TECHNISCHE UNIVERSITÄT WIEN

Institut für Computertechnik
Institute of Computer Technology

Master's Thesis

submitted by

Robin Arbaud

A Mini-Robot using Self-Learning for Navigation

In partial fulfillment of the requirements for the degree of

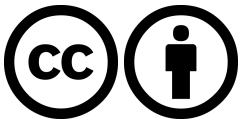
Master of Science (MSc)

Vienna, Austria, 2018

Study code:	066 504
Field of study:	Embedded Systems
Supervisor:	Univ.Prof. Dipl.-Ing Dr.techn Axel Jantsch
Co-Supervisor:	Univ.Ass Nima Taherinejad, PhD

If you find this work useful, please cite it using the following BibTeX entry:

```
@Thesis{AuthorLastName2019,
 type        = {Master's Thesis},
 author      = {Robin Arbaud},
 title       = {A Mini-Robot using Self-Learning for Navigation},
 school      = {Vienna University of Technology (TU Wien)},
 year        = {2019},
 address     = {Gusshausstrasse 27--29 / 384, 1040 Wien},
 month       = {October},
}
```

Contents

List of Figures

List of Tables

Glossary

AC Alternate Current. 23

ADC Analog to Digital Converter. 34, 43, 44

API Application Programming Interface. 43

BGA Ball Grid Array. 15, 22, 30, 47

BLE Bluetooth Low Energy. 29, 32, 43

CMOS Complementary Metal Oxide Semiconductor. 18

CPU Central Processing Unit. 21

DC Direct Current. 17, 23, 29

DSBGA Die-Size Ball Grid Array. 21

DSP Digital Signal Processor. 21

FPGA Field Programmable Gate Array. 23

FPU Floating Point Unit. 21, 45

GPIO General Purpose Input Output. 43, 45

I2C Inter Integrated Circuits. 17, 23, 33, 43–45

IC Integrated Circuit. 15, 22, 25, 30, 32, 34, 38

IMU Inertial Measurement Unit. 17, 18

IR Infrared. 18, 22

LED Light-Emitting Diode. 18, 33, 39, 43

Li-ion Lithium-ion. 24

Li-Po Lithium-Polymer. 17, 18, 24, 29

LVDS Low Voltage Differential Signaling. 23

MAC Multiply and ACcumulate. 21

MCU Microcontroller Unit. 17, 29, 30, 32–34, 43–45

mROBerTO Milli-Robot-Toronto. 18, 23, 28, 33

NN Neural Network. 15, 16, 21, 22, 28, 33, 34, 40

OpAmp Operational Amplifier. 39

PCB Printed Circuit Board. 15, 17, 18, 22, 23, 27–30, 32, 34, 39, 40, 47

PTC Positive Temperature Coefficient. 24

PWM Pulse Width Modulated. 32, 43, 44, 47

QFN Quad Flat No-lead. 21, 22

RAM Random Access Memory. 44

RF Radio Frequency. 17, 29

RGB Red-Green-Blue. 18

RSSI Received Signal Strength Indication. 47

RTOS Real Time Operating System. 43, 44

SDK Software Development Kit. 43

SMD Surface Mounted Device. 47

SoC System-on-Chip. 17, 18, 29

SPI Serial Peripheral Interface. 23

TinyTeRP Tiny Terrestrial Robotic Platform. 17, 18

ToF Time of Flight. 18, 29, 33, 44, 45

UART Universal Asynchronous Receiver Transmitter. 23

VQFN Very thin Quad Flat No-lead. 21, 30, 47

WL-CSP Wafer Level Chip Scale Package. 21, 22

Abstract

The robot described in this thesis is aimed at being used for research on machine learning. The idea was to provide an interface to the physical world to a neural network running on a host computer. This thesis describes the background literature, designed robot, as well as a charging station which enabled the robot to recharge itself. The main focus of this work was to build a robot as small as possible, without designing custom actuators or batteries. The final size of the robot is of 2.7 cm^3, which is much smaller than the smallest robot designed so far with significant sensing capabilities (TinyTERP with extension board, 7.65 cm^3), to the best of my knowledge.

Chapter 1

Purposes and Requirements

1.1 Motivations

The application domain of the designed robotic platform is Developmental Robotics. This particular field of robotics studies aims at building robots that evolve to mimic the development of the human brain [1]. Research has already showed that physical interaction is a key point, hence the necessity to build suitable robotic platforms. While there are some much more advanced robots around, like iCub [2], a simple system like the one presented in this report is sufficient to study navigation and spatial perception.

A second key element in developmental robotics is prediction and error correction. Algorithms using this scheme seem to work best [1]. Therefore, the primary goal assigned to this robot would be to accurately predict the evolution of sensory inputs, first in the next cycle, then as far as possible in the future. A second goal would be to recognize charging stations on the field and be able to self-manage the battery load, by driving onto a charging station when the battery is low. A third goal could be the acquisition of some form of spatial representation and exploration.

Next to those application considerations, it was deemed interesting to attempt to build a robot as small as possible. That requires careful selection of actuators, battery and even Integrated Circuits (ICs). All selected components should be small, power efficient, and, ideally, allow easy assembly of the robot. This latter point was especially difficult to abide by, as some of the selected components only exist in Ball Grid Array (BGA) package, and therefore require machine assembly for the Printed Circuit Boards (PCBs).

Optimizing the mechanical design, as well as the PCB layout and routing was a major task.

1.2 Requirements

This robot was designed as an interface to the real world for a machine learning algorithm running on a host computer. Therefore, the purpose was to implement as few primitives as possible in the robot itself. The values measured by the various sensors are sent with little processing to the Neural Network (NN). The inputs to the motors are also quasi

proportional to the integer value sent by the NN. The little processing which was done on the robot was mainly aimed at simplifying the communication between robot and host.

The robot would ideally be able to recharge itself by driving over charging stations scattered over the test field. Therefore, extended battery life was not a target.

Instead, it was decided that the robot should be as small as possible, without building custom components like actuators or battery. The aim was to build a robot of less than 3 cm^3, with an autonomy of 15 minutes between recharges.

Chapter 2

Related Works

2.1 Tiny Terrestrial Robotic Platform

The Tiny Terrestrial Robotic Platform (TinyTeRP) [3] is a very well-designed modular small robot, of size 17x18x20 mm, that is 6.12 cm^3. It was intended to be used in swarm robotics studies.

The drive train is a common symmetrical one, with 4 wheels including two driving ones. The actuators used are 4x8 mm brushed Direct Current (DC) motors. A worm wheel gear transfers power from the motors to the driving wheels.

Power is supplied by a 30 mAh Lithium-Polymer (Li-Po) battery, which is hold in place by magnetic pads. The resulting autonomy of the robot is claimed to be about 35 minutes, but under which conditions is not precisely stated. A bigger, 50 mAh battery can also be used, at the cost of a 2.2 mm height increase.

The base PCB of the TinyTeRP uses a TI CC2533 System-on-Chip (SoC) as main processing unit. This chip includes a small Microcontroller Unit (MCU) and a radio interface. Next to it are the antenna for Radio Frequency (RF) communication, and a dual H-bridge chip for motor control.

A pin header allows to stack additional boards on top of the base one, by providing connection for power, Inter Integrated Circuits (I2C) communication, and debug. This makes the robot modular and easy to adapt to specific use cases. An inertial sensing board comprising a 6-axis Inertial Measurement Unit (IMU) and an additional MCU to process the IMU data was designed as example. Adding this board increases the height of the robot by 5mm, resulting in a robot of 7.65 cm^3.

The possibility to stack boards horizontally on top of each other provides good modularity, however there is a limitation regarding the orientation of the board, and many sensors (cameras, rangefinders for example require that at least one board is mounted vertically, and facing in the sensing direction. Therefore, the TinyTeRP is not suitable to carry such sensors.

2.2 Milli-Robot-Toronto

The Milli-Robot-Toronto (mROBerTO) [4] is slightly bigger than the TinyTeRP, with a size of 16x16x32 mm, that is about 8.19 cm^3, however it has a couple of additional features.

The drive train is simplified to the maximum: the motors shafts are directly in contact with the floor and act as wheels, without additional gear trains. A third contact point is provided by a polymer ball. This ball is actually not tied to the robot, rather the robot simply lays onto the ball.

The robot holds three Li-Po batteries, for a total capacity of 120 mAh, allowing about 1.5 hours of autonomy.

Like the TinyTeRP, the main processor is a SoC with integrated Bluetooth interface, namely the nRF51422 from Nordic Semiconductors. Next to it are two sensing modules, one fixed and the second switchable. The fixed sensing module includes a Complementary Metal Oxide Semiconductor (CMOS) camera and a VL6180X Time of Flight (ToF) sensor. For the interchangeable module, 2 PCB were designed, one featuring a 9-axis IMU and two Light-Emitting Diode (LED), the other being specifically intended for swarm robotics studies, with 8 Infrared (IR) emitters and 6 IR detectors.

The LED can be used as part of an external absolute positioning system. A camera watching the field from the top can detect the LED and deduce the position of the robot, which can then be sent back to the robot via Bluetooth. With one Red-Green-Blue (RGB) LED, it is possible to assign a unique LED color to each individual robot to differentiate between individuals in a swarm context.

2.3 Zooids

The Zooid [5] is a peculiar robot, developed for swarm user interface. A swarm user interface is basically a set of robots, which can behave as both a way for a computer system to display information, and for a user to interact with the system. A short video illustrating this is available on the ACM publication page. The Zooids are of cylindrical shape, with a diameter of 26 mm and a height of 21 mm. The volume is therefore of about 11.15 cm^3.

The drive train of the Zooid is interesting, as the wheels axes are not colinear. The wheels are directly attached to the motors shafts, and the motors are mounted side by side to reduce the size of the robot. Two small metal balls clipped into the chassis provide additional contact points. The non-colinearity of the wheels axes causes difficulty in modelling and requires a relatively complex control algorithm.

The embedded battery has a capacity of 100 mAh, allowing one hour of autonomy in continuous movement.

The main processor is a STM32F051C8 microcontroller. Contrary to previous designs, the radio chip is separate. Concerning sensors, the Zooids are wrapped in an electrode enabling capacitive tough sensing on the whole side of the robot. A specific chip is in charge of processing this data.

The Zooids positions are determined through an external system roughly similar to the one proposed for the Toronto robot, but reversed: instead of an external camera on top of the field, and LED on the robot, the Zooids feature two photodiodes, and a projector is used to illuminate the field with a sequence of gray-coded patterns. Each Zooid can

decode these patterns to determine its location. Compared to the external camera method, the latency of transmitting position feedback from the camera to the robot is suppressed, enabling easier and better position control.

Chapter 3

Components Selection Considerations

3.1 Main Processing Chip

The main processing chip was selected on 3 main criteria, besides the available peripherals. Those are the Central Processing Unit (CPU) capabilities, the output drive current delivered by the chip, and the package size. At some point, building an autonomous robot was discussed. Such a robot would not need Bluetooth communication, but should run a neural network on its embedded processor (the training of the NN was still planned to be done on a host computer, but once the robot was trained, it would have been autonomous). Such a robot would require a relatively fast processor, and would benefit a lot from Floating Point Unit (FPU) or Digital Signal Processor (DSP) extensions (as Multiply and ACcumulate (MAC) is a large part of NN output computation). However, this idea was dropped when it became clear that the target size of 3 cm^3 could not be reached with a powerful processor, which would consume too much power and have a too large package.

Component	manufacturer	CPU	max drive current	Package Footprint
PIC32MK	Microchip	MIPS 120 MHz with FPU, DSP	22 mA	9x9 mm (QFN)
SAMD5X	Microchip	MIPS 120 MHz with FPU	8 mA	3.6x3.5 mm (WL-CSP) 7x7 mm (VQFN)
XMC1300	Infineon	ARM-M0 32 MHz with math co-proc.	50 mA	5x5 mm (VQFN)
XMC4200	Infineon	ARM-M4 80 MHz with FPU, DSP	30 mA	7x7 mm (VQFN)
nRF52810	Nordic Semiconductors	ARM-M4 64 MHz	9 mA	2.5x2.5 mm (WL-CSP) 5x5 mm (QFN)
CC2640R2F	Texas Instruments	ARM-M3 48 MHz	8 mA	2.7x2.7 mm (DSBGA) 4x4 mm (VQFN)

Table 3.1: Main processing chips characteristics comparison

The processors presented in table 3.1 are the ones I had looked into. The PIC32MK and both Infineon chips were interesting because the output drive current is high enough to connect the motors directly to the IC pins. That could have allowed to remove the H-bridge from the circuit and thus to reduce the amount of PCB area needed. The SAMD5X has lower drive current, but is available in a small Wafer Level Chip Scale Package (WL-CSP) package. However, WL-CSP or BGA packages types are very difficult to route without having a multiple layers board, with blind or buried vias. Those requirements increase both the cost and the thickness of the PCB. All things considered, sticking to a simple 2-layers board was preferred, so a Quad Flat No-lead (QFN) package type was chosen for the main processing chip.

After deciding that the NN should run on a host computer, the most important thing for the main processing chip became an integrated Bluetooth interface. The last two references in table 3.1 are the ones featuring a Bluetooth interface. The CC2640R2F from TExas Instruments was chosen for the smaller size available in QFN package type.

3.2 Sensors

3.2.1 Time-of-Flight Range Sensors

There are several technologies for measuring ranges, the most prominent ones being stereo cameras and time-of-flight sensing. Time-of-flight sensor can use either ultrasound or electromagnetic (mostly optical or infrared) waves. Electromagnetic sensors have better precision and faster response times, while ultrasound ones work much better on transparent surfaces like glass. Also, ultrasound sensors are much bigger, as some mechanical components are required to emit the sound pulse and detect its return, therefore, electromagnetic sensors are the only reasonable choice for miniature robotics.

Two different sensors were considered, both manufactured by ST Microelectronics: the VL53L0X and the VL6180X. The latter was selected in the end. ST Microelectronics claims that the VL53L0X is the smallest range sensor available. The main characteristics of those two sensors are summarized in table 3.2.

Sensor	Package Size	Minimum Range	Power Consumption (average, ranging)
VL53L0X	4.4x2.4x1 mm	2 m	19 mA
VL6180X	4.8x2.8x1 mm	10 cm	1.7 mA

Table 3.2: Optical Time-of-Flight Sensors characteristics comparison

As those sensors are using near IR wavelengths, the actual range is dependent on environmental conditions (global illumination) and on target reflectance, hence the specification of a minimal range. However, the accuracy of the measurement is not affected by those parameters. The power consumption can also be altered by tweaking some sensors settings, like measurement frequency and maximum convergence time. The latter parameter express the longest expected time between pulse emission and return signal detection. Shortening it can reduce the power consumption, at the cost of diminished range. The values written in table 3.2 are extracted from the components datasheets.

3.2.2 NanEye Camera

The NanEye camera is a tiny camera module from AMS, mainly aimed at medical applications. It is available as a single camera, or as a stereo camera. The integration of both cameras on a single die makes calibration very easy for stereo depth measurement. The footprint is 1x1 mm for the single camera, and 2.2x1 mm for the stereo camera.

The NanEye communication protocol is not standardized at high level. The electrical interface follows the Low Voltage Differential Signaling (LVDS) standard, but higher level layers are not standard. The data length, start and stop bits are specific to the camera. Therefore, the camera cannot be connected to a simple processor. Some Field Programmable Gate Array (FPGA) is required, with built-in custom hardware for interpreting the received data.

While using a Zynq chip from Xilinx, including a processor and an FPGA was considered, this option was dropped because the available packages had a footprint too large for the targeted size of the robot.

3.3 Actuators

3.3.1 Direct Current motors

As building an Alternate Current (AC) power rail onto the robot would require a lot of circuitry, DC motors were the obvious choice. The smallest motors I could find were the ones manufactured by Precision Microdrives [7]. I reused the 4x8 mm cylindrical motors used in the mROBerTO design. I considered some alternatives, like the square 4x9 mm motors, which could have been easily mounted on a PCB, and feature a 1.5 V rating instead of the 3 V rating of the cylindrical shaped motors. This yields less power consumption, especially at lower speeds.

Using a device with integrated motor and gear train was also considered. This would have allowed to reduce the current supply required to make the robot move. However, the smallest gearmotor available from Precision Microdrives is 6x16 mm. As the size of the robot was considered more important than its autonomy, and as the mROBerTO design showed that using motors directly without gear train was viable, this option was not chosen.

The same company also produces a smaller 3x8 mm model, but those were not available easily, as the minimum order quantity was of a thousand.

3.3.2 Piezo-Actuated Rotary Stage

An other option considered for actuator choice was the Piezo-Actuated Rotary stage from the company New Scales Technology [8]. This innovative system uses an array of piezo-electrical actuators to produce a rotary motion. This system is small, accurate, and can deliver high torque, with a diameter of 12 mm, a resolution of 0.025 deg and a stall torque of 0.04 N·mm. The low-level controller is built into the system which can be interfaced to the processor via I2C, Serial Peripheral Interface (SPI) or Universal Asynchronous Receiver Transmitter (UART) buses, as well as analog control. An absolute position sensor is also integrated.

While this system has plenty of very interesting features, its size and relatively high power consumption (500 mW) were not compatible with the targeted robot size.

3.4 Batteries

Lithium type batteries are always used when size or weight is a major concern. There are two main categories of lithium batteries: Li-Po and Lithium-ion (Li-ion). Li-Po batteries usually weight less, while Li-ion ones are generally more compact, but it is not always so. The main electrical characteristics of a battery are its minimum and maximum voltages, its total capacity (usually in mA·h), and its maximum discharge current (usually expressed as a multiple of the capacity, e.g. for a 10 mA·h battery, a 1 C discharge current means that the battery can deliver up to 10 mA). A range of lithium batteries are available on the PowerStream web catalogue [9].

3.5 Battery Protection

It is not recommended to use a lithium battery directly. Ideally, the battery should not be left unprotected. The main threats to a battery are overdischarge and overcurrent during the discharge (the latter case correspond to a short circuit), and overcharge and overcurrent during the recharge.

The protections needed during the recharge phase are integrated into the charging station (see section 5).

Protection against overdischarge can be implemented in software using the battery gauge, by setting a threshold under which the measured battery state of charge cannot fall. When this threshold gets higher than the measured value, the main processor can shut down all circuitry and enter idle mode.

Regarding discharge overcurrent, a traditional fuse is not a viable solution to embed in a small robot. There are two major drawbacks to that kind of components. First, they are relatively bulky. In any case, they are not suitable for an extremely miniature builds. Second, once they have locked, they require external intervention to unlock, that is if a peak current gets higher than the fuse threshold, then the whole system is down until someone puts it back into function.

A current limiting circuit like the one described in section 5.2 is not suitable, as such a circuit limit the current supplied to a system, but not the current drawn from the power source (since excess current is redirected to another circuit branch).

3.5.1 Polyfuse

The most suitable solution here is to add a Polyfuse [6] between the battery output and the rest of the circuit. A Polyfuse is basically a Positive Temperature Coefficient (PTC) thermistor. When the current flowing through the thermistor increases, the temperature rises, and the resistance of the Polyfuse gets higher. This effectively opens the circuit when the current is too high. Those components exist in tiny package (SMD 0402), but their efficiency is dependant on the

ambient temperature, and the trip time (the delay between the current threshold overflow and the effective opening of the circuit) is highly variable, depending on the exact model.

In the case of this robot, no protection is added for over discharge current, because a trip current of 100 mA would be needed to be useful, and the Polyfuse with the lowest trip current I could find was of 150 mA.

Some current limiting circuits can also be found integrated in ICs like the TPS63050 power converter used in this build, but again the current limit is too high to be of any use with such a small battery.

Chapter 4

Designed Robot Hardware

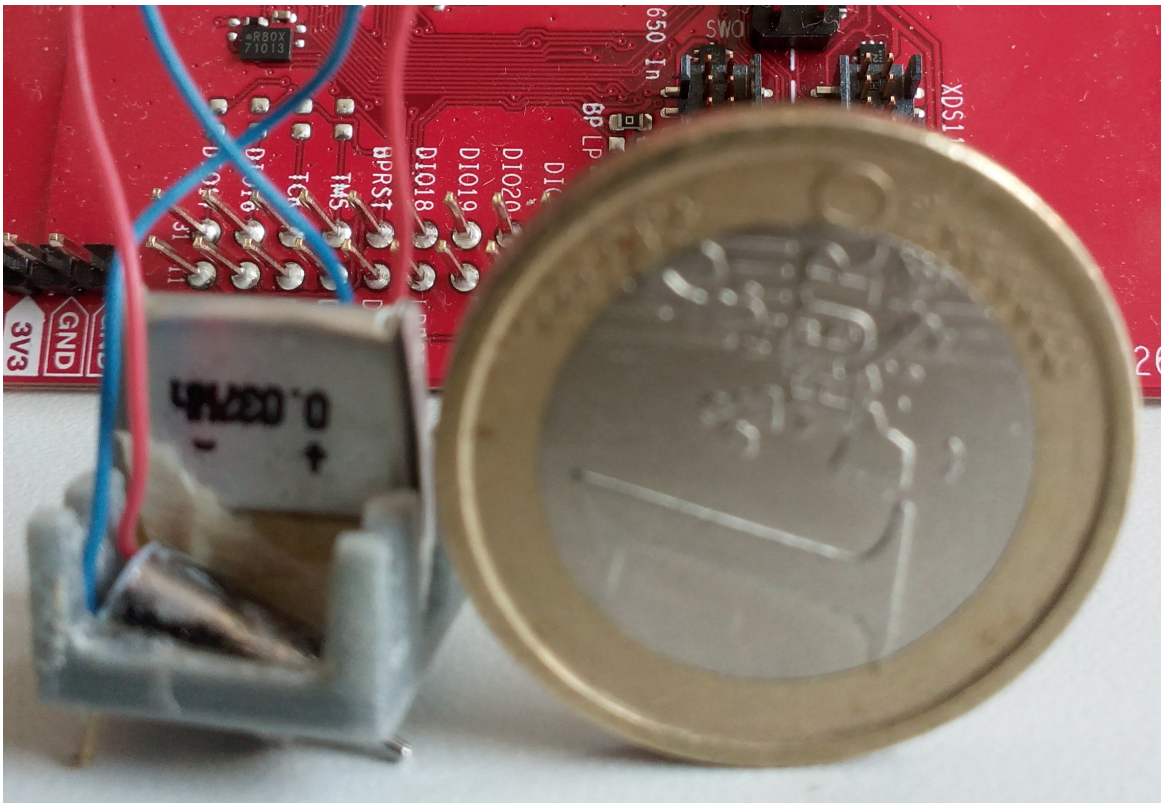


Figure 4.1: Picture of the robot, without its PCBs

4.1 Chassis and Drive Train

The CAD model of the robot is depicted in figure 4.2. The drive train of the robot is non-colinear, similarly to the Zooids, and the motors shafts are directly connected to the ground, as in the mROBerTO. Since the navigation is controlled by a NN, there is no control algorithm to implement on the robot, therefore the non-colinearity of the driving axes is not a problem for building the robot (however, it makes modelling of the robot behavior difficult). Two additional contact points are provided by rivets with a hemispherical head of diameter 1.6 mm, which are glued to the chassis.

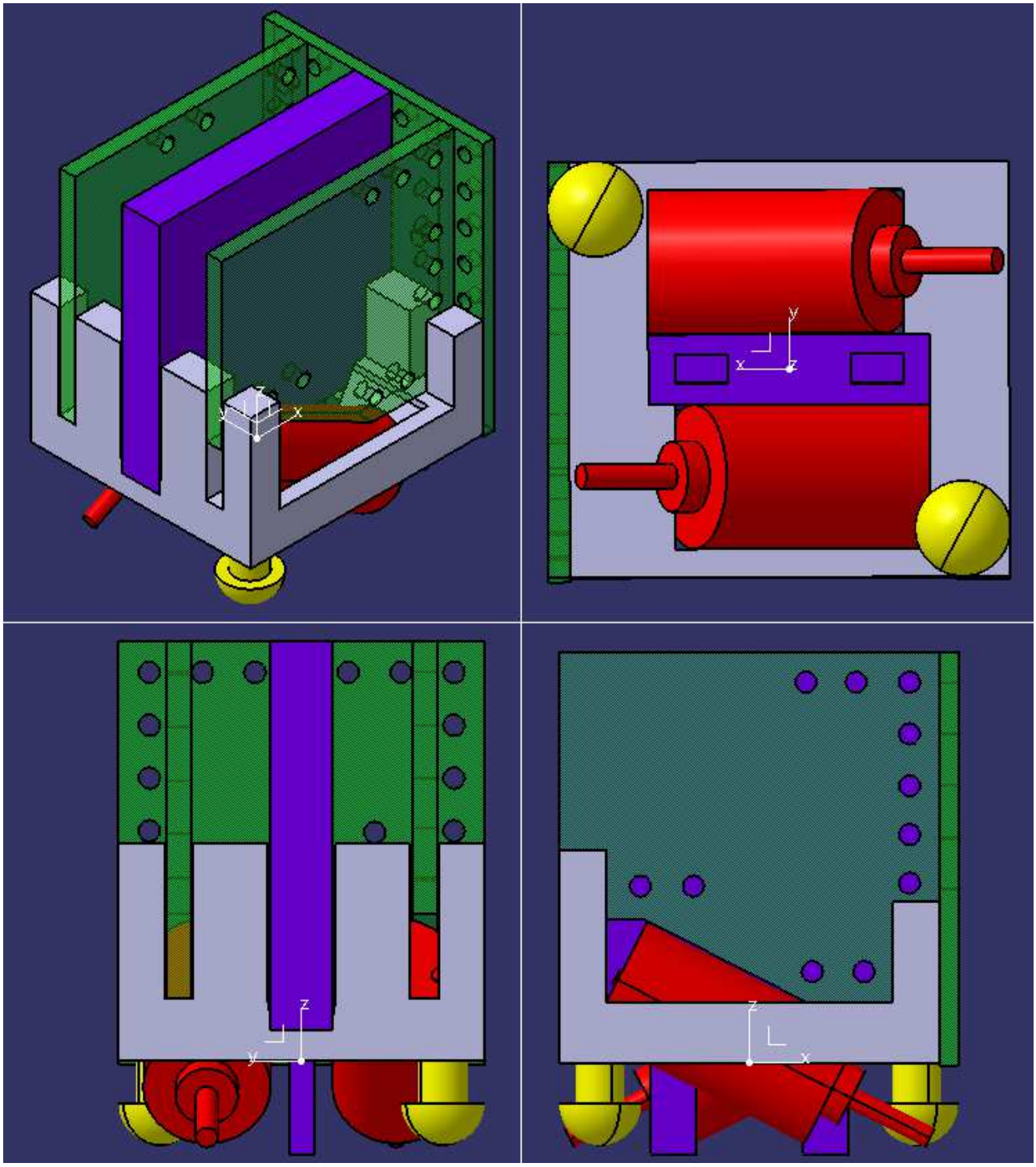


Figure 4.2: Robot CAD model. Isometrical, bottom, side, and rear view. Grey: chassis, green: PCBs, purple: battery, red: motors, yellow: rivets.

The actuators are 4x8 mm brushed DC motors from Precision Microdrive. The chassis was designed to be 3D-printed. It notably defines the positioning of the motors so that the angle is right. The tip of the motor shaft should be below the shoulder of the motor body, but the tilt should be as small as permitted by this condition, to reduce the height of the robot to the minimum.

The battery electrodes should touch the ground lightly, so that they do not hinder the motion of the robot, but also to have a large enough contact surface with the charging station. Were the electrodes above ground, the contact would be quasi punctual, and therefore unreliable and more resistive than ideal. With slightly longer electrodes, they should bend a bit when the robot drives over the charging station electrodes (which are 0.6 mm diameter copper wires). That way we can achieve a more linear contact, for increased reliability and power efficiency.

The battery is a 10 mAh high-discharge Li-Po of 2x12.5x12.5 mm from General Electronics Battery. It can deliver up to 100 mA (10C), but the robot should need less than that, even though the peak current consumption might be close to it. This is because the motors and the ToF sensors can require high current for short time, for acceleration or to initialize measurement. To function during the targeted 15 minutes, the mean power consumption should be under 40 mA.

The total size of the robot is of 16.3x14.1x11.8 mm, that is 2.71 cm^3. This is much smaller than all the robots described in section 2.

The actual PCBs were not made. Only a larger prototype version was assembled for testing the circuit. Therefore, the built robot lacks its PCBs. Apart from that, the robot depicted in figure 4.1 is consistent with the CAD model. A short video demonstrating the soundness of the wheelbase was taken.

4.2 Printed Circuit Board

The electronic circuit is split between three PCBs, on the front, right side, and rear of the robot. Apart from the sensors, the PCBs comprises some components for power regulation, motor control, bluetooth communication, programming and debugging for the MCU.

4.2.1 Microcontroller Connection

The selected MCU is the CC2640R2F from TI. The main processor is a 48 MHz ARM Cortex M3. This SoC includes a RF core for Bluetooth Low Energy (BLE) communication, with a Cortex M0. It accepts a wide range of power supply voltage, from 1.8 V of 3.8 V. As a 3.3 V supply voltage would be too high for some sensors, this allows to keep to a single power rail.

The MCU connection is depicted in figure 4.3. It follows mostly the 4XS reference design provided in the MCU datasheet. The only difference is that the reference design has a second 32 kHz crystal used for RF communication. This crystal is optional, as the chip features an internal RC oscillator. However, this requires some software mitigation to correct the drift of the internal oscillator, see section 6.1.

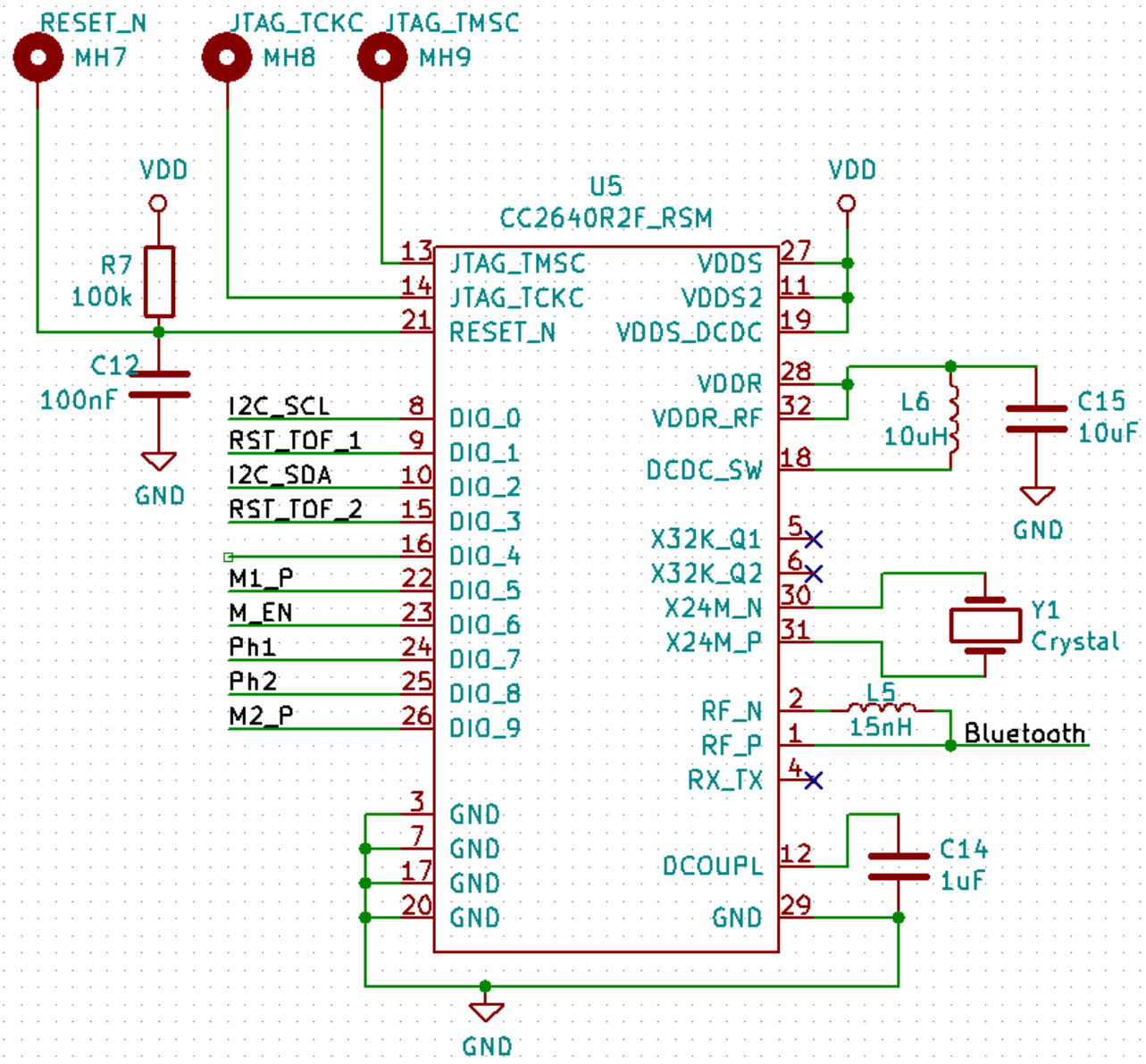


Figure 4.3: Schematic of the Microcontroller Connection

The 4XS reference design is well-suited for this application where the area available on the board is limited. The 4 refers to the version of the MCU, which is the 4x4 mm Very thin Quad Flat No-lead (VQFN) package. The XS refers to the bluetooth connection, which is a single-ended line with external biasing. The inductor L5 serves as bias.

There is an even smaller version of the same IC, but the BGA package is hard to route with only two copper layers on the board. Using more layers would have increased the cost, and the thickness of the PCB, so sticking to a standard dual-layer board was better.

4.2.2 Power supply

The battery voltage is regulated by a TPS63050 buck-boost converter from TI, with adjustable voltage output. It can provide a stable 2.5 to 5.5 V output voltage, with an output current of 500 mA. As the battery voltage can vary from 2.8 to 4.2 V depending on its state of charge, such a regulator was required to keep the power rail in a range compliant with all ICs.

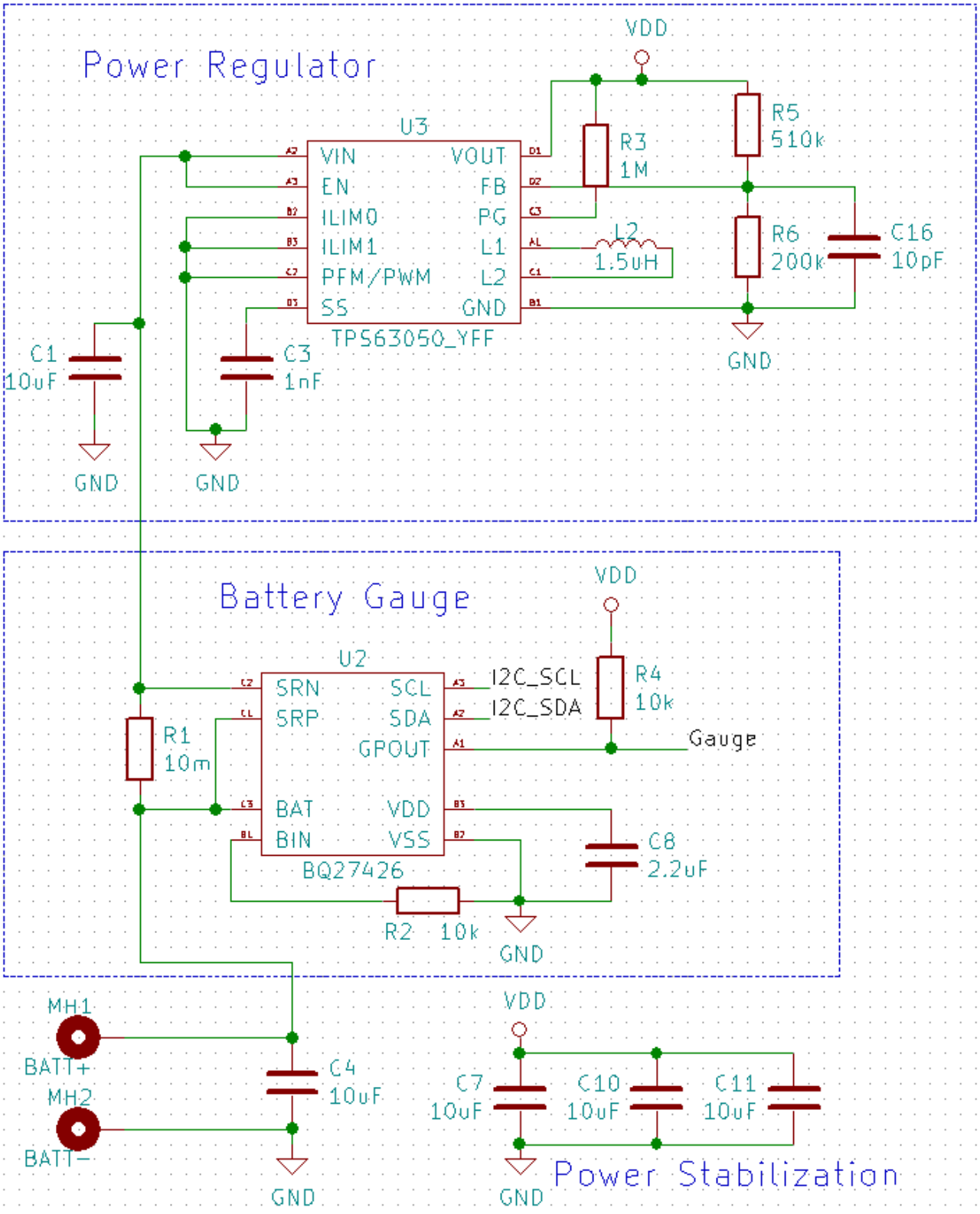


Figure 4.4: Schematic of the Power Regulation and the Battery Gauge Circuits

The circuit is depicted in figure 4.4. The output voltage is tuned through the R5-R6 voltage divider. The device attempts to maintain the feedback pin (FB) voltage at 800 mV. Therefore, the power rail is set to:

$$V_{DD} = \frac{R5 + R6}{R6} V_{FB} \quad \text{with } V_{FB} = 800\text{mV} \tag{4.1}$$

that is 2.8V. The two I_{lim} pins can be used to adjust a current limit, however it is too high to be useful, as the maximum discharge current of the battery is below the lowest current limit which can be set. The enable pin can be used to put the IC in idle mode and to disconnect the load from the battery, as suggested in paragraph 5.5.

The battery gauge itself is described in paragraph 4.3.3.

Capacitors C1 and C4 are used to stabilize the battery voltage. They are both on the front PCB, were the battery gauge and the power regulator are mounted. Capacitors C7, C10 and C11 are stabilizing the 2.8 V power rail at the output of the regulator. There is one per PCB.

4.2.3 Bluetooth Line

The Bluetooth feed line is depicted in figure 4.5. The pair of connection holes is because the antenna is not on the same PCB as the MCU. The line between those should have a resistance of about 50 Ω for a good matching. The two filters on either side of the connection are matching circuits, to optimize the wave transmission coefficients at the 2.45 GHz frequency used by the BLE protocol. Both those filters are reused from reference designs, of the antenna for the right part, and of the MCU for the left part. However, antenna matching is a difficult topic, and strongly dependant on PCB characteristics (track width, layer thickness, material used) and layout (parasitic components in the vicinity of the Bluetooth line), therefore this design is certainly not optimal. However, considering the size of the robot, a 2 meters range would be more than enough, so optimal transmission is not an absolute need.

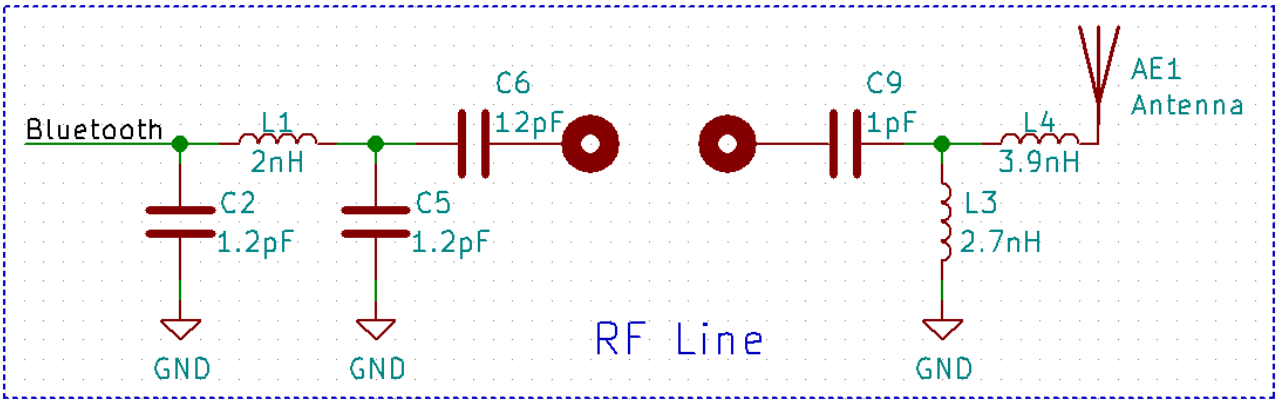


Figure 4.5: Schematic of the Bluetooth Feed Line

4.2.4 Motor Control

For interfacing the motors, a DRV8835 dual H-bridge from TI is used, depicted in figure 4.6. This IC can function in two modes. The selected mode is phase/enable: there is an enable signal to activate the device, and a phase signal which is the driving signal. The phase signal is Pulse Width Modulated (PWM). A duty cycle of 50 % corresponds to a null speed, a duty cycle between 50 and 100 % corresponds to a rotation in clockwise direction, and between 0 and 50 % to a rotation in counter-clockwise direction for the right motor. The left motor has crossed wires, because for a positive duty cycle to correspond to the same motion direction for the robot, it has to correspond to different rotation directions for both motors.

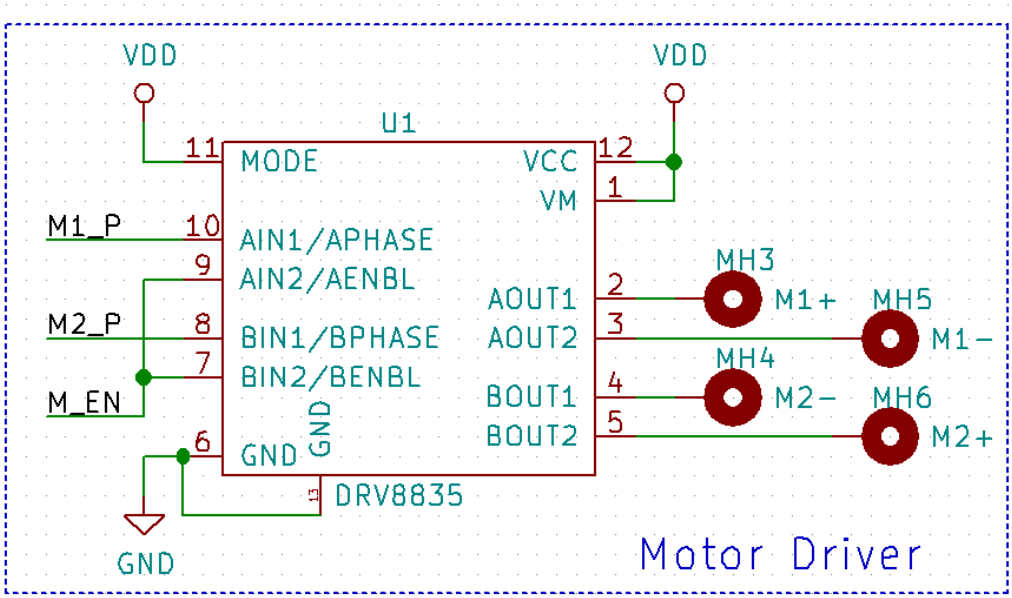


Figure 4.6: Schematic of the Motors Interface

4.3 Sensors

The sensors were chosen based on the learning goals which could be assigned to the algorithm. The first most simple goal would be to predict the future sensors inputs based on the motors voltages and current state of the sensors. At first, the prediction would be of the next measured value, then of as many values as possible. For this purpose, the sensors inputs should vary significantly, and in a consistent way as the robot moves. Also, the measured data should be relatively small in number of bits. For this primary aim, ToF sensors and a magnetometer module serving as compass were chosen.

A second goal of the NN, after being able to predict the evolution of sensory inputs, would be to self-manage the battery charge, in order to remain functional as long as possible. For this purpose, a battery gauge was needed, as well as a way to detect charging stations from afar. This is made possible by a LED beacon on each charging station coupled to two photodiodes on the robot.

The connection of sensors is depicted in figure 4.7 for most of them, and in figure 4.4 for the battery gauge. Most sensors are connected to the MCU through an I2C bus. R10 and R11 are pull-up resistors required by the I2C protocol.

4.3.1 Time-of-Flight Range Sensors

The ToF sensors were chosen for the expected simplicity of learning the way this data evolves when the robot moves. The VL6180X from ST Microelectronics (the same already used in the mROBerTO design) were selected for their low power consumption and good measurement range (almost zero to 150 mm). Two units are placed on the front and right side of the robot.

The Chip Enable (GPIO0/CE) pin is needed to be able to reset one of the sensors while changing the I2C address of the second one, as the default address is the same on both devices.

4.3.2 eCompass Module

In case the robot is too far away from any obstacles, the data collected by the range sensors would not change with any motion of the robot. As this impossibility to evaluate the effects of the NN decisions could impair the algorithm, an electronic compass was added to the robot. This sensory input would change significantly with any turning movement of the robot.

The selected compass is the LSM303AGR from ST Microelectronics. This module includes a 3-axis linear accelerometer and a 3-axis magnetometer. The accelerometer can be used to detect the direction of the gravity field, as tilt compensation is required in a number of applications using a compass, especially on smartphones. Only two axes of the magnetometer are used here. A little processing is then performed on the robot, to convert the magnetic field data into an absolute rotational position (see section 6.3.2). This reduces the number of bytes to transmit via Bluetooth.

4.3.3 Battery Gauge

The purpose of the battery gauge is to make possible for the NN to learn to self-manage the battery charge, to keep functioning for as long as possible. The selected component is the BQ27426 from TI. It was chosen for its small package and ease of use. This IC requires very little configuration, as three pre-programmed profiles are included, one of them suited to the battery.

The 10 mΩ resistor R1 serves as Coulomb counter. Other data such as temperature and battery voltage measured at the BAT pin are used. The BIN pin can be used to connect an external thermistor to accurately measure the battery temperature, but here the IC internal temperature sensor is used. The gauge is placed on the bottom side of the front PCB, that is in the small space between the PCB and the battery, therefore the measured temperature should be accurate enough.

4.3.4 Photodiodes

Both photodiodes are used in photovoltaic mode, that is the generated voltage is measured. This mode is less responsive than the photocurrent mode, where the generated current is measured, but requires no additional components. The photodiodes are directly connected to the Analog to Digital Converter (ADC) embedded in the MCU.

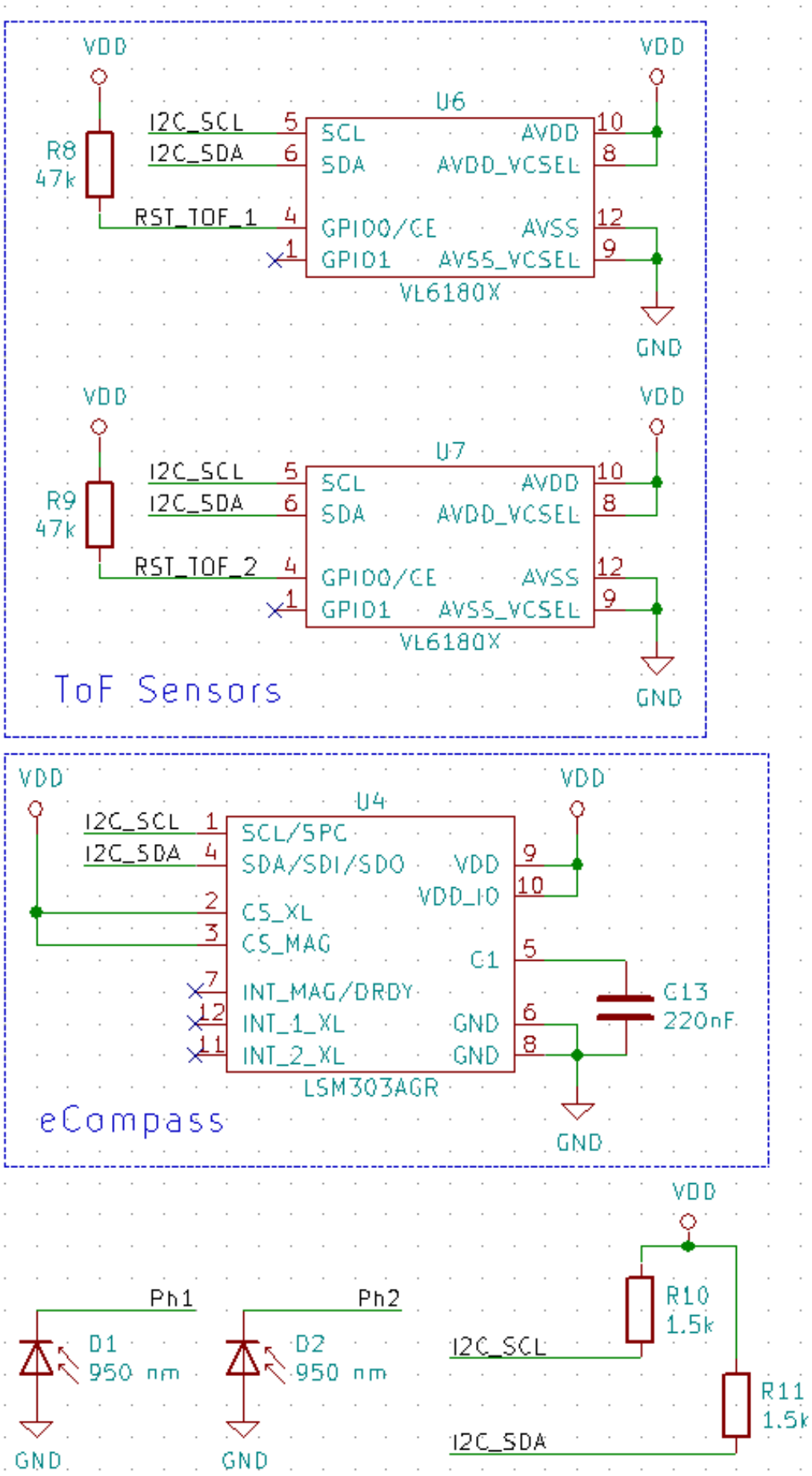


Figure 4.7: Schematic of the Sensors (except the battery gauge)

Chapter 5

Charging Stations

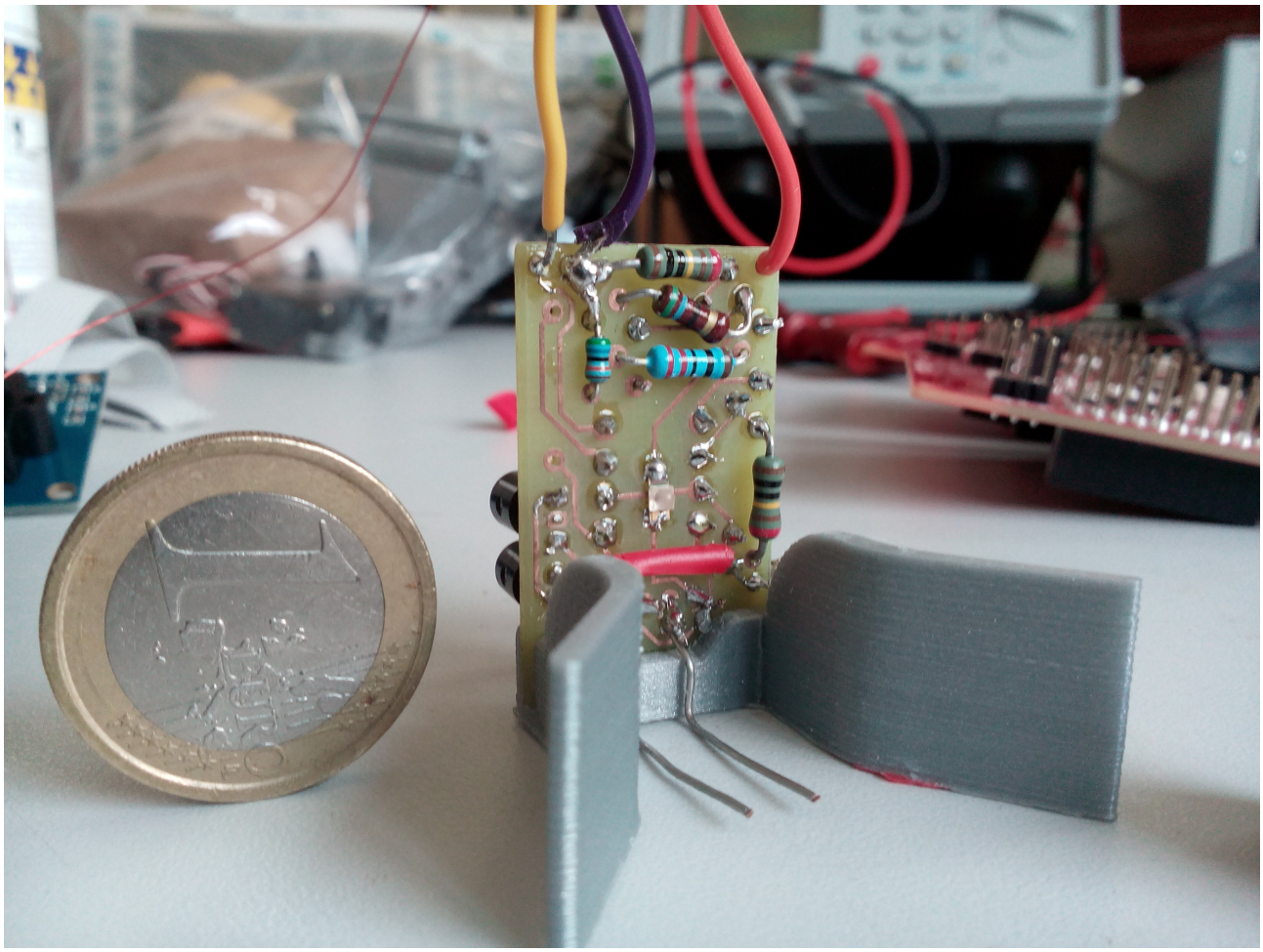


Figure 5.1: Picture of the Charging Station

5.1 Charging Requirements

According to the battery datasheet, the ideal charging profile is in two steps : first, charge under constant current at 0.2C, until the cell voltage reaches 4.2 V. Then, charge under constant voltage at 4.2 V, until the current decreases under 0.05C. The datasheet also indicates that the maximal charge current is 1C, that is 10 mA. While there are complex ICs available to enforce such charging profiles, it is much simpler to build a small circuit limiting both current and voltage, and to let the charge profile free within those independent constraints. So the circuit depicted in figure 5.2 is basically a current limiter coupled with a voltage comparator. It requires a stable symmetrical power supply (+6V; 0V; -6V).

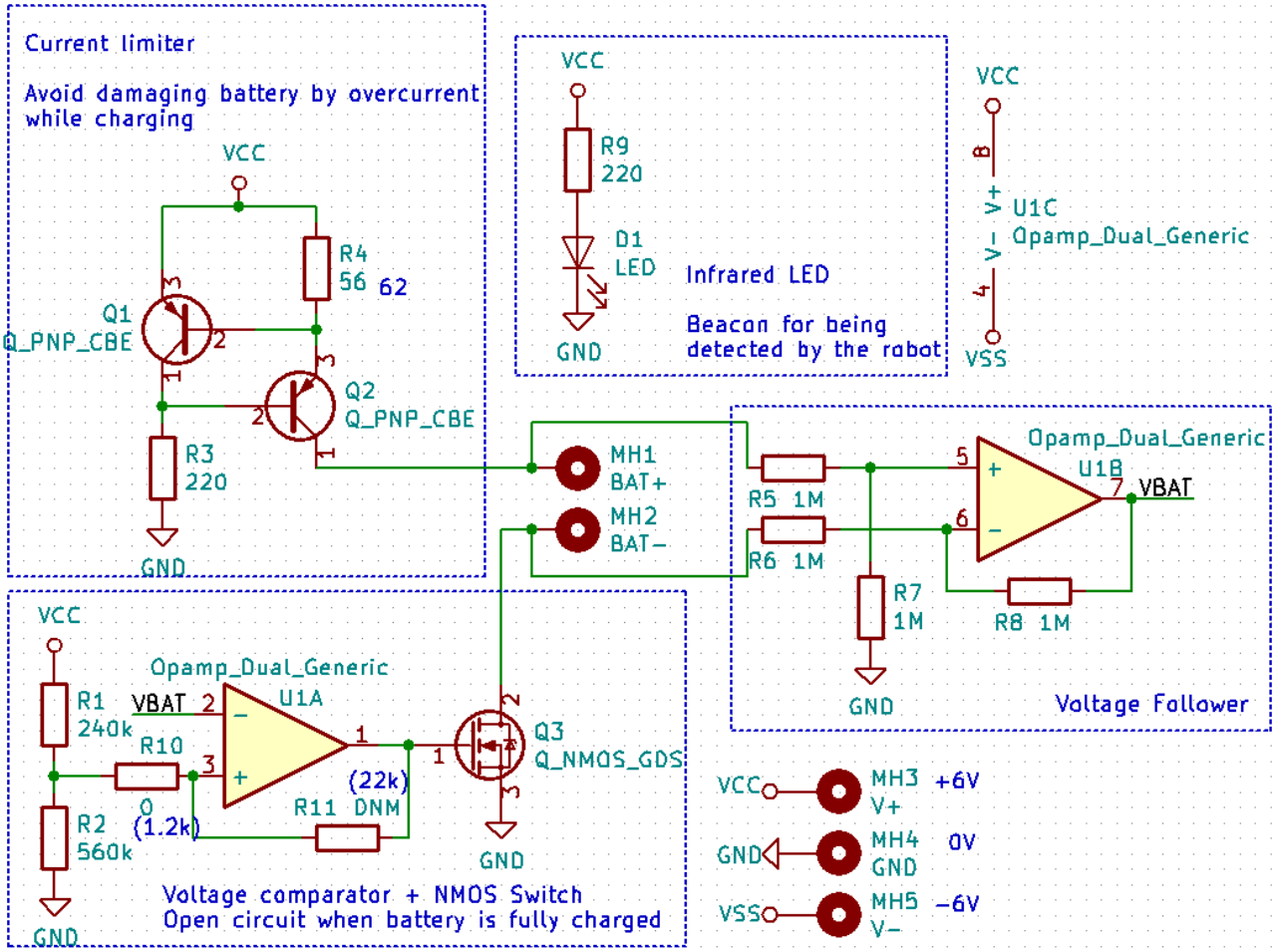


Figure 5.2: Schematic of the Charging Station Circuit

5.2 Current Limiter

The current limiter circuit was designed according to [10]. It comprises two bipolar PNP transistors Q1 and Q2, a biasing resistor R3, and a tuning resistor R4. There are two current paths. The right path goes through R4, Q2, and the load. The left path goes through Q1 and R3. The idea is to limit the current in the right path to 10 mA, and to redirect additional current in the left path.

R4 value is chosen, so that the voltage across R4 equals the base-emitter saturation voltage when the current flowing trough R4 is equal to the maximum admissible current :

$$R4 = V_{BE,sat}(Q1)/i_{max} \tag{5.1}$$

As long as the current in the right path remains below 10 mA, the voltage across R4 is lower than the base-emitter saturation voltage of Q1, and therefore the protection transistor Q1 stays blocking. When the current flowing through R4 increases, so does the base-emitter voltage of Q1, and if the current gets too high, Q1 becomes passing. Then, if Q1 is open, the base-emitter voltage of Q2 decreases, and Q2 becomes blocking. This mechanism regulates the current flowing through the right path, and thus through the load. R3 serves only as biasing to make Q2 passing in normal conditions.

5.3 Voltage Comparator

The voltage comparator itself comprises two main elements. The R1-R2 voltage divider produces a reference voltage to which the battery voltage is compared, according to the equation:

$$V_{ref} = \frac{R2}{R1 + R2} V_{cc} \tag{5.2}$$

The Operational Amplifier (OpAmp) U1A performs the actual comparison between both voltages. R10 and R11 are optional, and can be used to add a hysteresis loop to the comparator. The NMOS switch Q3 is not part of the comparator itself. It opens the circuit when the battery voltage V_{bat} gets larger than the reference voltage of 4.2V.

5.4 Additional Components

Next to the main elements previously described, a few additional components are mounted on the charging station PCB. A voltage follower circuit is made from the second OpAmp U1B and the resistors R5 to R8. This voltage follower converts the differential voltage between the two battery pads into a single ended voltage of the same value. Using only the BAT+ pin voltage and connecting this pin to the negative input of the OpAmp U1A was tested, but the variations of the drain-source voltage of the NMOS switch Q3 proved to be too important to get an accurate image of the battery voltage that way.

A LED D1 serves as beacon, so that the charging station can be detected from afar by the robot. Resistor R9 makes sure the current flowing through the LED is about 20 mA, as specified by the LED datasheet.

5.5 Necessary Refinements

The circuit described so far has been thoroughly tested and works fine to recharge a battery. However, once the battery is mounted onto the robot, a problem arises. The robot power consumption is higher than the 10 mA supplied to the battery. Therefore, unless we put the robot in some kind of idle state, the battery will keep discharging even when the robot is on a charging station, only slower. Putting the robot in idle state is not a good option for two reasons. First, the whole control of the robot is performed by a NN, and implementing this algorithm with as few primitives as possible was one of the purposes of this work. Hard-coding a transition to idle state upon detection of a charging station would contravene this purpose. Second, it is not necessarily clear when the robot is on a charging station or not, so the NN would (at the best) take a long time of training to learn to detect that the robot has reached a charging station.

The proposed mitigation is to increase the maximum current delivered by the charging station, so that it is equal to 10 mA plus the minimal current consumption of the robot. That way, the battery would never receive more current that it can absorb without damage. However, if the current consumed by the robot changes a lot, for example in case of a significant motor acceleration, the robot will discharge nonetheless. In any case, the charging time would increase.

A better and more elegant solution would be to use a 4 pins connection between the robot and the charging station. With one additional pin, a binary signal could be transmitted from the station to the robot and then used as input to a MOS switch to disconnect the battery from the robot's circuitry. Another possibility would be to connect it to the enable pin of the power regulator, which would have almost the same effect, except that the battery gauge would not be disconnected, as it is connected directly to the battery. With again one more pin, the charging station could supply the battery and the robot's circuitry independently. This would require a wide mechanical, as well as PCB layout, redesign of the robot, because a 4 pins connection between the robot bottom and the floor would not be viable, considering how sensitive to friction the robot is in its current state. Therefore those connection pins would have to be placed on the sides of the robot.

5.6 Mechanical Design

The charging station is depicted in figure 5.1. It comprises a 3D-printed frame, the PCB, and two electrode wires laid on the floor, which should touch the battery contacts when the robot drives onto the charging station (see figure 5.3) The frame serves both as support for the PCB, and to prevent the robot from driving over the charging station sideways. As the electrode wires are longer than the space between the battery electrodes, a short-circuit might occur if the robot drives perpendicularly to the charging station. To prevent this, walls are present on lateral sides of the charging station. As the battery electrodes are not exactly centered on the robot (because of the side PCB), the electrodes of the charging station are placed with a matching offset, that is the wires should contact when the robot is in the middle of the charging station. Some space can be seen on figure 5.3 on the right side of the robot. This is where the side PCB should be fitted.

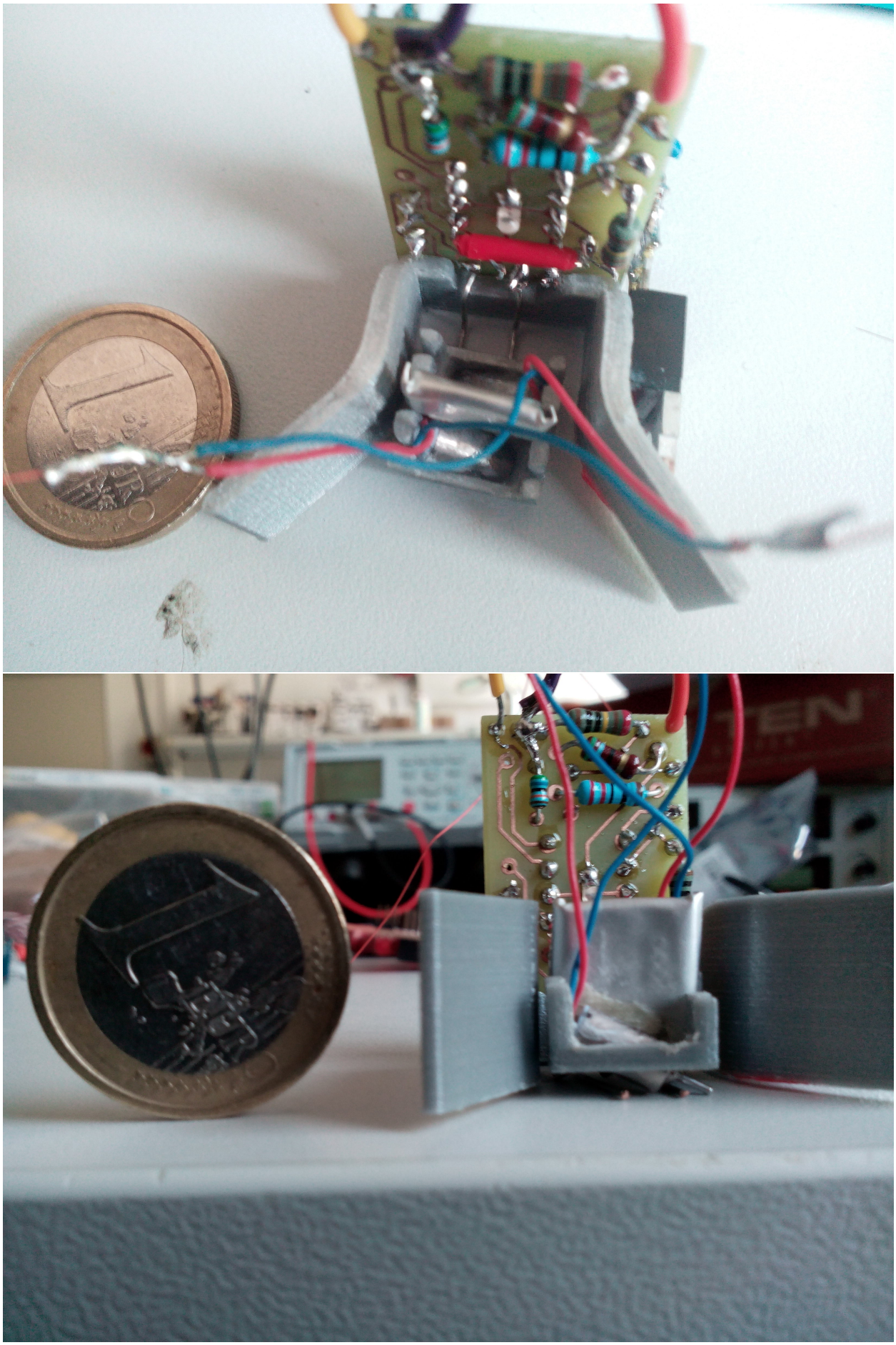


Figure 5.3: Robot in charging position

Chapter 6

Robot Software

6.1 Configuration

The MCU software is based upon the Simple Peripheral demo project provided by TI. This demo can serve as base to develop BLE applications. It uses the TI Real Time Operating System (RTOS), and the TI drivers runtime Application Programming Interfaces (APIs) [11].

The board largely follows the 4XS reference design. The only difference between this and the actual design is the absence of the low frequency oscillator. This oscillator is used by the Bluetooth, and can be replaced by an internal RC oscillator according to a TI application note [14]. However, this requires continuous re-calibration, as the clock drift of the internal oscillator is larger than what is specified by the BLE standards. All the re-calibration code is provided by TI and included in the Simple Peripheral project, the only thing to do to use it is to set the RCOSC build configuration active.

The board configuration is done through board files, which contain some defines and specific data structures used to initialize the drivers. Especially the pin multiplexing is done through these files. The code for the robot uses modified versions of the 4XS files contained in the Software Development Kit (SDK). Modifications are mostly setting the configuration of the General Purpose Input Output (GPIO) pins and adding the data structures for the I2C driver, plus some minor details like changing the pins for the PWM output and ADC inputs.

Using the TI RTOS can make such hardware configuration tasks quite complicated (depending on how cleanly the API are working), since conflicts between drivers or software layers arise easily, with limited error management, especially as some configurations files can be included from the SDK. Typically, the GPIO driver tend to cause troubles with the underlying pin driver, which is used by most other drivers like the I2C driver, for pin multiplexing and configuration. Making a LED blink is actually a difficult task, but setting up the I2C bus was very easy.

6.2 Modifications to the Simple Peripheral Demo Project

The main file was slightly changed: the SimplePeripheral_createTask() function, used to span the Bluetooth demo thread, has been replaced with myThread_create() to span the data acquisition thread. Also, the ICall_init() and ICall_createRemoteTasks() have been removed. Those function initialize the ICall library, which is used for Bluetooth and inter-thread communication. As the Bluetooth antenna is not functional and the application is single-threaded, this library is not needed. Besides, it tend to yield errors. It includes a tool that automatically sets a heap memory size, which can often cause memory access rights violations. Possible mitigations would be to deactivate that tool and to specify a fixed heap size manually, or to configure the system to use flash memory as additional Random Access Memory (RAM). With more RAM available, the automatic setting should work better.

The Board.h, CC2640R2DK_4XS.h and CC2640R2DK_4XS.c files are the board configuration files copied from the SDK, and adapted to specify the pins for I2C, ADC, PWM and so on.

myThread.h contains the main thread. As the bluetooth communication is not working, the values measured by the sensors can be read through the debugger.

Other files (bat.h, commI2C.h, compass.h, motorControl.h, photodiode.h, sensors.h, tof.h) contain functions and data structures used by the main thread.

Some Bluetooth profiles code files were also made, using the code generator from [12], but they are not integrated into the project.

6.3 Data Acquisition

Even though using the TI RTOS caused a lot of troubles for the board configuration, the high-level drivers made everything else quite straightforward.

The photodiodes are directly connected to the MCU ADCs, so the code to read a value is very simple: open the ADC, perform the conversion, close the ADC. Other sensors are connected through the I2C bus, and there again the I2C driver is relatively straightforward.

6.3.1 Time-of-flight Sensor Specificity

Both ToF sensors have the same default I2C address. To be able to change one and not the other, the sensors are enabled one after the other. This is done using the RST_TOF_1 and RST_TOF_2 signals. Those signals are driven by the MCU to force one sensor on idle state while the other is being configured.

The battery gauge and the compass are performing measurements continuously, so when a data update is requested, only read request through the I2C is needed. For the ToF sensors, a continuous measurement would yield high power consumption, as the light pulse emission needed for the sensing draws a significant amount of current. Therefore, range acquisition is performed on demand. When data update is requested, first both ToF sensors are notified that they should

initiate a mew measurement, then all other data is acquired, and at the end data is collected from the ToF sensors. Ideally, the connection between the MCU and the sensor could be used for sending a signal to the MCU that the measurement is finished. That requires some configuration of the sensor, to change the configuration of the pin from chip enable to GPIO, and to dynamically reconfigure the GPIO on the MCU side, to turn it into an input. Unfortunately, this latter part did not work. Therefore, a delay is introduced before reading the signals through I2C.

If an error is raised because the target is too far or too close to the sensor, the returned value is set to 255 or 0, respectively. If an other type of error is returned, the error message is returned with an 'F00' prefix to make the returned signal negative.

6.3.2 Absolute Rotational Position Computation

To extract the angle of the robot with the external magnetic field from the compass raw data, an arctangent computation is required. As a near exact computation is not possible to implement on the limited MCU, an approximate computation is performed according to [15]. To keep the result as an integer, the radians are converted to degrees. As a division by π is not possible without a FPU, the conversion is done by multiplying the result by $3667/64$, where the division by 64 is implemented as a 6-bits shift to the right.

Chapter 7

Conclusion - Results

The PCBs for the robot were not assembled, for reasons of costs and delays. As some components exist only in BGA or VQFN packages, machine pick-and-place and soldering is required. A prototype board has been assembled using as many through-hole components as possible. A first attempt at soldering the Surface Mounted Devices (SMDs) manually, using a stencil and a soldering oven, was not successful. Therefore, assembly of the SMD components was ordered from a company.

Some functioning code was written for this prototype board. The sensors and PWM generation are working. The one thing not functioning is the Bluetooth communication, and the main problem comes from the hardware. I ran some tests using the program SmartRF Studio, which is a utility provided by TI to check RF designs. With the receiver at less than 2 cm from the test board, the Received Signal Strength Indication (RSSI) of the received signal was around -80 dB with a simple carrier wave. Trying to transmit data packets was not successful, even at very close range, as the receiver never detected that a packet was sent. This is because the matching filters are not working properly. It may not be about the values of components used, but rather the fact that through-hole components were used for the capacitors, which according to some posts on the TI forums is unlikely to work for RF communication. The layout also might have some negative influence, since the ideal layout described in [16] is rather different from the prototype board layout.

A robot prototype was built, lacking only the PCBs. It was used to demonstrate that the wheelbase is sound, by showing the robot moving in a short video. Without the electronics, the motors were directly tied to a power supply. Considering how light the robot is, the stiffness and weight of the wires have a significant influence over the robot's trajectory. In the video, the robot can be seen having a curved motion, while both motors are supplied with the same voltage source, therefore the robot should go straight. The curvature is caused by the wires only. With the electronics on board, this external tethering would not be necessary.

Charging stations are working, even though the interface between the charging station and the robot should be improved, to enable decoupling of the robot's circuit from the battery during recharge.

Bibliography

[1] A. Cangelosi, M. Schlesinger, "Developmental Robotics: From Babies to Robots", 2015 MIT Press, ISBN 9780262028011

[2] "iCub - An Open Source Cognitive Humanoid Robotic Platform", iCub.org, last accessed 11/07/2019

[3] A. P. Sabelhaus, D. Mirsky, L. M. Hill, N. C. Martins, S. Bergbreiter, "TinyTeRP: A Tiny Terrestrial Robotic Platform with Modular Sensing", in *2013 IEEE Int. Conf on Robotics and Automation*, 2013 IEEE. doi: 10.1109/I-CRA.2013.6630933

[4] J. Y. Kim, T. Colaco, Z. Kashino, G. Nejat, B. Benhabib, "mROBerTO: A Modular Millirobot for Swarm-Behavior Studies", in *2016 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2016 IEEE. doi: 10.1109/IROS.2016.7759331

[5] M. Le Goc, L. H. Kim, A. Parsaei, J. Fekete, P. Dragicevic, S. Follmer, "Zooids: Building Blocks for Swarm User Interfaces", in *Proc. of the 29th Annual Symp. on User Interface Software and Technology*, 2016 ACM. doi:10.1145/2984511.2984547

[6] Polyfuse PTC Selection Guide, Littlefuse company, EC327-E

[7] Precision Microdrives DC motors catalogue, https://www.precisionmicrodrives.com/product-catalogue/dc-motor, last accessed 22/07/2019

[8] New Scale Technologies Piezo-Actuated Rotary Stage, https://www.newscaletech.com/micro-motion-modules/m3-rs-rotary-smart-stages/, last accessed 22/07/2019

[9] PowerStream batteries web catalogue, section on small lithium batteries, https://www.powerstream.com/ultra-light.htm, last accessed 22/07/2019

[10] Wikipedia article on Current limitting, https://en.wikipedia.org/wiki/Current_limiting, last accessed 15/05/2019

[11] Texas Instruments Drivers API, http://dev.ti.com/tirex/explore/node?node=ADfS7ixpgUmHy56H6NON4Q__krol.2c__LATEST, last accessed 21/06/2019

[12] Texas Instruments Tutorial, "Bluetooth Low Energy Custom Profile", http://dev.ti.com/tirex/explore/content/simplelink_academy_cc2640r2sdk_3_10_01_00/modules/blestack/ble_01_custom_profile/ble_01_custom_profile.html, last accessed 21/06/2019

[13] Texas Instruments Bluetooth Low Energy Software Developer's Guide, http://dev.ti.com/tirex/content/simplelink_cc2640r2_sdk_1_30_00_25/docs/blestack/ble_sw_dev_guide/html/cc2640/index.html, last accessed 21/06/2019

[14] C. Lee and F. Kervel, "Running Bluetooth Low Energy on CC2640 without 32 kHz Crystal", TI application note, SWRA499B.

[15] R. G. Lyons, "Performing efficient arctangent approximation", https://www.embedded.com/design/other/4216719/Performing-efficient-arctangent-approximation, last accessed 11/04/2019

[16] "CC13xx/CC26xx hardware configuration and PCB design considerations", TI application note, SWRA640A.