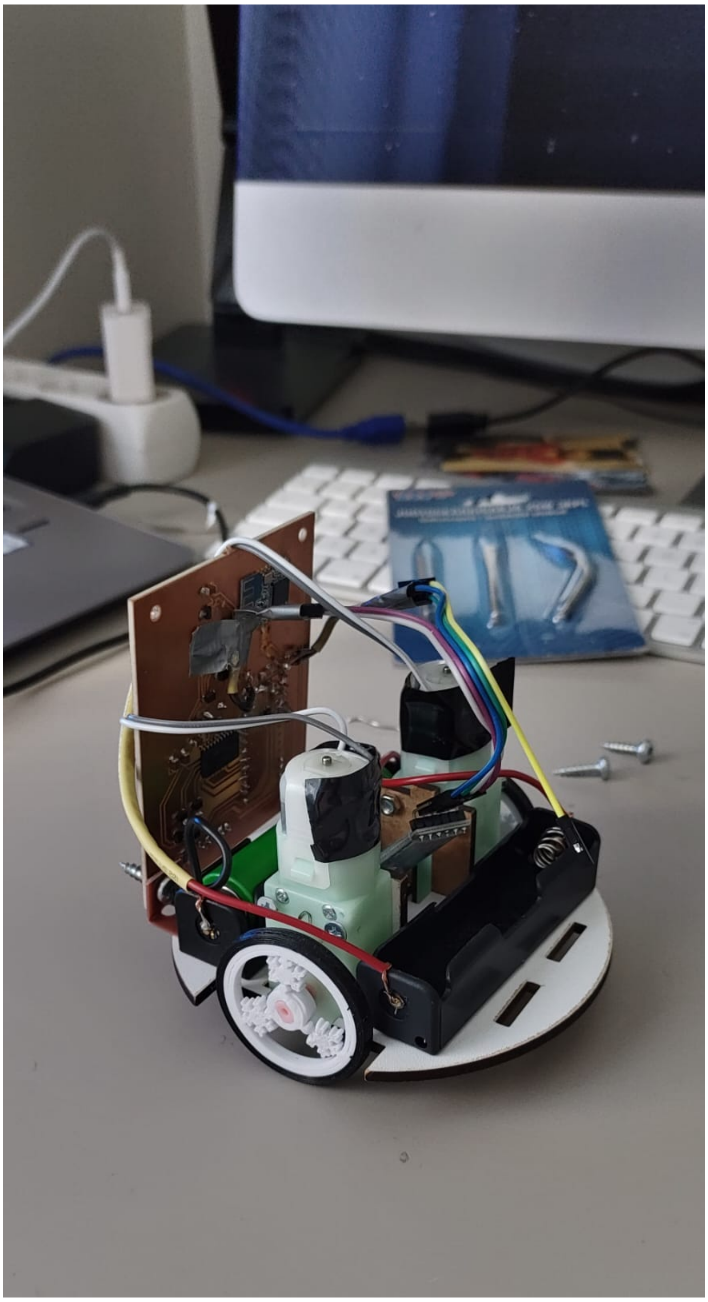


Figure 3.3: Robot/Final Build

3.5 Software

The objective is to collect data from the LSM303CTR accelerometer, and the Lidar sensors, and then process this data to drive two DC motors through the L293DD. Additionally, we'll establish a Bluetooth connection using the HM-11 to relay data or commands.

3.5.1 System Initialization

Ensure the proper setup and initialization of:

- Processor clock (48 MHz)
- SPI and I2C interfaces (for LSM303CTR and VL53L1X)
- GPIOs (for L293DD motor control, etc.)
- UART (for HM-11 Bluetooth communication)

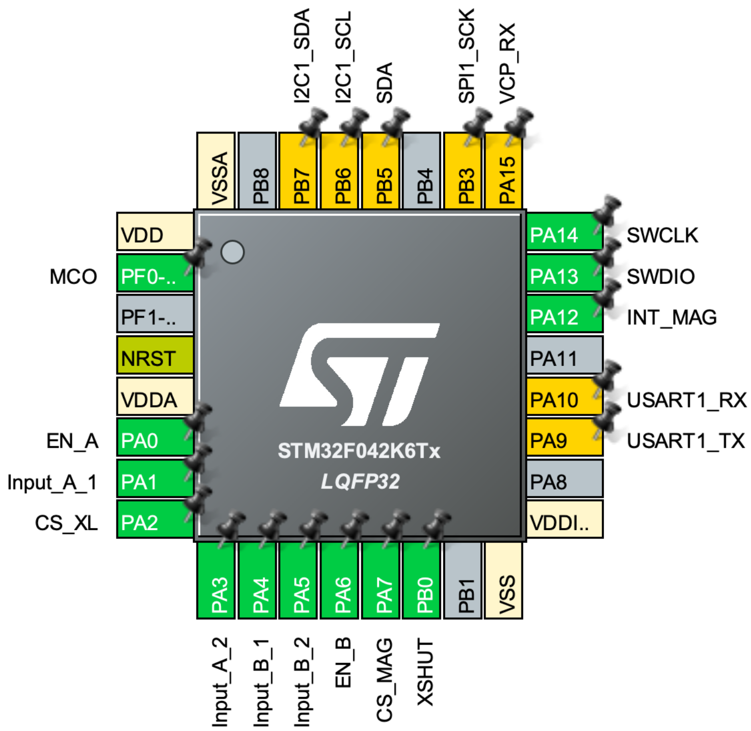
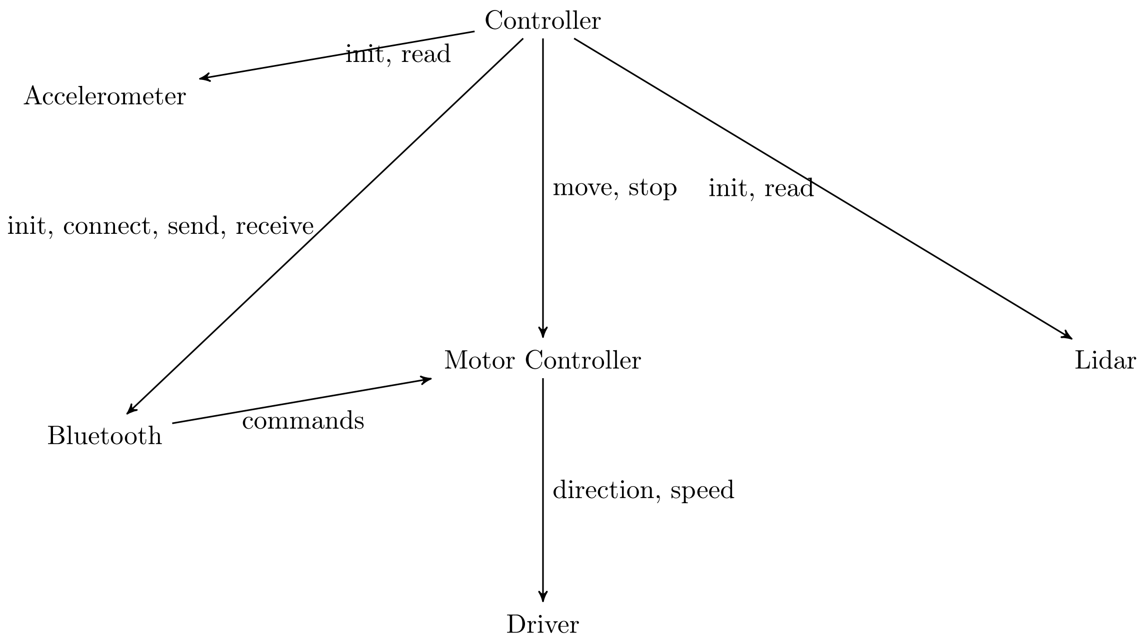


Figure 3.4: Completed PIN configuration

3.5.2 Architecture and Planning: Interface



3.5.3 Bluetooth Communication (HM-11)

- Setup UART communication for HM-11.
- Define command structure (e.g., start/stop motors, request sensor readings).
- Handle data reception: Parse received commands and take appropriate action.
- Handle data transmission: Send requested sensor data or system status.

3.5.4 LSM303CTR Accelerometer and Magnetometer

- Initialize SPI for LSM303CTR.
- Define a routine to periodically read data or upon request.

3.5.5 Lidar Sensors (VL53X)

- Initialize I2C interface, considering shared I2C, ensure unique addresses.
- Periodically poll the sensors.
- Process the readings for relevant information, e.g., obstacle distance.
- Depending on distance, make motor decisions: slow down, stop, or redirect.

3.5.6 L293DD Motor Control

- Setup necessary GPIOs.
- Define functions for motor actions: start, stop, turn left/right, speed control.

3.5.7 Interfaces and Middleware

- Develop an API that abstracts the hardware details. For example:
  - `setMotorDirectionA()`, `Motor_Stop()`, `getMotorStatus()`.
  - `Bluetooth_Connect()`, `Bluetooth_ReadMessage()`.
  - `LSM303CTR_Init(add)`, `LSM303CTR_ReadAccel(add)`, `LSM303CTR_ReadReg(reg)`.
  - `Lidar_ReadDistance(sensor_id)`.
  - `moveForward(speed), moveBackwardLeft(speed)`.

3.5.8 Development Environment and Tools

The software was developed in the STM32CubeIDE environment, an all-in-one multi-OS development tool offered by STMicroelectronics. This environment offers a comprehensive software toolchain, including a C/C++ compiler, debugger, and simulator. STM32CubeMX was used for the initial configuration of peripherals and clock settings. The libraries used as an interface for different protocols were uart, soc and i2c from **stm32f0xx_hal** and **stm32f0xx_hal_***.

3.5.9 Software implementation

The initialization phase involves setting up the system clocks, configuring the GPIO pins, and initializing any peripherals required for the robot's operation. This includes initializing the PWM outputs used to control the DC motors, configuring the UART port for Bluetooth communication, I2C for Lidar sensors and SPI for Accelerometer.[3]

3.5.10 Main loop

Main Loop: The main loop is the software's core, responsible for controlling the robot's movement and handling sensor data. The main loop consists of the following sub-modules: '

1. Accelerometer (LSM303CTR) Implementation - accel.c:

   The software interfaces with the LSM303CTR accelerometer and magnetometer module to gather motion data. Functions are provided to initialize the accelerometer, read acceleration values, and process the data for further use.

2. Bluetooth (HM-11) Implementation - bluetooth.c:

   The HM-11 Bluetooth module is utilized for wireless communication. Functions are available for initializing the Bluetooth module, sending and receiving data, and handling Bluetooth events.

3. Motor Controller (L293DD) Implementation - motor_controller.c:

   The L293DD chip is used to control the two DC motors of the robot. The software provides functions to control the direction, speed, and operation of the motors. Motor initialization, start, stop, and direction control functions are implemented to give precise control over the robot's movement.

4. Lidar Sensors (VL53X) Implementation - lidar.c:

   The two external VL53X Lidar sensors, connected via jumper cables and sharing the I2C interface, are crucial for distance measurement and obstacle detection. The software includes functions to initialize the Lidar sensors, read distance values, and process the data for navigation purposes.

5. Main Driver - main.c & driver.c:

   The main.c file serves as the controller of the software, orchestrating the operation of all modules and ensuring synchronized functioning. The driver.c file contains utility functions and drivers that aid in the overall operation of the robot, ensuring smooth integration of all components.

[3] Git repository: `https://github.com/davy320/Embedded_robot/tree/main/Core`

3.6 Cost calculations

Most of the components, ICs and development were done in Fablab so the cost for this project has been minimal. Below estimates assume that none of the facilities were available and the project is done independently from Fablab's assistance.

Component/Item	**Estimated Cost (in USD)**
HM-11	$2
LSM303CTR	$1.5
VL53L1X	$3
Nucleo-F042K6T6	$11
L293DD	$2
2 x DC Motors	$5
2 x 3.7v Li-ion	$15
Passive components (Diodes, resistors, etc.)	$10
Total cost	**$50**

Table 3.3: Cost breakdown for the project components.

Chapter 4

Testing

4.1 Basic hardware tests without power and without ICs on board

Test ID	Description	Result
Test-1.1.1	Before connecting IC:s and power	✓
Test-1.1.2	Visual inspection shows no shortcuts or bad solder connections.	✓
Test-1.1.3	GND, V_{CC} and $3v3$ lines are not shortcutted.	✓
Test-1.1.4	GND is in contact with the DC connectors to the GND pins of Diode sockets.	✓
Test-1.1.5	V_{CC} line is in contact from the DC connectors to the V_{CC} pins of Diode sockets.	✓
Test-1.1.6	Data pins of the HM-11 Bluetooth module are in contact to corresponding pins in MCU	✓
Test-1.1.7	Data pins of the HM-11 Bluetooth module are not in contact to GND or V_{CC}	✓
Test-1.1.8	Data pins of the LSM303CTR sensor module are in contact with the corresponding pins in MCU	✓
Test-1.1.9	Data pins of the LSM303CTR sensor module are not in contact to GND or V_{CC}	✓
Test-1.1.10	Jumper Data pins of the Lidar module are in contact with the corresponding pins in MCU	✓
Test-1.1.11	Jumper Data pins of the Lidar module are not in contact to GND or V_{CC}	✓

4.2 Basic hardware tests with power and without ICs on board

Test ID	**Description**	**Result**
Test-1.2.1	When powering on the device the current consumption is under 50 mA	✓
Test-1.2.2	Voltage between V_{CC} and GND is 5 volts within a tolerance of 0.2 V	**5.46V**
Test-1.2.3	Any component does not heat up rapidly	✓

4.3 Tests after connecting IC:s without power

Test ID	**Description**	**Result**
Test-1.3.1	GND and V_{CC} are not in short circuit	✓

Tests 4.1, **4.2** and **4.3** suggest, there are no issues with the power management and connections. Additionally, the voltage regulator is working as expected. 0.5V tolerance is acceptable by the Nucleo device and DC Motors.

4.4 Tests when connecting power to PCB with IC:s

Test ID	Description	Result
Test-1.4.1	Any component does not heat up rapidly.	-*
Test-1.4.2	The current consumption is under 100 mA.	✓

* Plugging in the Nucleo to the extension board is causing the board to heat up rapidly. Unable to figure out the issue (connecting Nucleo through jumper cables to Motor driver and Bluetooth doesn't cause the same issue)

4.5 Unit Tests

Test	Description
Test-2.1	**MCU is operational**
Tools	Device with microcontroller attached.
Steps	Upload code to Nucleo device.
Accepted result	LED1 is lit and ST/LINK debugger intiatilises
Test result	Error: failed to run debugger and link Nucleo
Test-2.2	**Bluetooth module is operational**
Tools	Device with microcontroller and bluetooth module attached.
Steps	Establish bluetooth connection to the device. Send "AT+CON" command to the device.
Accepted result	Device must send response: "OK+CONN"
Test result	"OK+CONN" is received
Test-2.3	**Acceleration sensor module is operational**
Tools	Device with microcontroller and acceleration sensor module attached.
Steps	Enable acceleration sensor test from source code. Flash new software to the device. Hold device in normal position. Switch device power on. Turn device upside down. After test flash the original software back to the device.
Accepted result	Print Accelerometer data in debugger
Test result	Unable to get device working due to overheating and flash upload issues when Nucleo is embedded
Test-2.4	**Mode switch/Bluetooth controls**
Tools	Device with microcontroller, Bluetooth and motor driver.
Steps	Flash new software to the device. Switch device power on. Connect using bluetooth and control device motors
Accepted result	Able to control device motors and switch operational mode between Autonomous/Bluetooth.
Test result	Accepted
Test-2.5	**Reset button is operational**
Tools	Device with microcontroller.
Steps	Press the reset button.
Accepted result	Device is reset to initial conditions of the software, mode (autonomous)
Test result	Device get's reset to initial conditions

Table 4.1: Unit Test Plan

4.6 System tests

4.6.1 Tests for general functionality

Overall, Nucleo device is over-heating and hence unable to perform standard operations. Connecting through jumper cables, able to control motor drivers and control using Bluetooth module. Overheating is most likely coming from wrong connection of two of the IMU's pins to Nucleo. When those two connections are been scraped off with a sharp knife we can connect the Nucleo to PCB without jump cables. Wrong connections can be fixed with cross soldering those pins with jump cables. After this fix the Nucleo is working properly.

There is also significant rise of temperature on the corner of the voltage regulator of our PCB. Although this is an alarming feature, after testing and consideration we conclude that this is normal component behaviour and the voltage regulator is working correctly. When we test the robot with our code we notice that the voltage regulator occasionally drops the voltage when changing motor speeds. This is most likely because the voltage regulator can not handle too rapid voltage changes.

4.6.2 Tests for mechanical requirements

Robot can perform its task of moving/controlling Motor #2 using PWM. Due to incorrect connection of PA_0 port to EN_A. Motor #1 speed can't be controlled using PWM as it is not supported by the Nucleo. This is fixed by using both connections with our own PWM code.

With this code we can control both motors separetly. We are controlling the 5V digital output of Nucleo so that we get decired voltage from 0 to 5V by changing the duty cycle from 0 to 100 percent, how much time the pin is on state high. This method needs sharp timing and altough the code is working fine when testing only the motors it gets more complicated after we add the code for bluetooth too. We should have been using timers to build our own PWM code but we used basic delays. These are the reasons why we are not using our PWM code for final version and instead we use only full speed for motors.

Robot frame is modified with special plastic brackets where we can mount our PCB. The PCB is positioned upright to the frame so that connections for motors and other external equipments can be easily reached with cables. Upright position is the easiest way to mount our relatively big PCB to the small and tight robot frame.

Chapter 5

Reflection and Feedback

5.1 Improvements and Fixes

Continuous improvements to components, software and the board have helped resolve issues. The overheating issue was resolved through a meticulous process of disassembling and systematically reassembling the components on the board. This allowed for a thorough inspection of each part for potential defects or misalignments and also ensured that all components were securely and correctly seated, enhancing the overall thermal efficiency. Flashing the program to the microcontroller when the component is boarded on top of Nucleo was resolved by removing the pin connections *NRST* and *5V* outputs (Nucleo) *INT_XL* and *DRDY* (Accelerometer) 3.4.1.

The issue with Bluetooth module HM-11 receiving data from the master device. Initially, we configured connected incorrect pins to the Bluetooth chip from Nucleo, connecting *Rx* to *Rx* and *Tx* to *Tx*, which was fixed by connecting them alternatively using jumper cables 3.4.1. We had to replace the Bluetooth chip with JDY-31 **HC-05/06**, though it doesn't support Bluetooth LE, it was easy to configure and connect.

There were a few deviations from the initial plan, especially from *Phase 1* of the project 5.2. The initial scope mentioned the development of a dedicated Bluetooth application to receive sensor data and control the robot. It was also discussed to map the environment using data from the 2 Lidars and IMU was not considered part of the extension module. What was covered to an extent was the speed, accuracy, low power consumption and reliability of the robot. A couple of more months of part-time work may be required to meet the requirements for sensor data which will include redesigning the board to embed a new IMU and dynamical reading of sensor data through software as there are no dedicated pins for each sensor.

Regardless of the unit tests and continuous improvements, many functionalities and features didn't work as planned. Due to the IMU being too small (2.0x2.0x1.0 mm), the PIN connections had improper connections and still couldn't be embedded by hand soldering or reflow and hence we weren't able to get the acceleration data required to predict movements. The IMU soldering was done twice in the lab by adding soldering phaste under microscope and using the soldering oven. Despite this, we could not verify the quality of the solders for IMU other than by measuring with a multimeter because we could not see the result underneath the small component. By measuring the PCB lines for IMU with multimeter we found out that there was no connections between the separated lines but we still could not verify if the lines have cood connection to the pins of IMU. And so as mentioned, in the first tests with Nucleo we noticed the mapping error for two of the pins from IMU to Nucleo. All this could be improved by choosing an IMU of a bigger size or getting the print done with more professional equipment or from a dedicated lab. Though the libraries for Lidar and IMU were written and compiled for the Nucleo using SPI and I2C protocol respectively, we require more development time to test and make them work with the Nucleo.

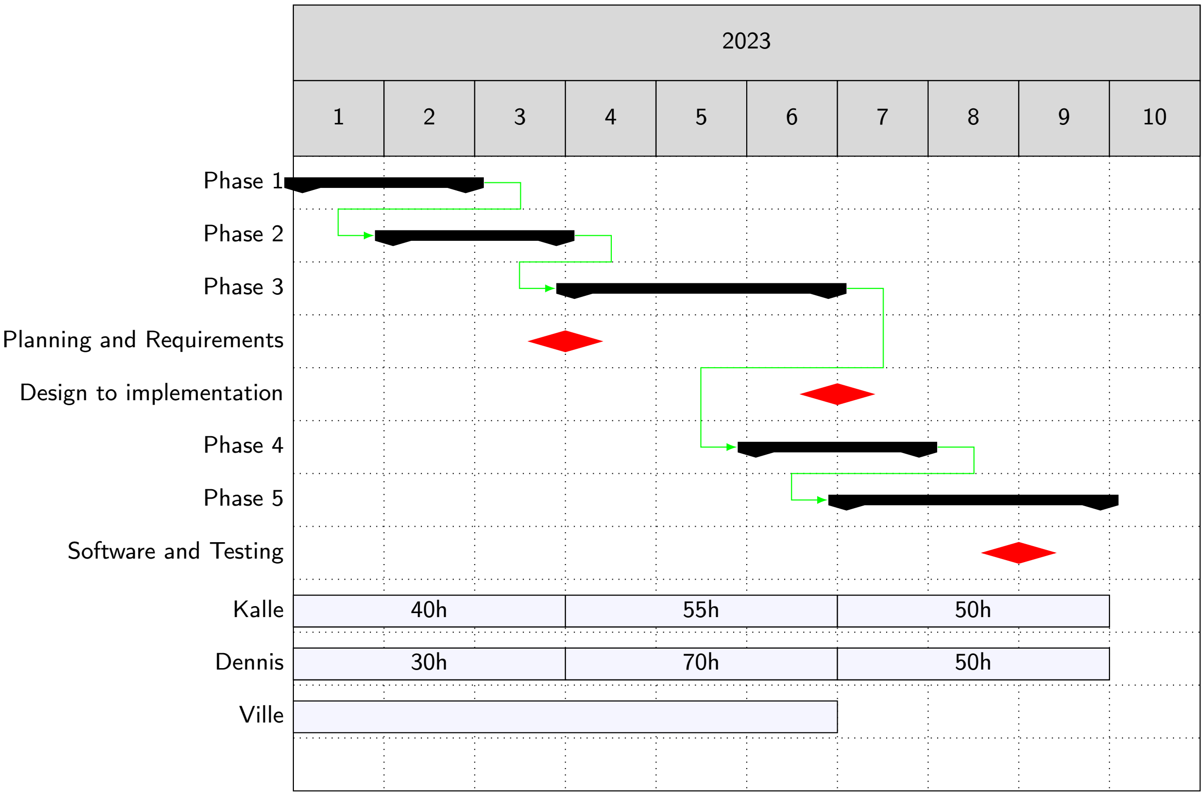
In the final tests we noticed that when giving the motor speed commands too rapidly to the device the voltage regulator could not keep up with the commands and dropped the voltage occasionally which leads to short but visible outage of the Nucleo board. This could be avoided by using the voltage for motors directly from batteries and not through a voltage regulator. When sending the commands in slower pace the device corresponded fine. When the Nucleo starts again because of the voltage drop the initialization of the bluetooth happens again and also the delays in the code needed for the bluetooth initilization is repeated which most likely are also reasons of visible idle of the device when sending commands too rapidly.

Table 5.1: Individual contribution on different areas

	Requirements and Planning	Design and Schematics	Manufacturing and Fabrication	Software and Testing	Meetings and workshops (total)
Kalle	40h	20h	35h	50h	≈50h
Dennis	30h	70h	10h	50h	

5.2 Project Progression Timeline

Activity Chart



Legend:

Task/Part Duration

Milestone

→ Task Dependency/Link

Phase 1 Planning

Phase 2 Schematics and component use

Phase 3 Layout and manufacturing files, PCB ordering

Phase 4 Populating the board and electrical testing

Phase 5 Software

5.3 Final Feedback

This course required for a lot of independent work which required prior Embedded systems knowledge and experience with printing of PCB boards. Both of the team members had minimal previous studies or experience with it, which was one of the major reason project faced major obstacles. More guidance and demonstrations on the design and printing of a PCB board would have definitely helped. In the beginning we did not even realize that we need more guidance or we did not know how to ask. Though we did receive a good amount of help from the team at Fablab who helped with finding parts, assembling and production of the PCB extension board. Most of the manual work on lab was easy soldering but PCB manufacturing and usage of the soldering oven with surface mount component was a new thing for us. The surface mount component for IMU was our only choice in the Fablab and we did not know how hard it is to get it right with the oven but it was also a good lesson how the surface mout component is mounted to the board. On the software side, we should have chosen a board with more I/O as it was decided to use $>$ 4 ICs will be connected to the Nucleo and limited I/O was a major constraint.

Appendices

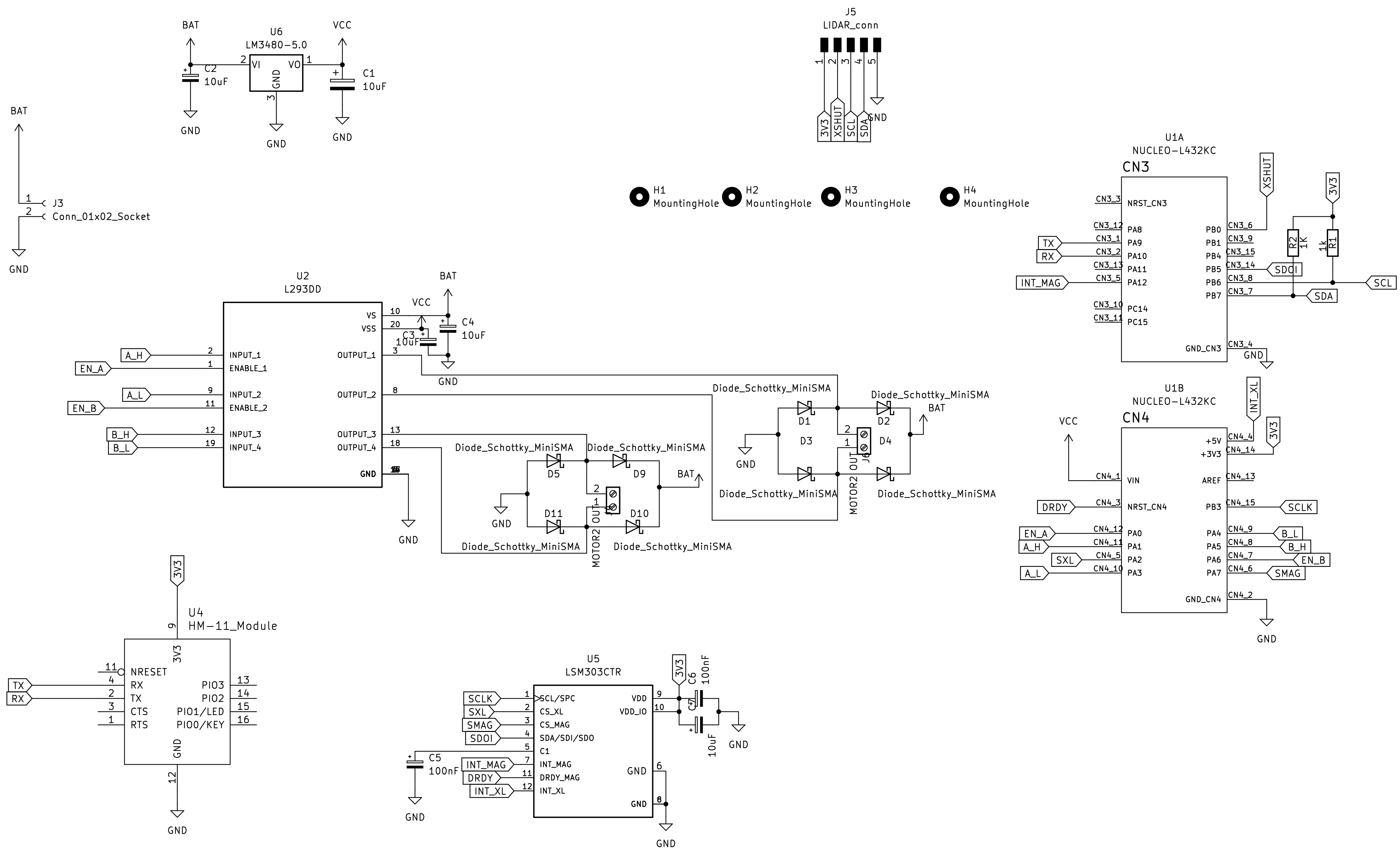
Appendix A

Glossary

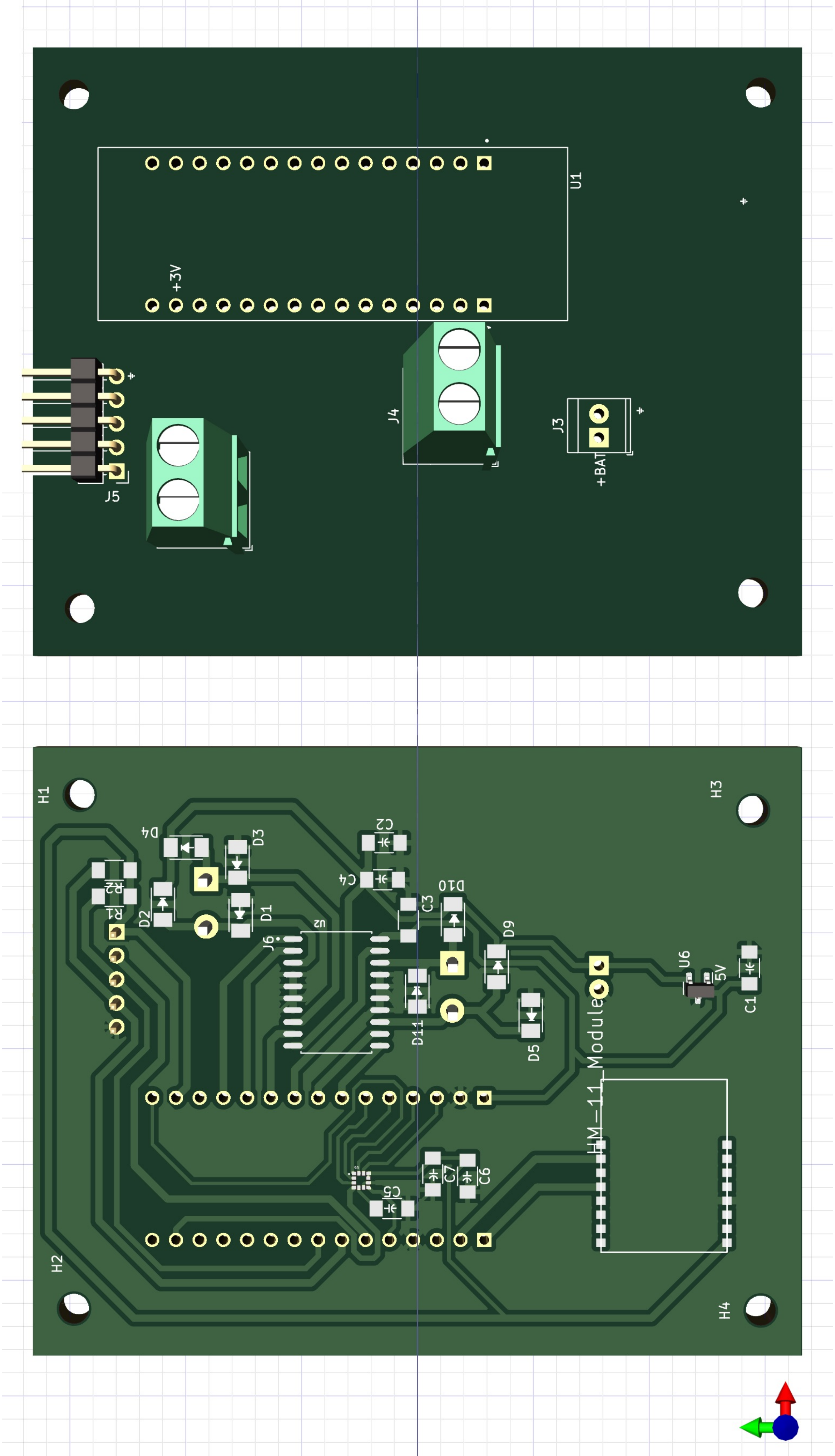
MCU	Microcontroller Unit
IDE	Integrated Development Environment
GPIO	General Purpose Input Output
LOS	Line of Sight
IMU	Inertial Measurement Unit
I2C	Inter-Integrated Circuit
SPI	Serial Peripheral Interface
ACC	Accelerometer
PWM	Pulse Width Modulation
VSS	Supply Voltage
VDD	Logic Supply Voltage
EN	Enable
GND	Ground
MSIO	Multiple Serial Input Output
SDA	Serial Data
3V3	3.3 Volts
V_{cc}	5V at the Common Collector
RX	Receive
TX	Transmit
INT	Interrupt

Appendix B

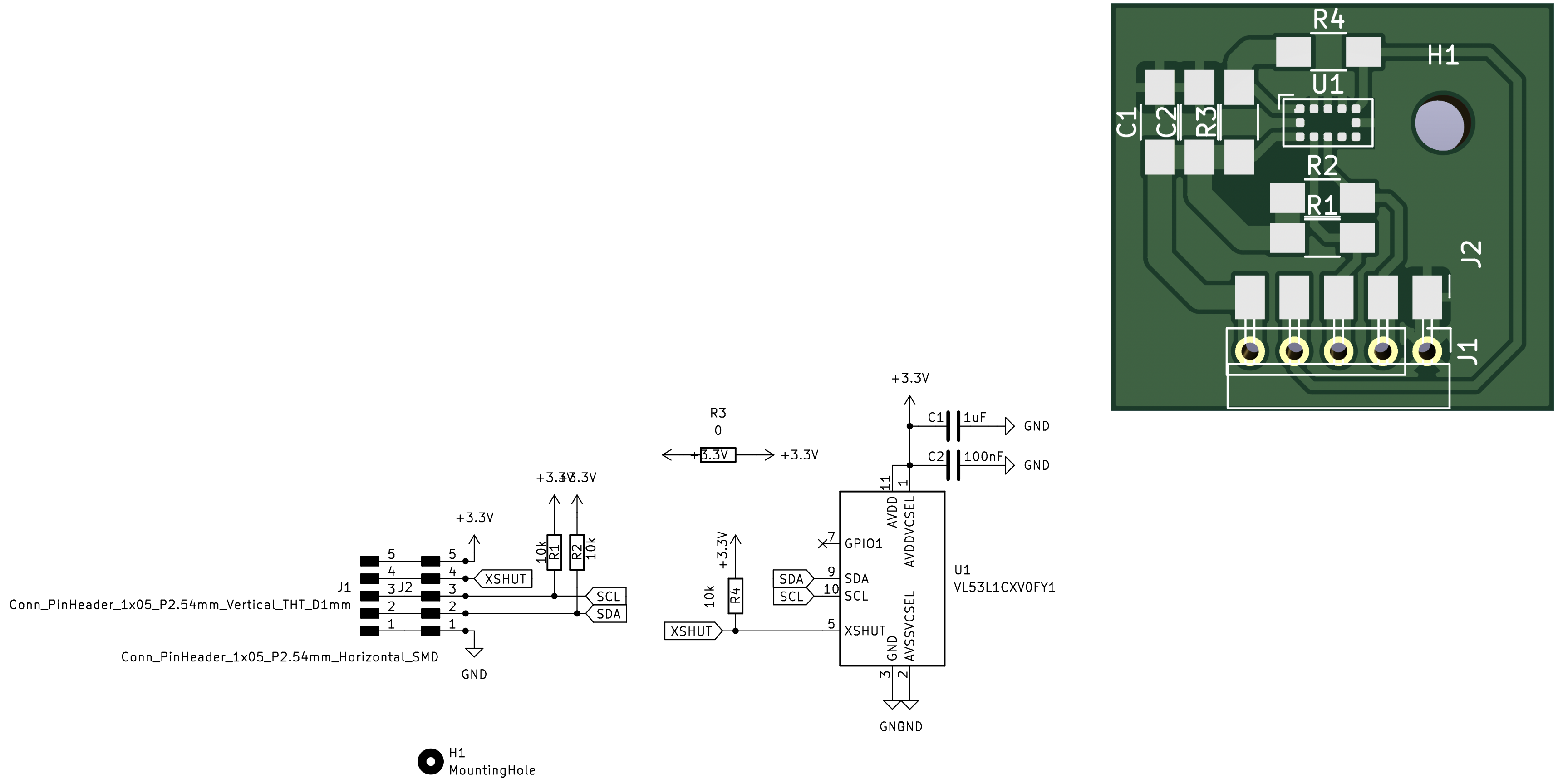
Final Schematic and PCB Layout



U6
LM3480-5.0
C1
10uF
C2
10uF
J3
Conn_01x02_Socket
J5
LIDAR_conn
H1
MountingHole
H2
MountingHole
H3
MountingHole
H4
MountingHole
U1A
NUCLEO-L432KC
CN3
U1B
NUCLEO-L432KC
CN4
R1
1k
R2
1K
U2
L293DD
C3
10uF
C4
10uF
Diode_Schottky_MiniSMA
MOTOR2 OUT
U4
HM-11_Module
U5
LSM303CTR
C5
100nF
C6
100nF
C7
10uF
BAT
VCC
GND
3V3



U1
+3V
J5
J4
J3
+BAT
H1
H2
H3
H4
D1
D2
D3
D4
D5
D9
D10
D11
R1
R2
C1
C2
C3
C4
C5
C6
C7
U2
U6
J6
5V
HM_11_Module



Sheet: /

File: laser_breakout.kicad_sch

Title:

Size: A4 | Date: | **Rev:**

KiCad E.D.A. eeschema 7.0.5-0 | Id: 1/1