ARTIFICIAL NEURAL NETWORK-BASED ROBOTICS

A Thesis

presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Electrical Engineering

by

Justin Ng

June 2018

COMMITTEE MEMBERSHIP

TITLE:	Artificial Neural Network-Based Robotics
AUTHOR:	Justin Ng
DATE SUBMITTED:	June 2018
COMMITTEE CHAIR:	Andrew Danowitz, Ph.D. Assistant Professor of Electrical Engineering
COMMITTEE MEMBER:	Xiao-Hua Yu, Ph.D. Professor of Electrical Engineering
COMMITTEE MEMBER:	Fred W. DePiero, Ph.D. Professor of Electrical Engineering

ABSTRACT

Artificial Neural Network-Based Robotics

Justin Ng

Artificial neural networks (ANNs) are highly-capable alternatives to traditional problem solving schemes due to their ability to solve non-linear systems with a non-algorithmic approach. The applications of ANNs range from process control to pattern recognition and, with increasing importance, robotics. This paper demonstrates continuous control of a robot using an actor-critic algorithm based on deep deterministic policy gradients (DDPG) originally conceived by Google DeepMind. The robot performs tasks such as locomotion within an enclosed area and object transportation. The paper also details the robot design process and explores the challenges of implementation in a real-time system.

ACKNOWLEDGMENTS

Thanks to:

- Everyone for everything.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

MECHANICAL DESIGN

1.1 Introduction

The robot meets the design specification shown in Table 1.1. It consists of four subassemblies: the base platform, shooting mechanism, ball hopper, and control unit. Each section was first modeled in SolidWorks, an industry-standard solid modeling CAD program. The designed parts were then fabricated using a laser cutter or 3D printer and assembled with metric hardware. Figure 1.1 displays a photograph of the robot while Figures 1.2 through 1.5 show standard view renders of the SolidWorks model. Note that the robot uses mecanum wheels (a type of omni-directional wheel) which are modeled as plain wheels for simplicity. Figure 1.3 also indicates the front, left, back/rear, and right sides of the robot as referenced in the rest of the paper. The robot contains 53 unique parts, of which 24 are custom designed, and 394 parts in total. The bill of materials can be found in Appendix A.

Table 1.1: Roborodentia 2018 Mechanical Requirements

	Requirement
1	Maximum footprint of 12" x 14" or smaller at start of match but may expand up to 14" x 17" during match.
2	Maximum height of 14" at start of match but no restriction during match.
3	Robot may not disassemble into multiple parts.
4	Robot may not be airborne.
5	Shooting mechanisms may not accelerate balls past 50 feet per second.

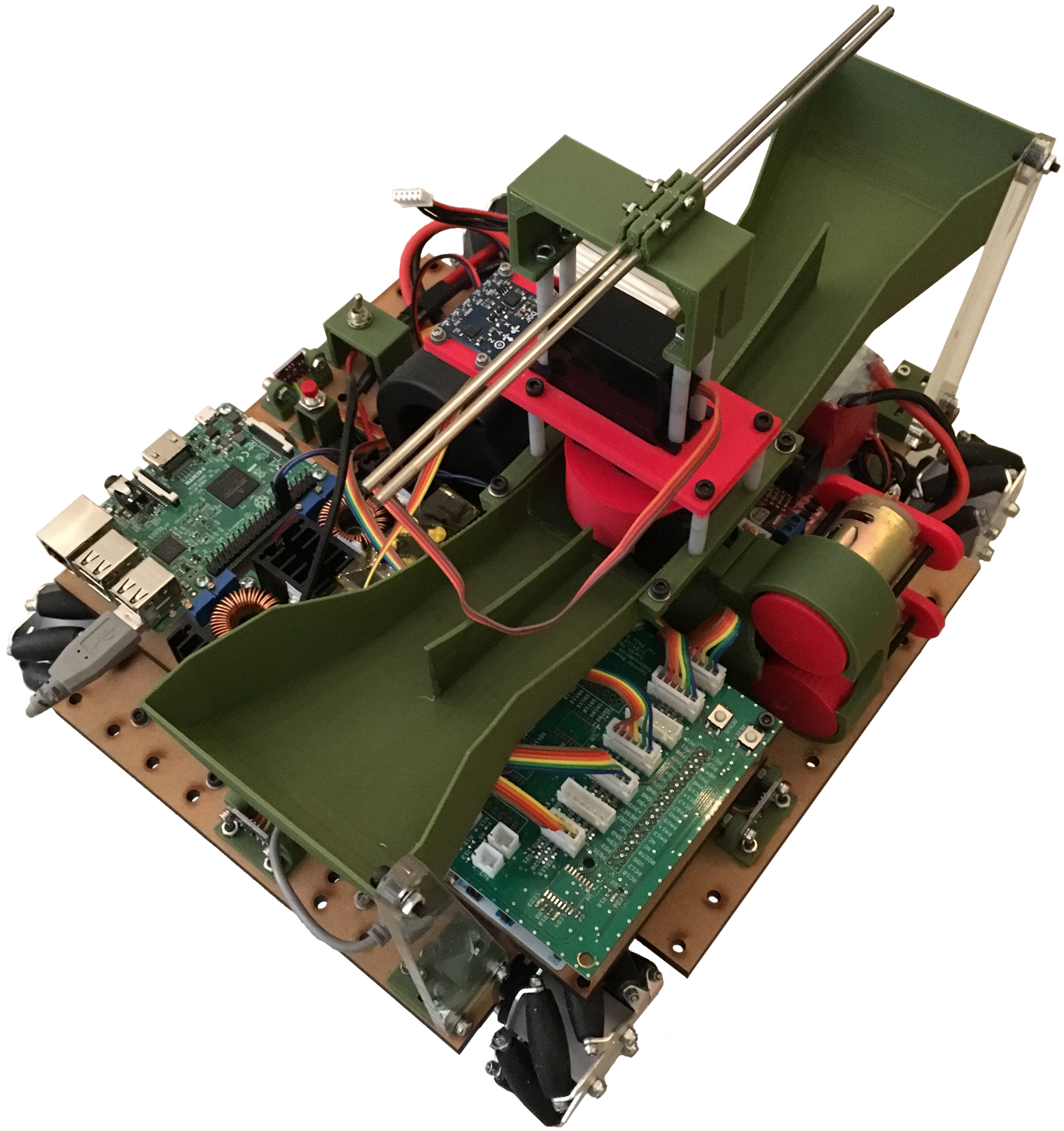


Figure 1.1: Photograph of Robot

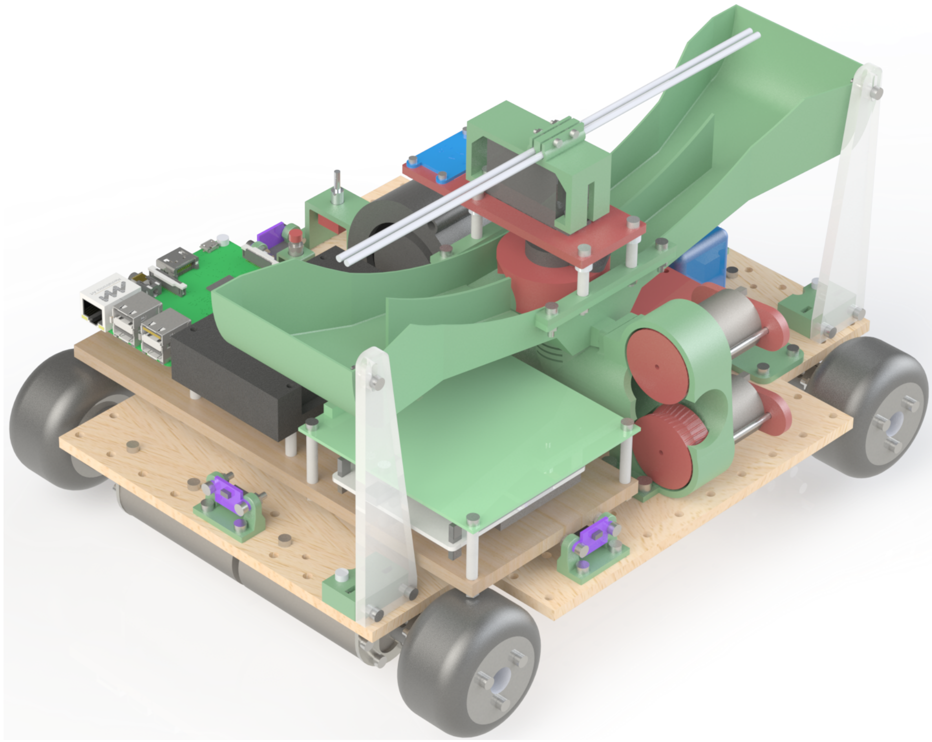


Figure 1.2: Full Robot Render – Isometric View

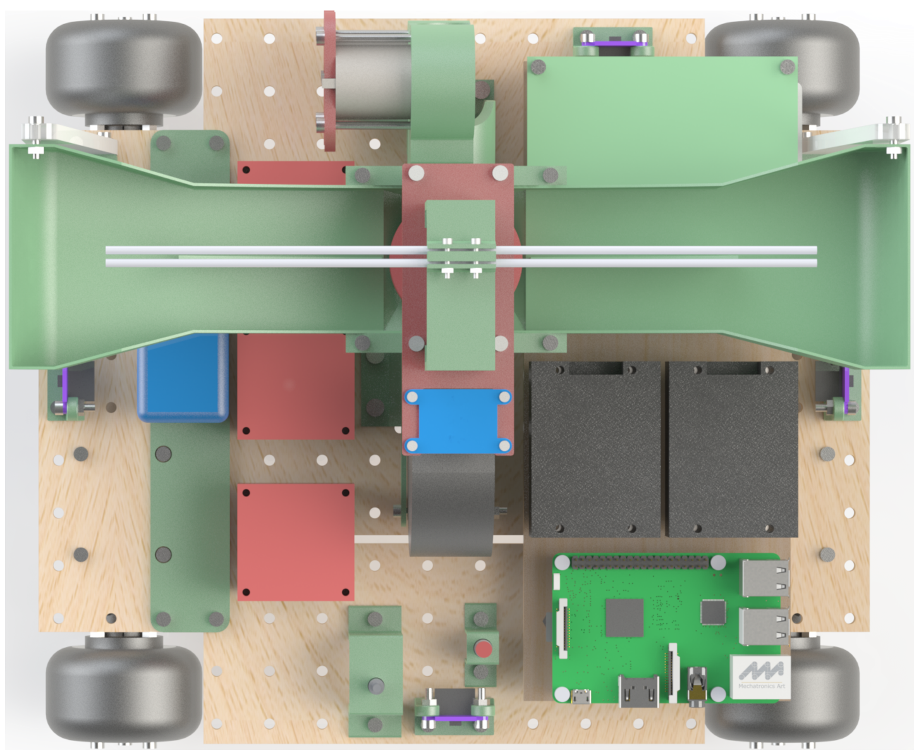


Figure 1.3: Full Robot Render – Top View

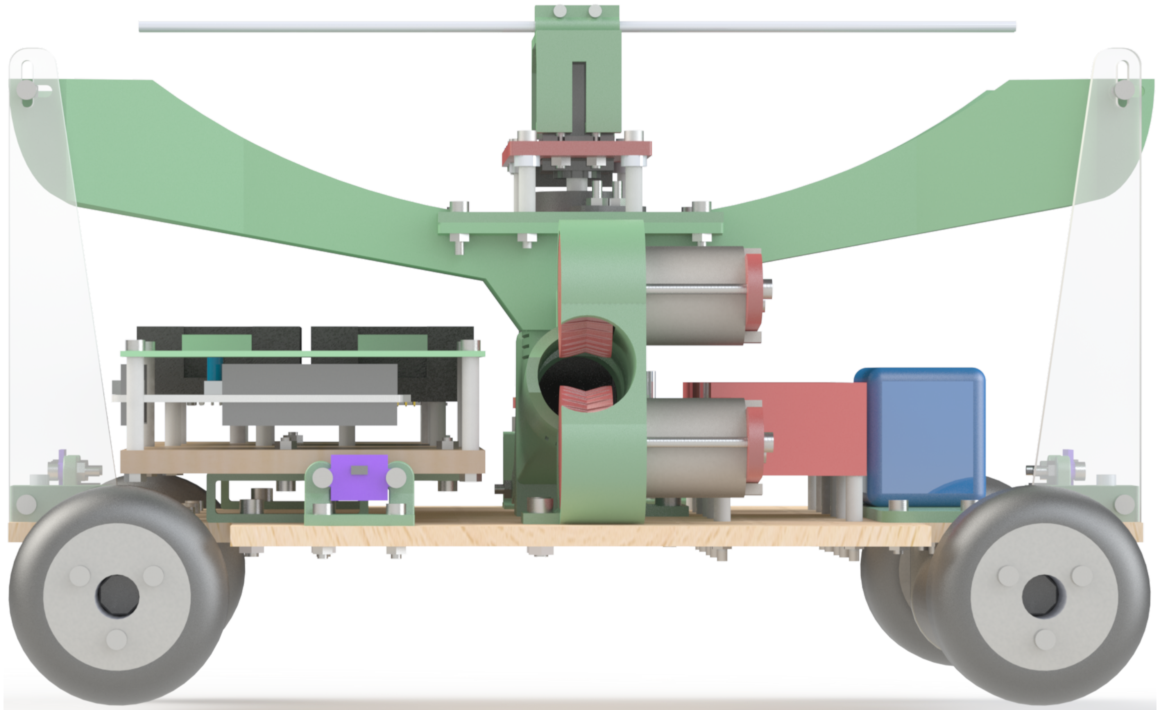


Figure 1.4: Full Robot Render – Front View

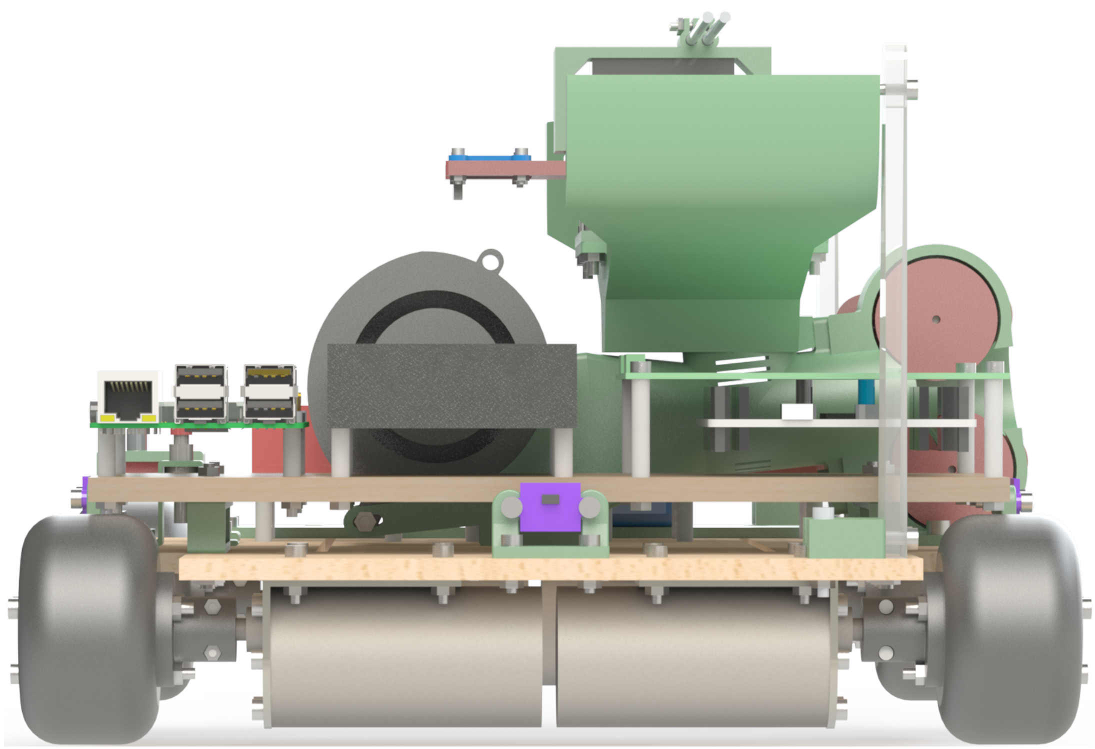


Figure 1.5: Full Robot Render – Right View

The design emphasizes the use of 3D printed parts to take advantage of the benefits of the technology including rapid part production, high part complexity, and low fabrication cost (excluding the cost of the printer, relative to other methods such as machining, casting, and injection molding). Parts were printed on a MakerGear M2 fused deposition modeling (FDM) 3D printer equipped with a 0.35 mm diameter nozzle using MakerGeeks 1.75 mm ABS thermoplastic filament. Print speeds of 20 to 80 mm/s with 0.20 mm layer height, depending on part dimensions and minimum feature size, lead to part print times of 2 to 5 hours for smaller components up to 26 hours for the ball hopper. Due to the non-isotropic strength characteristic of 3D printed components (greater tensile strength in the X and Y axes but significantly weaker in Z), the designs must specifically take into account printing direction and orientation. Additionally, the print volume of 200 mm x 250 mm x 200 mm limits the maximum part dimensions thus requiring the ball hopper to printed as three separate pieces [8].

1.2 Base Platform

The 315 mm x 275 mm base platform of the robot, made from 1/4" thick medium density fiberboard (MDF), serves as the primary structural component and mounting point for the motors, electronics, shooting mechanism, and hopper. The wood is laser cut with a 20 mm grid of 4 mm diameter holes to allow modular placement of components, and the corner cutouts allow clearance for the wheels. Long slots permit wire routing from the motors to the motor drivers above. Figure 1.6 shows the subassembly with labels.

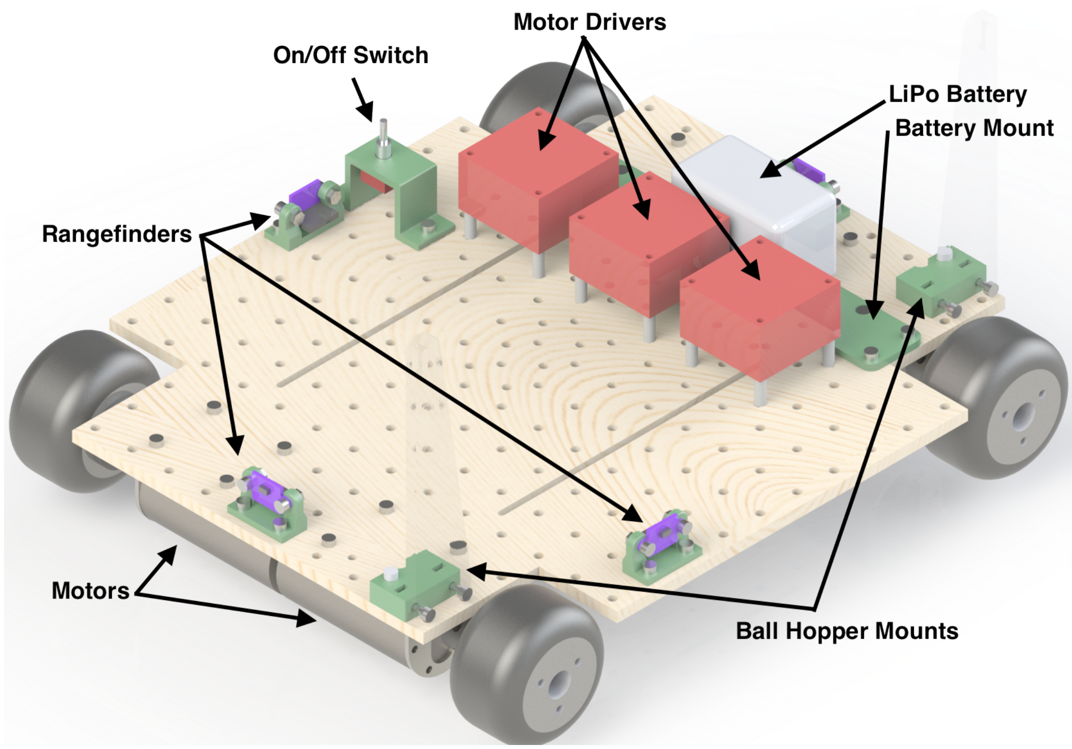


Figure 1.6: Base Platform

Four 12V Pololu 37D motors geared at a 70:1 ratio drive each of the 60 mm diameter mecanum wheels. Figure 1.7 displays an exploded view of the motor assembly. 3D printed couplings, detailed in Figure 1.8, connect the wheels to the 6 mm diameter, D-shaped motor shafts. The couplers use M2 nuts and bolts to clamp onto the motor shaft and an octagonal stub that press fits into the center of the wheels. Three long M3 bolts and nylon lock nuts clamp the mecanum wheels together and to the coupler. Each wheel contains eight angled rollers mounted with two ball bearings each to smooth operation under load. Unlike regular wheels which only produce a force vector perpendicular to the axis, mecanum wheels also produce a vector parallel to the axis. With the appropriate combination of speed and direction of each wheel, the robot can achieve simultaneous translation and rotation in any direction.

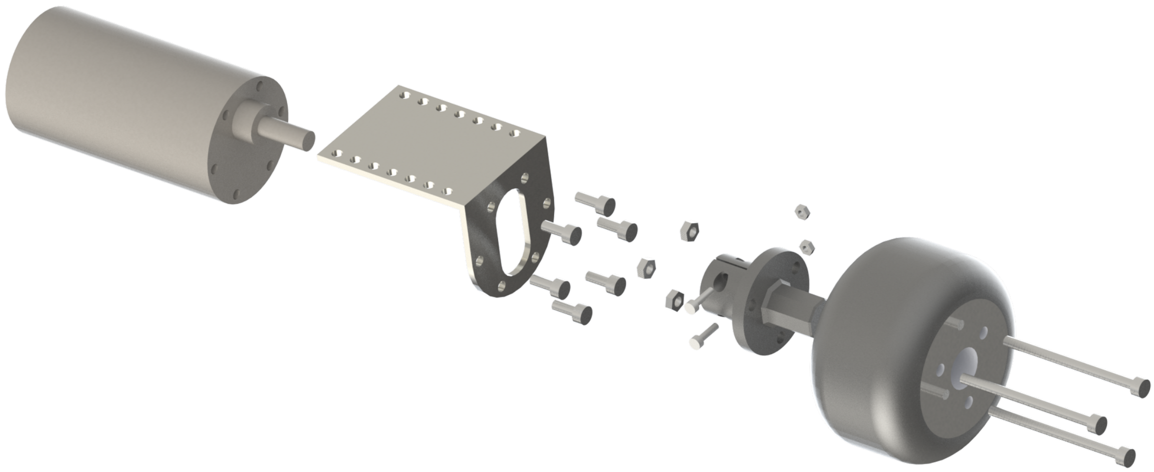


Figure 1.7: Motor Assembly Exploded View

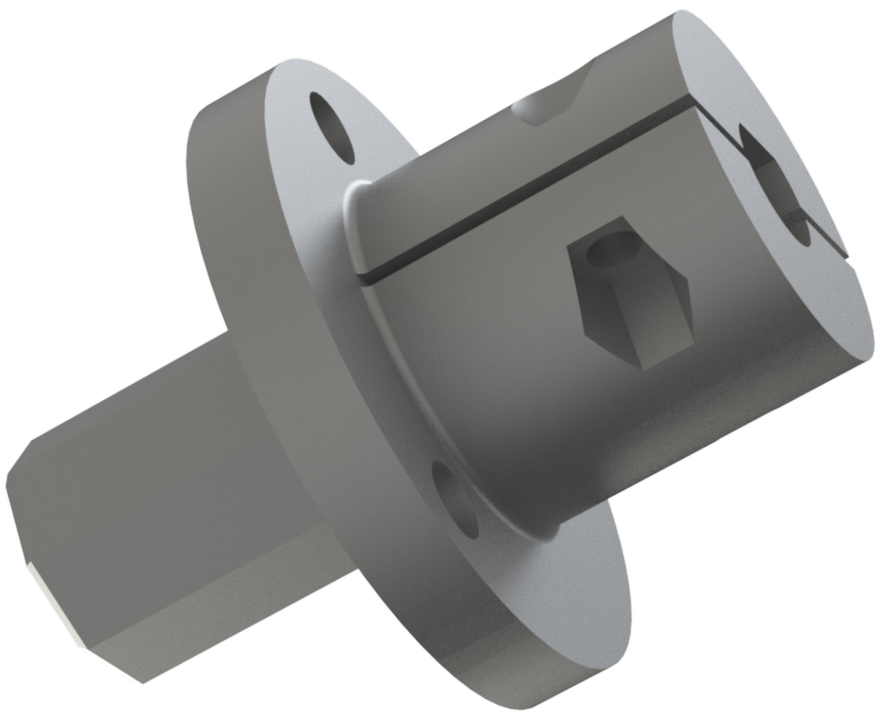


Figure 1.8: Wheel Coupler

Several electronic components are mounted directly on the base platform. Along the four edges of the platform, four STMicroelectronics VL53L0X laser rangefinders seated on 3D printed brackets sense distance. These sensors cost between \$6 to \$20 mounted on a small PCB with supporting circuitry and can sense distances between 30 mm and 2000 mm at a rate of 30 Hz and less than 10% error in most test conditions [16]. A 4S, 1200 mAH LiPo battery mounted with an industrial strength hook-and-loop fastener powers the system through a on/off toggle switch seated in a 3D printed bracket. Three dual H-bridge motor drivers are attached with nylon standoffs and M3 nuts and bolts. Finally, a momentary push button in a 3D printed bracket toggles power to the Raspberry Pi microcomputer.

1.3 Shooting Mechanism

The shooting mechanism naturally takes inspiration from the official Nerf Rival Blaster toys since they're specifically optimized to fire Nerf Rival balls; the system works similarly to a baseball pitching machine. Figure 1.9 shows an exploded view of the subassembly while Figure 1.10 displays the top view. The mechanism consists of two sections: the **barrel** (left green part in Figure 1.9) and the **wheel housing** (right green part in Figure 1.9). Both parts were 3D printed as the geometries are highly complex. Therefore, the shooting mechanism consists of two separate components versus a unibody design to allow each half to be fabricated with optimal print direction, strength, and finish quality. The barrel is angled 6° above horizontal, targeting the vertical center of the nets 6 feet away.

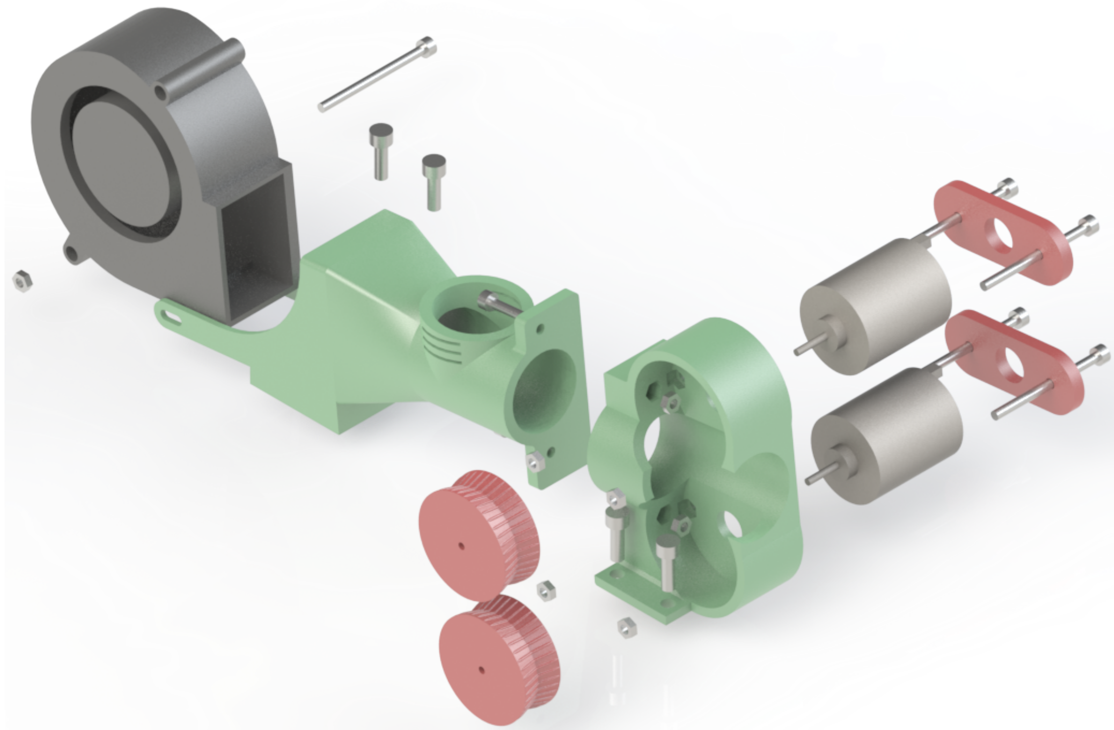


Figure 1.9: Shooting Mechanism – Exploded View

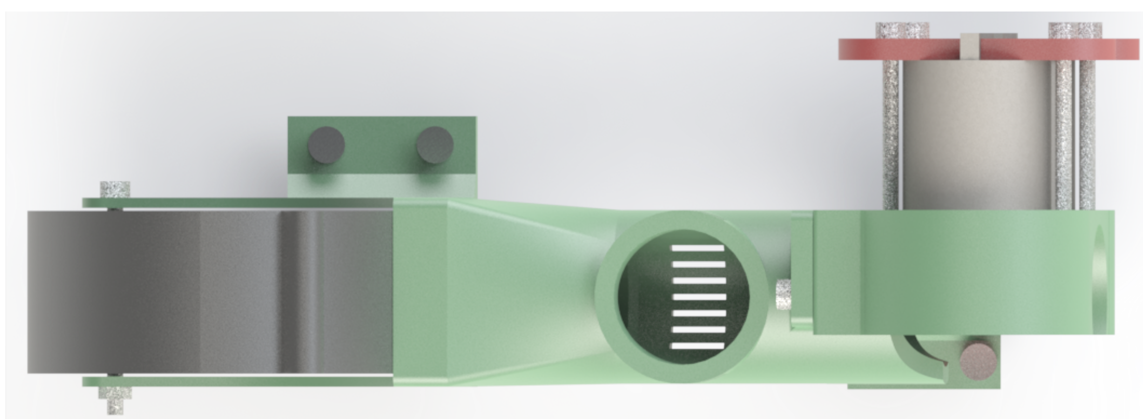


Figure 1.10: Shooting Mechanism – Top View

The **barrel** directs balls from the **ball hopper** to the **wheel housing**. First, the ball enters the barrel through a vertical chute by force of gravity. As the ball falls into the barrel, a high-pressure centrifugal (or blower fan) attached at the back of the barrel pushes it into the wheel housing inlet. As seen in Figure 1.11, the barrel slightly narrows in the area behind the top chute to prevent the ball from rolling backwards towards the blower fan. A loft feature creates a smooth transition between the rectangular fan inlet and the circular barrel. The foam balls, nominally 23 mm in diameter, would occasionally jam in a 24 mm inner diameter barrel due to ball surface imperfections so the barrel was increased to 25 mm inner diameter. In the initial design, the pressure created by the blower fan was so high that it prevented the ball from falling down the vertical chute. The revised barrel uses strategically placed vents reduce the barrel pressure before the ball enters the chute. As the ball travels down the chute into the barrel, it blocks the vents, increasing the pressure and forcing the itself into the wheel housing.

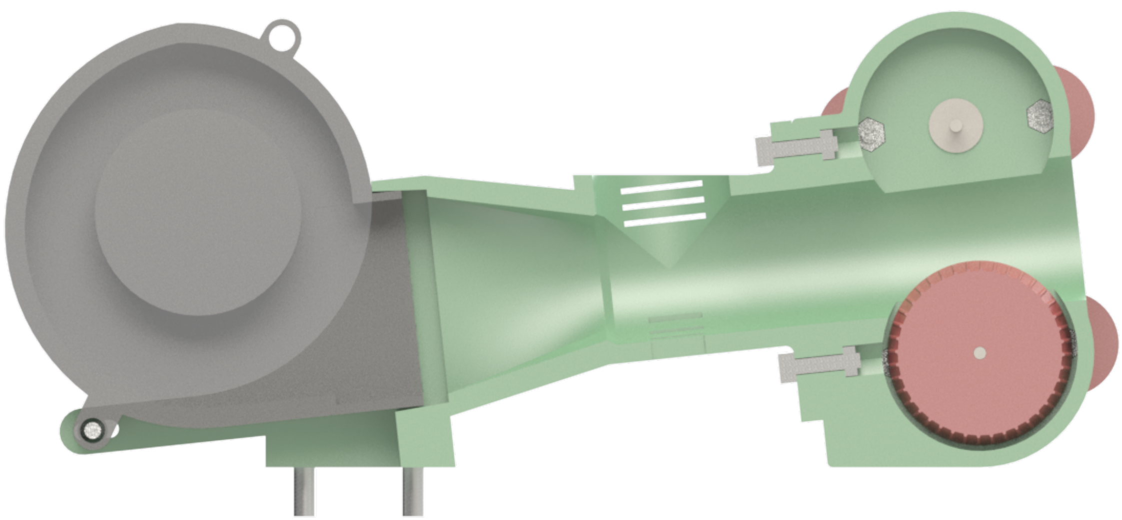


Figure 1.11: Shooting Mechanism – Cross Section View

Inside the **wheel housing**, two counter-rotating 34mm wheels press fitted to two high-speed 12 V motors rapidly accelerate the foam ball up to 50 feet per second. The 14 mm gap between wheels compresses the ball to increase grip, thereby improving energy transfer. The motors lightly press fit into the wheel housing and are secured with 3D printed braces. The perimeter of each 3D printed wheel, detailed in Figure 1.12, consists of a ribbed V-groove to increase the contact patch and grip with the compressed foam ball. Two "feet" with bolt holes at the bottom of the barrel and wheel housing secure the shooting mechanism to the base platform.

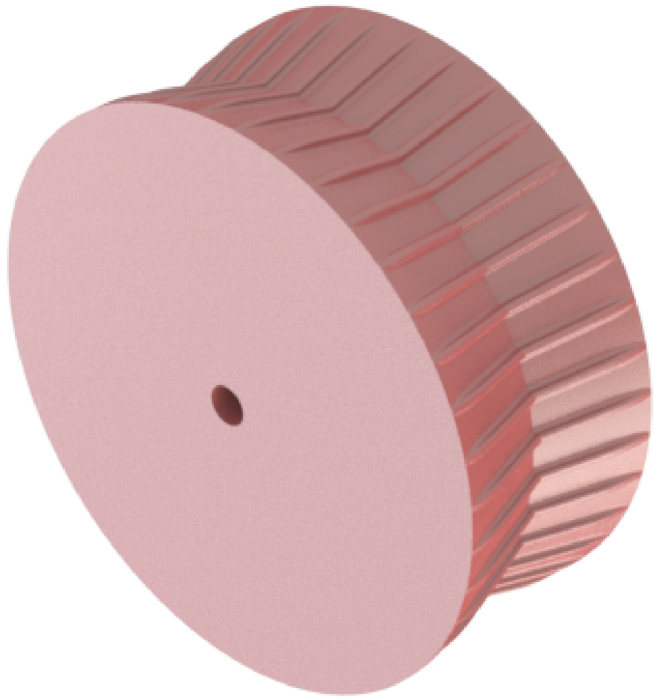


Figure 1.12: Shooting Mechanism – Shooter Wheel

The shooting mechanism performed consistently; in a trial of 100 launches, 100% of balls passed through a 6 inch diameter ring placed 6 feet away where the center of the competition net would be. Since the actual nets are 24 inches in diameter, no further testing was required. The projectile speed averaged 48.1 ft/s with 1.2 ft/s standard deviation as determined with a light gate speed measurement tool after tuning motor speed in the microcontroller.

1.4 Ball Hopper

During the competition, the robot must obtain the foam balls from supply tubes mounted on two sides of the side. The supply tubes consist of two 1" inner diameter

PVC tee joints and an eyebolt. The bottoms of the supply tubes are positioned seven inches above the floor and a swinging flap holds the balls in as shown in Figure 1.13. The ball hopper, shown in Figure 1.14 is a large 3D printed component designed to push the swinging flap away, collect the balls, store them, and dispense them into the shooting mechanism. Figure 1.15 shows an exploded view of the subassembly.

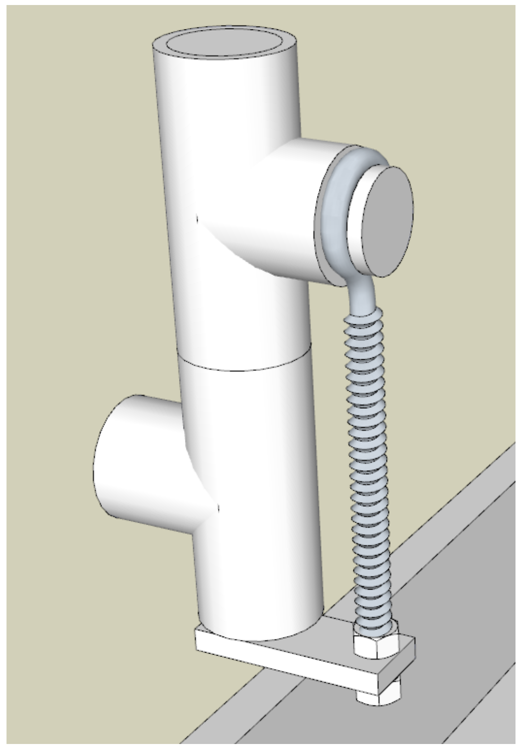


Figure 1.13: Supply Tube

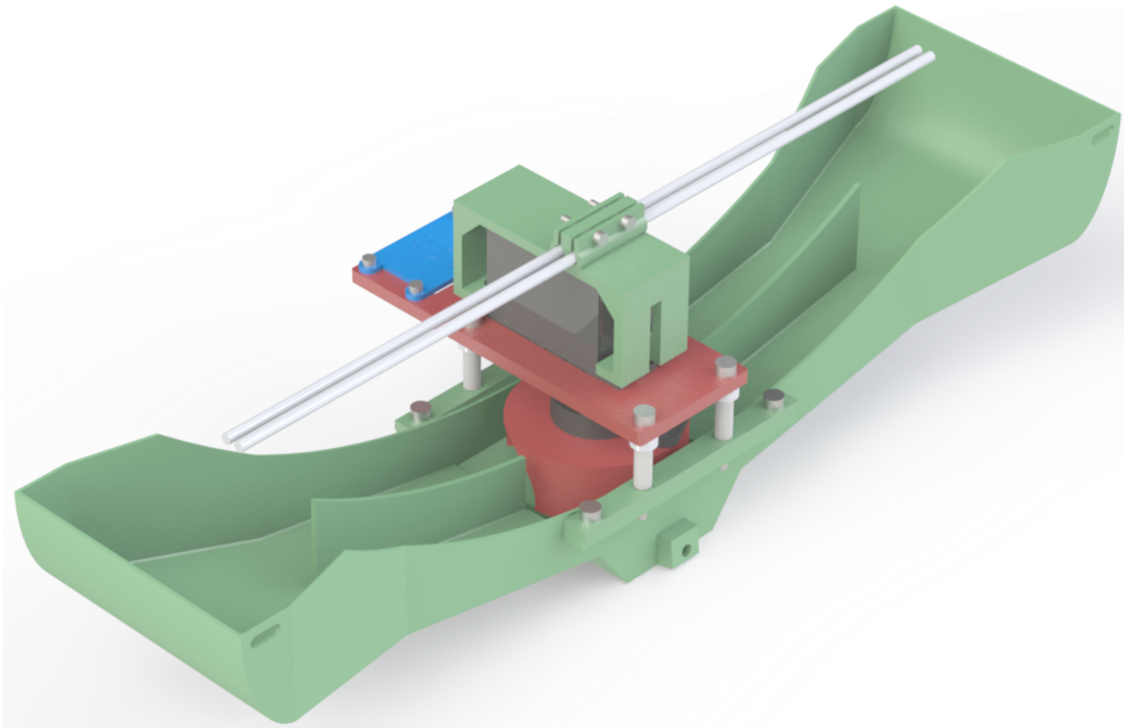


Figure 1.14: Ball Hopper

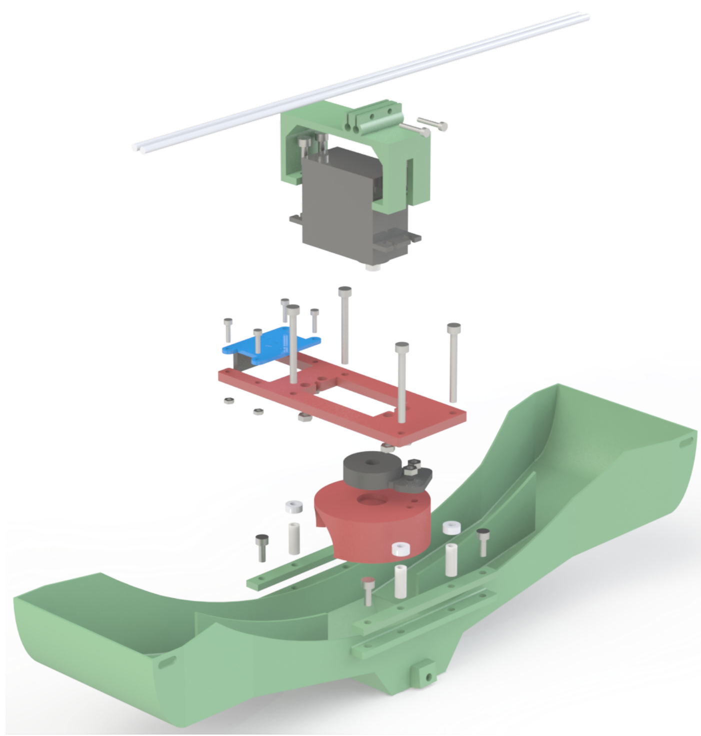


Figure 1.15: Ball Hopper – Exploded View

To obtain balls, the robot first positions itself such that the wide portion of hopper resides next to the supply tube. The robot then moves such that the metal rods at the top of the ball hopper push the eyebolt aside. As the flap swings open, the balls roll out of the supply tube and down the steep sloped portion of the hopper. Visible in the cross section view of Figure 1.16, the slope rapidly becomes shallower in order to convert the balls' downward momentum into sideways momentum which keeps balls from jamming against each other. The balls then roll into one of two channels before stopping at the dispensing gate.

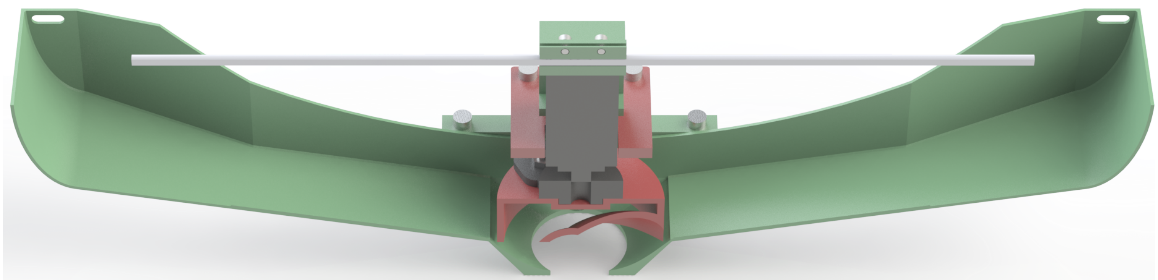


Figure 1.16: Ball Hopper – Cross Section View

The dispensing gate, shown in Figure 1.17, controls the movement of balls between hopper channels and the shooting mechanism entrance. Its complex shape directs balls into the hole at the bottom of the ball hopper from one channel at a time to prevent jamming. A standard 180° rotation servo, mounted in a 3D printed bracket above the center of the hopper, controls the dispensing gate. Fastened to the same bracket, an inertial measurement unit (IMU) measures magnetic compass heading and acceleration in three dimensions. The IMU is positioned at the rotational center of the robot to prevent the accelerometer from measuring robot rotation as linear movement. The ball hopper is mounted at three points: the top of the shooting mechanism and the left and right edges of the robot using 3D printed and acrylic braces shown in Figure 1.18.

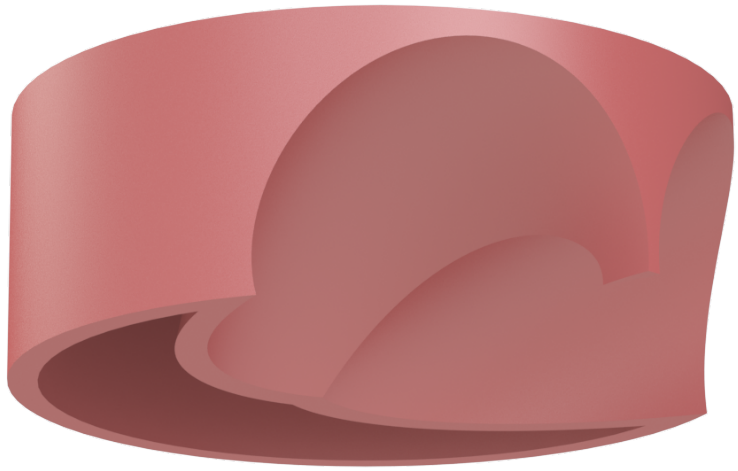


Figure 1.17: Ball Hopper – Dispensing Gate

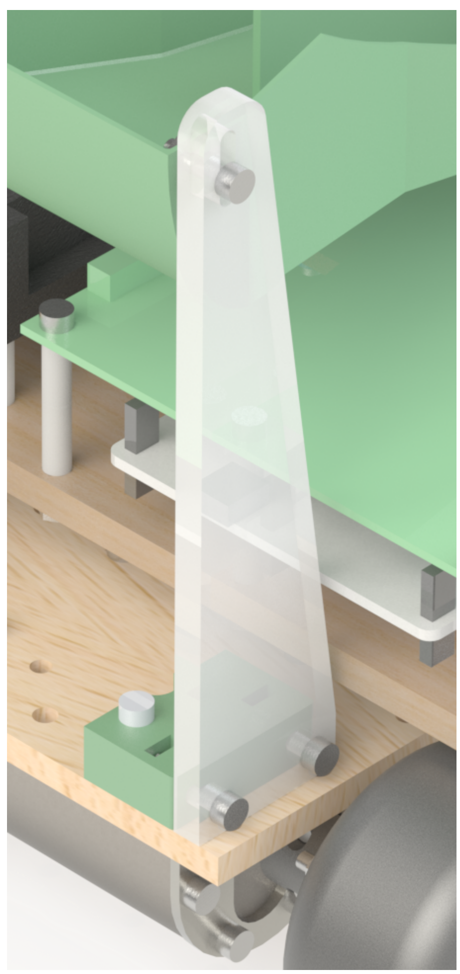


Figure 1.18: Ball Hopper – Braces

1.5 Control Unit

The control unit, shown in Figure 1.19, consists of a 1/4" MDF board with various electronic components mounted: two off-the-shelf DC-DC switching converters, a custom interconnect printed circuit board (PCB), an off-the-shelf STM32 Nucleo-64 development board, and the Raspberry Pi microcomputer. Two 3D printed standoffs, the green parts shown in Figure 1.20, connect the control unit to the platform and raise it slightly to avoid colliding with the robot's wheels.

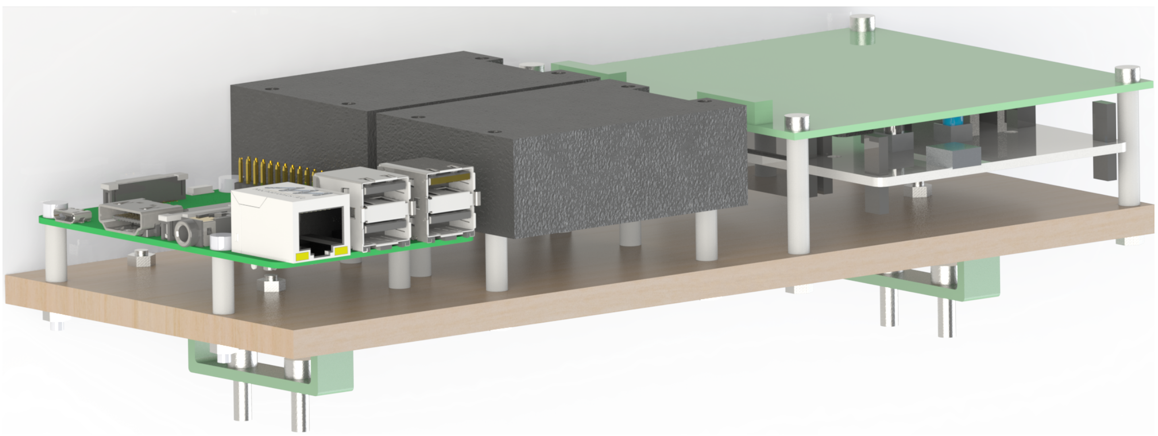


Figure 1.19: Control Unit

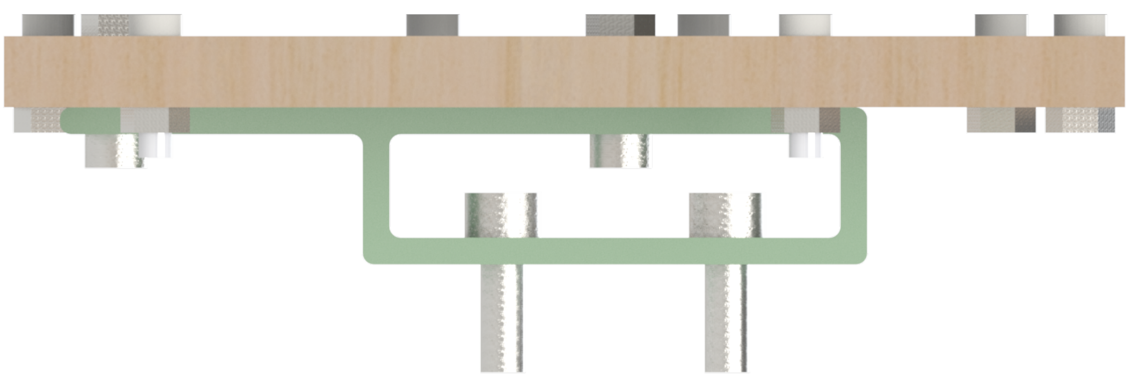


Figure 1.20: Control Unit – Standoffs

Chapter 2

ELECTRICAL DESIGN

2.1 Introduction

The electronics of the robot use a combination of off-the-shelf parts and custom designed circuits. A microcontroller handles low-level hardware control and interfacing such as reading sensors and supplying motor driver control signals while a Raspberry Pi microcontroller processes the data within the artificial neural network and determines motor speeds and directions. The two processors communicate through a 921,600 baud UART link. During development, the robot used either a 2015 MacBook Pro or desktop computer instead of the Raspberry Pi to simplify programming and omit limitations imposed by the Pi's lower processing speed.

2.2 Power

A four cell, 1800 mAH lithium polymer (LiPo) battery powers the entire system. The battery is connected using polarized XT60 connectors to prevent reverse connection. The battery voltage varies between 16.8 V when fully charged and 14.8 V when depleted so two off-the-shelf DC-DC switching regulators, shown in Figure 2.1, buck battery voltage down to 12 V and 7 V supplies. The switching regulators accept a 7 – 40 V supply and can output 1.2 – 35 V at 8 A each. The 12 V bus powers the three motor drivers boards while the 7 V bus powers the STM32 Nucleo-64 development board and two AZ1085CD low-dropout linear regulators (LDO). One LDO produces a 5 V bus while the other provides 3.3 V, each at 3 A.

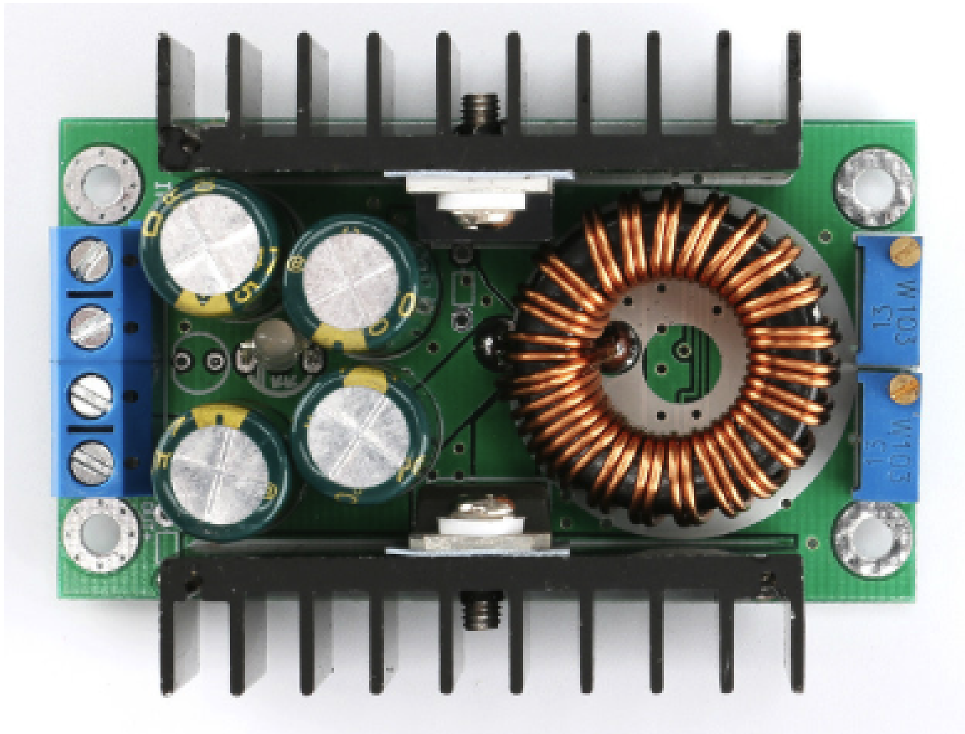


Figure 2.1: DC-DC Buck Regulator [3]

2.3 Sensors

The robot uses five off-the-shelf sensors for determining its position: four VL53L0X 1-D LIDAR rangefinders and one Adafruit 9-DOF inertial measurement unit (IMU).

2.3.1 Adafruit 9-DOF IMU

The Adafruit 9-DOF IMU incorporates the L3DG20H gyroscope and LSM303DLHC accelerometer/compass combo on a single carrier board to allow full inertial measurement in a convenient form factor [2]. Figure 2.2 shows the IMU mounted in the 3D printed bracket. The robot only utilizes the accelerometer and magnetic compass to realize a tilt-compensated compass. The LSM303DLHC can measure both acceleration and magnetic fields in three dimensions with configurable bandwidth and full-scale ranges. It uses 400 kHz I^2C for control and data transfer and draws power from the 3.3 V supply. Raw measurements from the IMU possess significant offset and scaling error so a calibration routine is required.

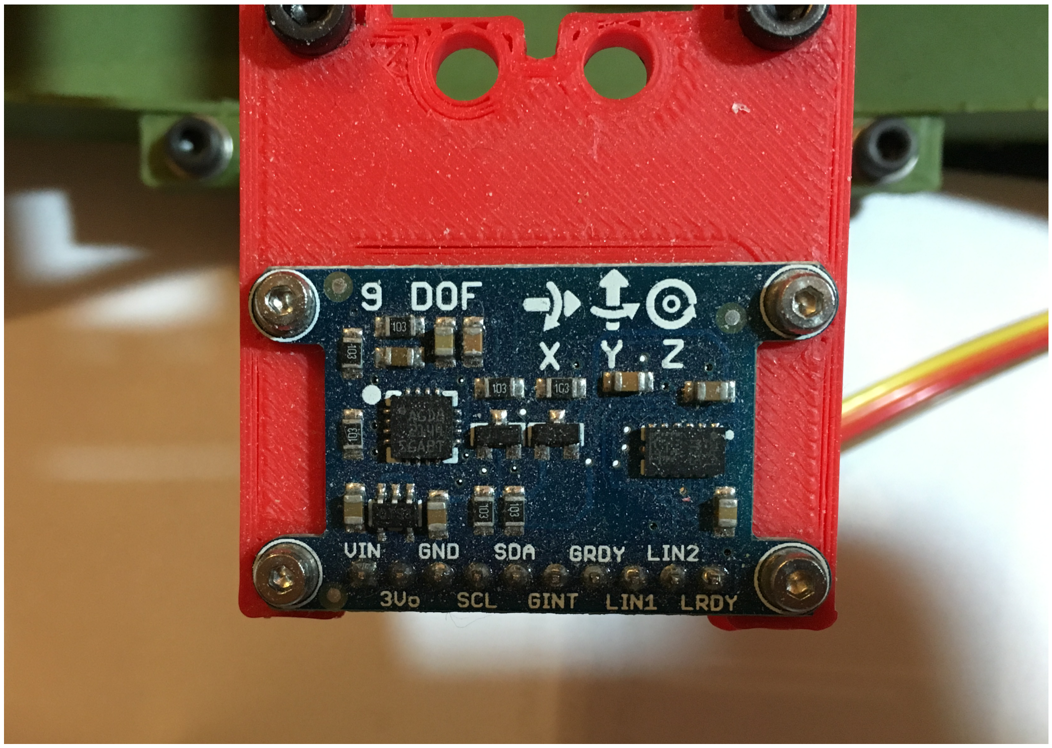


Figure 2.2: Adafruit 9-DOF IMU Mounted

The accelerometer and magnetometer calibration process utilizes the FreeIMU program written by Fabio Varesano in Python [5]. The program features a graphical user interface (GUI), shown in Figure 2.3, to display accelerometer and magnetometer measurements in real time and plots them in 3D space. The calibration program was originally designed to calibrate the open-source FreeIMU IMU when connected to an Arduino with the FreeIMU calibration firmware so the program's device communication back-end was modified to accept the LSM303DLHC IMU connected to an STM32 microcontroller with custom firmware. The calibration algorithm assumes that the sensor's measurements are linearly distorted and therefore produces a linear scaling factor and offset for each of six measurements (accelerometer X, Y, Z and magnetometer X, Y, Z). The correction algorithm is shown below where *val* can represent any of the six measurements.

$$val_{calibrated} = m_{scale} val_{measurement} + b_{offset} \tag{2.1}$$

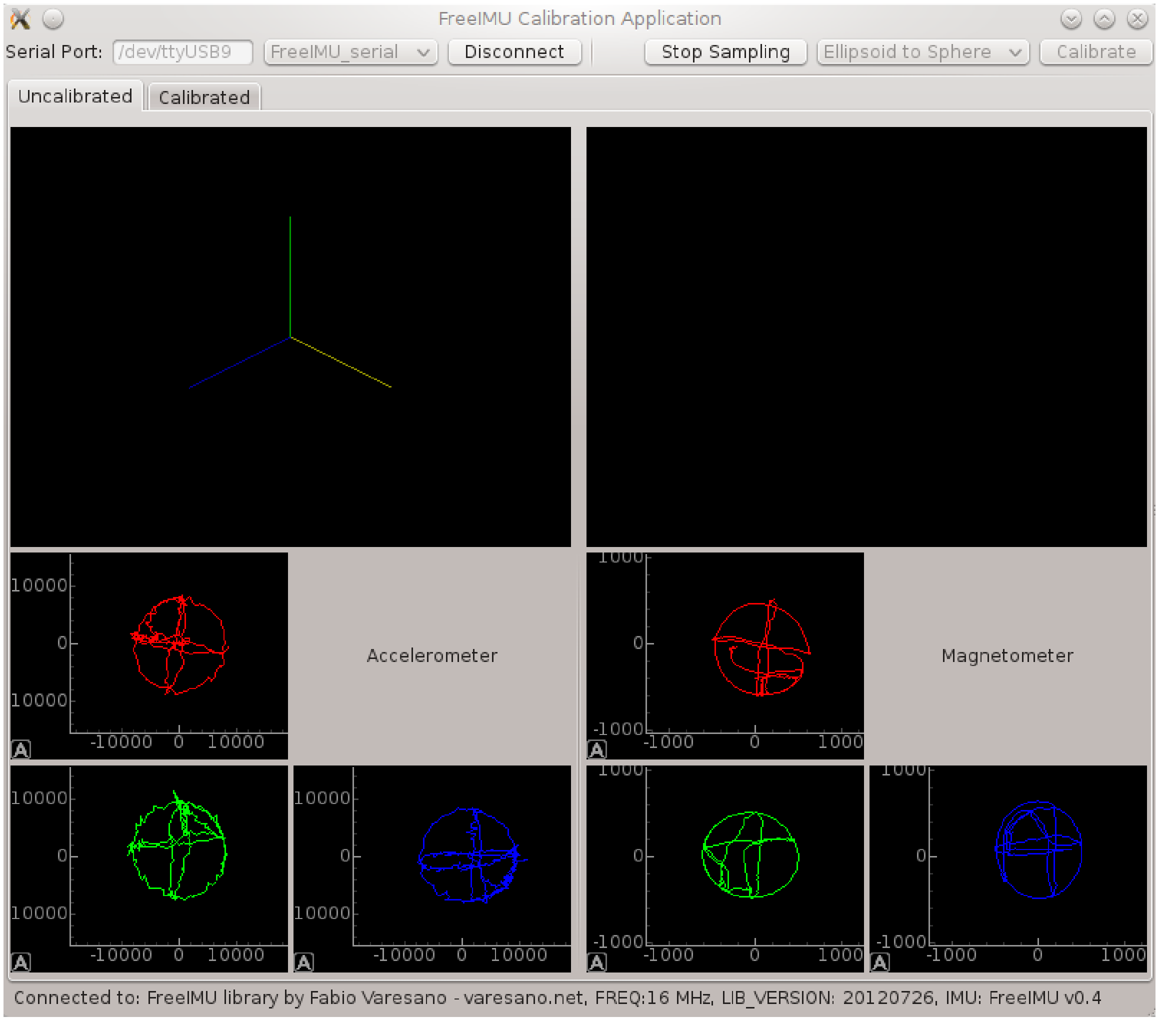


Figure 2.3: FreeIMU GUI [5]

The calibration procedure is as follows:

1. IMU should be mounted on fully assembled robot and connected to microcontroller.
2. Connect microcontroller serial port to computer through Serial-to-USB converter.

3. Click "Begin Sampling" to start recording magnetometer and accelerometer measurements.

4. Point the IMU x-axis at the ground and rotate 360° around that axis. Repeat for y-axis and z-axis.

5. Point the IMU x-axis at the sky and rotate 360° around that axis. Repeat for y-axis and z-axis.

6. Repeat Steps 3 and 4 at least twice to increase data size.

7. Click "Stop Sampling".

8. Click "Calibrate" to calculate scaling and offset constants.

The procedure ensures that ends of each axis eventually receive the maximum acceleration and magnetic field. To ensure hard-iron errors (such as from motors and permanent magnets) as well as soft iron errors (from local ferromagnetic materials like steel) are compensated for in the calibration, the process should be performed with the fully assembled robot and redone each time the robot is modified [11]. Since the expected fields are known (acceleration due to gravity, strength of Earth's magnetic field), the required linear transformation can be calculated. See [5] for details of the exact algorithm.

2.3.2 VL53L0X Rangefinders

The rangefinders, marketed by STMicroelectronics as the "world's smallest Time-of-Flight ranging sensor", are capable of measuring between 30 and 2000 mm with a 30 Hz sample rate [15]. It operates by firing pulsed light from a vertical cavity surface emitting laser (VCSEL), measuring time taken for the laser pulse to reflect back to the sensor, and calculating the distance based on the known speed of light. The robot uses cheap $8 VL53L0X breakout boards in lieu of designing and assembling custom carriers; the sensor itself comes in a 4.4 x 2.4 mm lead-less package making hand soldering prohibitively difficult. Figure 2.4 shows the sensor board mounted in the 3D printed bracket. The sensor breakout boards include the VL53L0X module, decoupling capacitors, an LDO for the sensor's 2.8 V power supply, and level shifters for the I^2C lines. Control and data transfer both occur over 400 kHz I^2C so the breakout only requires four wires: 3.3 V, ground, and the I^2C data and clock lines. To characterize the accuracy and precision of the sensor, distance measurements were taken by targeting the sensor at a sheet of standard white printer paper. The data is compared with measurements from a tape measure; results are shown in Figure 2.5.

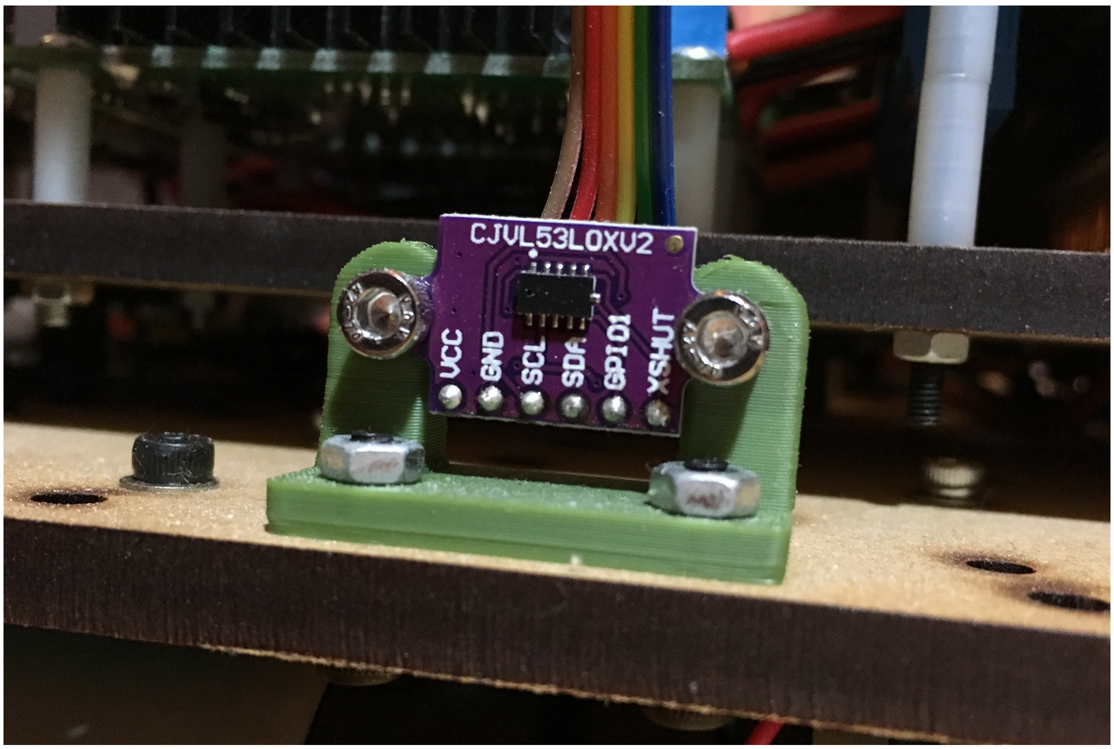


Figure 2.4: VL53L0X Rangefinder Mounted

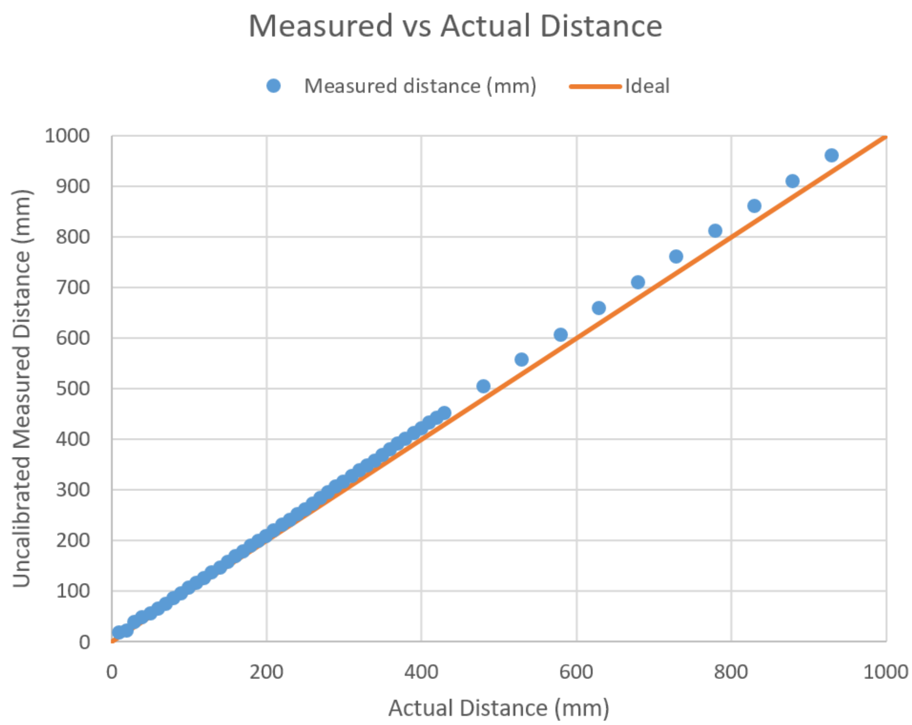


Figure 2.5: VL53L0X Measured vs. Actual Distance

Since the measurements are linear, only a linear correction is required as follows:

$$d_{calibrated} = m_{scale} d_{measurement} + b_{offset} \tag{2.2}$$

where $d_{measurement}$ is distance as returned by the sensor, $m_{scale} = 0.96502507$, $b_{offset} = -3.8534743$, and $d_{calibrated}$ is the calibrated distance. The squared error for each data point before and after calibration is shown in Figure 2.6. The mean squared error for the uncalibrated data points is 342.3 mm and 15.0 mm after calibration, indicating the use of an appropriate model and constants.

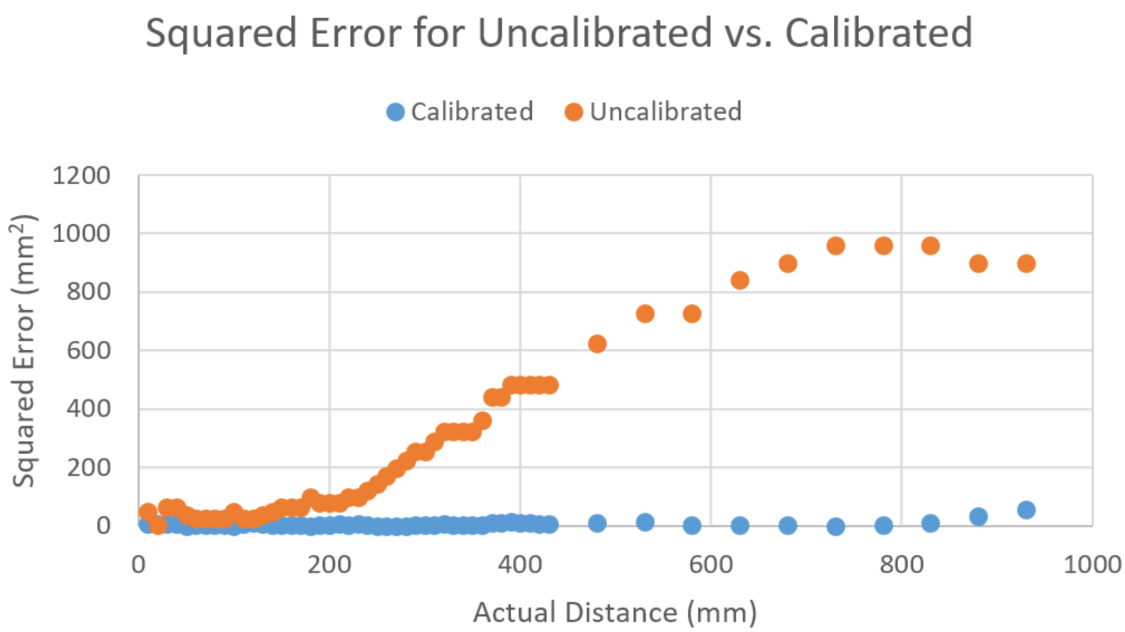


Figure 2.6: VL53L0X Squared Error for Uncalibrated vs. Calibrated

Additionally, 100 measurements were taken at each distance and the variance computed to characterize the spread. The results are shown in 2.7. The rangefinders are very accurate; at 900 mm with calibration, the error is only 7 mm or 0.8% error. Of course, this error is with respect to the average measurement at 900 mm. At 900 mm, the standard deviation is 14 mm so individual measurements can vary, especially with non-ideal surfaces. The closer the target, the more accurate and less varied the measurement. No experiments were carried out to determine the measurement characteristics on various colored, textured, transparent, or angled surfaces as the expected target surface is white painted wood.

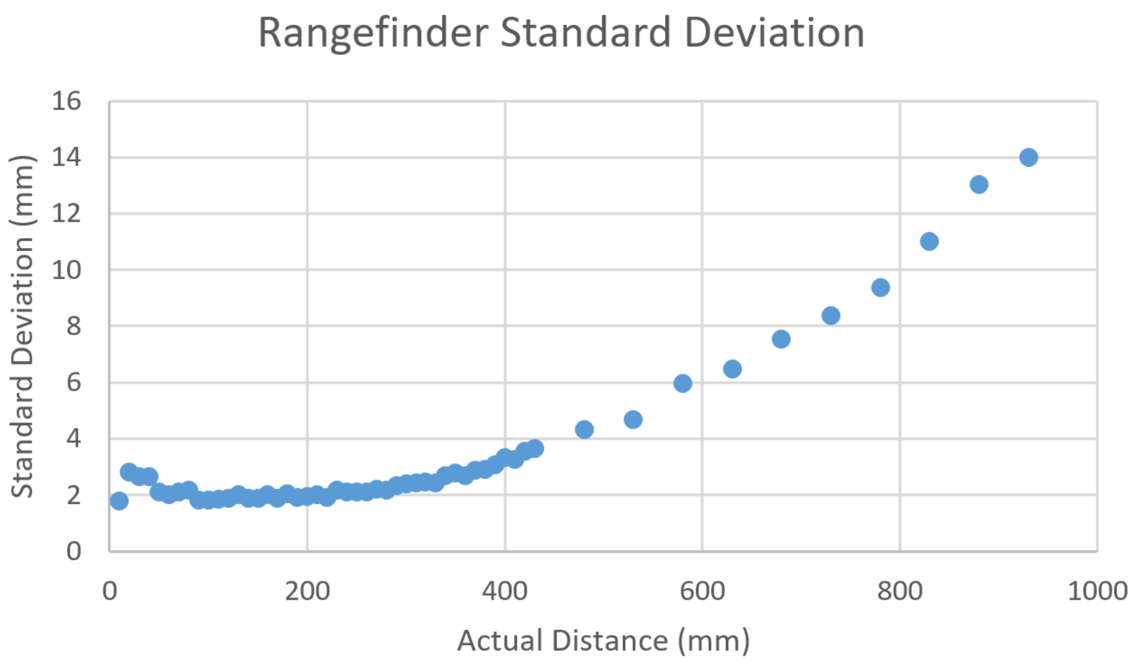


Figure 2.7: VL53L0X Standard Deviation

2.4 Motors

The robot uses four 12V Pololu 37D motors to drive the wheels. The motors draw 630 mA each at full speed steady state of 234 rpm (24.46 rad/s) and have a maximum acceleration of 4,100 rpm/s (430 rad/s^2).

2.5 Motor Drivers

The system uses three off-the-shelf L298N motor driver boards since they are easily obtainable for less than $6 each and incorporate features such as heat-sinking, flyback voltage protection, supply filtering, and screw terminal connections. Implementing comparable motor drivers with a similar feature set would undoubtedly cost more. Each L298N is a dual H-bridge driver with 2 A maximum output per bridge using a 5 – 35 V supply. Two motor drivers handle the four robot drive motors while the third powers the blower fan and shooting mechanism motors.

Figure 2.8 shows a wiring diagram for each motor driver. The board uses four digital control inputs, each controlling the state of one half-bridge. Each motor uses a pair of inputs: IN1 and IN2 control one motor while IN3 and IN4 control the other. To achieve direction and speed control, IN1 and IN3 are pulse width modulated (PWM) while IN2 and IN4 are digitally set. Table 2.1 is a truth table of the motor state versus inputs.

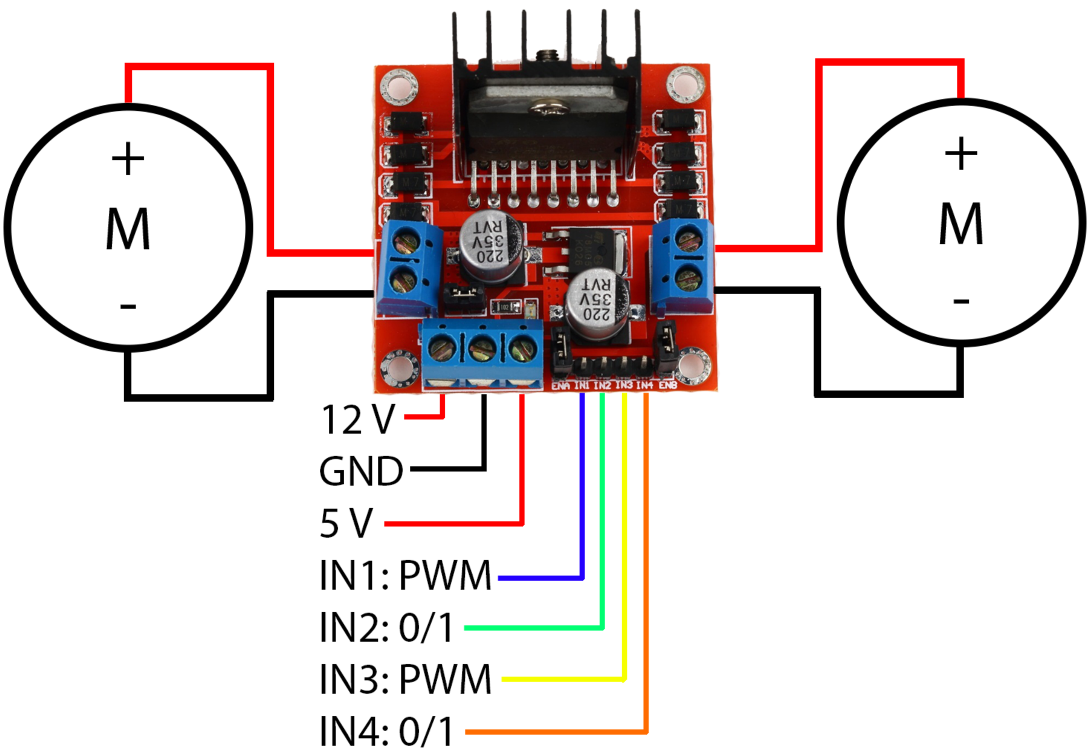


Figure 2.8: L298N Motor Driver Wiring Diagram [9]

2.6 Servo

A GWS S03N standard servo powered from the 5 V bus actuates the gating mechanism in the ball hopper. Most servos are controlled by driving the control wire with a pulse width modulated (PWM) signal with a period of 15 – 25 ms and a pulse width between

Table 2.1: Motor Control Truth Table

IN1/IN3 Duty Cycle	IN2/IN4 State	Motor State
0%	0	Stopped
>0%	0	Forward, speed increases with duty cycle
<100%	1	Reverse, speed decreases with duty cycle
100%	1	Stopped

0.5 ms and 2.5 ms where the pulse width determines the position of the servo as shown in Figure 2.9.

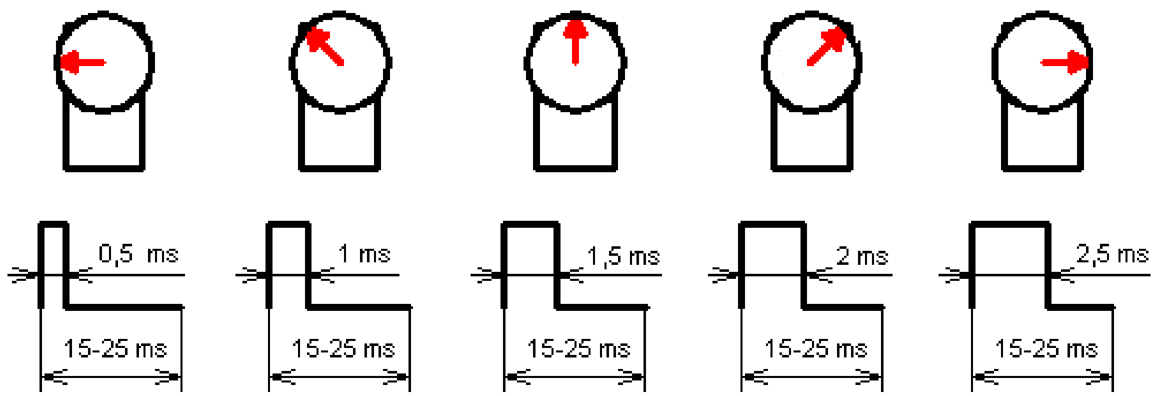


Figure 2.9: Servo PWM Control Scheme [7]

Due to cheap manufacturing and loose tolerances, the actual required pulse widths can vary from servo to servo, requiring calibration to obtain accurate positional control. The process for calibration is simple: apply various pulse widths and record the resulting servo positions. The calibrated pulse widths and servo angles are recorded in Table 2.2.

Table 2.2: Servo Required Pulse Widths

Servo Position	Pulse Width (ms)
0°	0.674
45°	1.082
90°	1.490
135°	1.898
180°	2.306

2.7 Microcontroller

An STMicroelectronics STM32F446RE microcontroller (MCU) serves as the bridge between the robot's low-level electronics and the high-level control system running on a desktop computer. Specifically, the MCU collects data from sensors over I^2C and general purpose input/output (GPIO), generates control signals for the motor drivers and servo, and services commands from UART (universal asynchronous receiver-transmitter). The MCU resides on an STMicroelectronics Nucleo-64 development board, shown in Figure 2.10, which conveniently integrates an ST-LINK V2 debugger, programmer, and USB-to-UART interface.

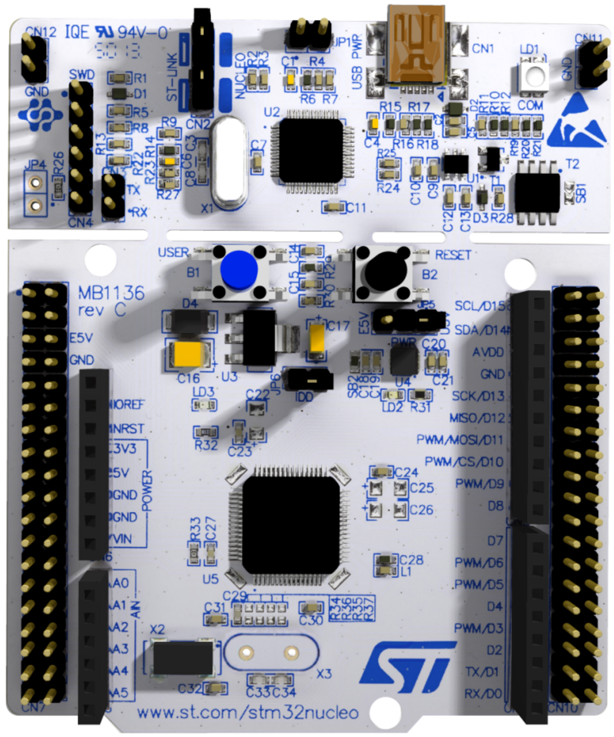


Figure 2.10: STM32 Nucleo-64 Development Board [14]

2.8 Interconnect PCB

The interconnect PCB is a custom designed board with three goals: regulate voltage, route power, and connect peripherals to the development board. The PCB routes input battery power to two external buck regulators, producing 12 V and 7 V. The 7 V bus is further regulated to 5 V and 3.3 V logic rails using two AZ1085CD LDOs and then routed to the development board, sensors, etc. Most importantly, the interconnect board provides dedicated 2.0 mm JST connectors for all sensors, motor drivers, and the servo to simplify and robustify wiring across the robot. In addition, hand-crimped cables made with JST connectors and 28 gauge ribbon cable prevent rats nest wiring. A block diagram is shown in Figure ??. Due to uncertainty in the final robot design, extra hardware and circuits were added. Table 2.3 lists the supported features.

Table 2.3: Interconnect PCB – Supported Features

Quantity	Feature
6	VL53L0X laser rangefinder
1	Adafruit 9-DOF IMU
3	Dual H-bridge motor driver
2	5 V Standard servo
2	8 sensor IR proximity array
4	Debug LEDs
1	Debug button
1	Reset button

The system utilizes two 400 kHz I^2C buses to improve communication bandwidth with the numerous I^2C devices. Table 2.4 lists I^2C device addresses and bus allocation. The two buses are named I2C2 and I2C3 to maintain parity with the microcontroller's naming convention.

The schematic capture and board layout were created in CadSoft EAGLE 7.4 due to its simplicity of use, popularity, and community support. The schematic can be found in Appendix B, layout in Appendix C, and bill of materials in D. The board uses two 1 ounce copper layers for cost effectiveness. Two large headers underneath the board connect to the headers on the top of the microcontroller development board while the JST connectors are placed across the top of the PCB. 18-gauge wires soldered to large plated through-holes at the right edge of the board connect the external buck regulators. All components are placed on the top side of the board (except for the development board connectors) to make hand-soldering easier. The completed electronics assembly is shown in Figure 2.11.

Table 2.4: Interconnect PCB – I^2C Devices

I^2C Bus	Device	Address (7-bit)
I2C2	Rangefinder 4	0x52
I2C2	Rangefinder 5	0x53
I2C2	Rangefinder 6	0x54
I2C2	Adafruit 9-DOF IMU Gyroscope	0x69
I2C2	Adafruit 9-DOF IMU Accelerometer	0x19
I2C2	Adafruit 9-DOF IMU Magnetometer	0x1E
I2C2	GPIO Expander 3	0x3A
I2C3	Rangefinder 4	0x52
I2C3	Rangefinder 5	0x53
I2C3	Rangefinder 6	0x54
I2C3	GPIO Expander 1	0x38
I2C3	GPIO Expander 2	0x39

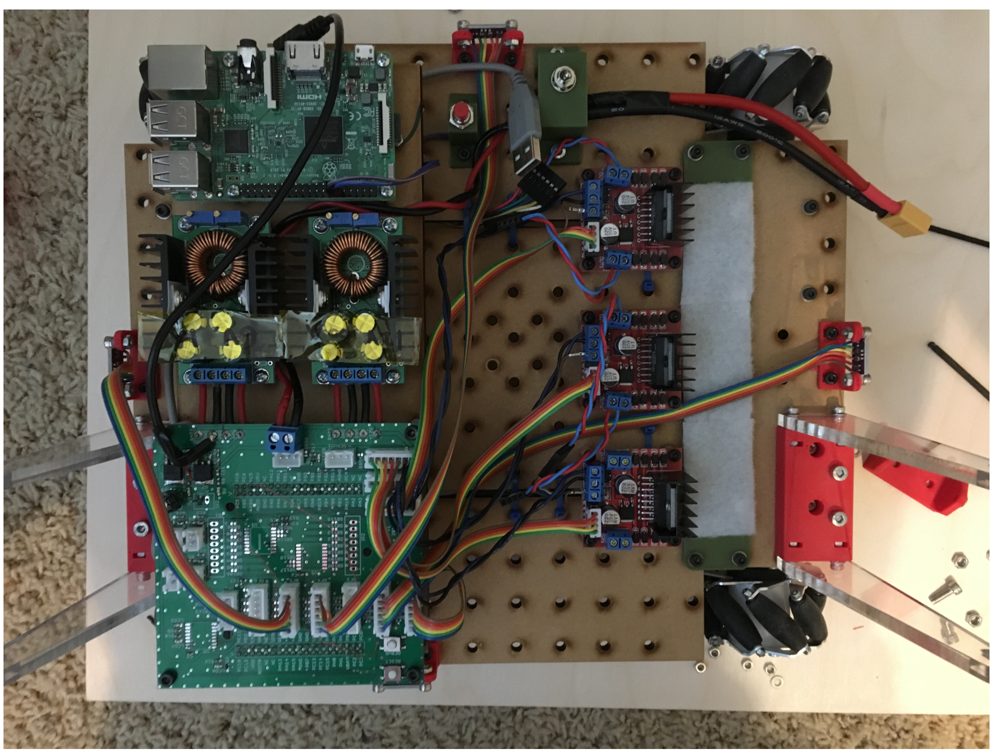


Figure 2.11: Fully Assembled Electronics

BIBLIOGRAPHY

[1] . . , 2018. [Online; accessed May 25, 2018].

[2] Adafruit. Adafruit 9-DOF IMU Breakout. https://www.adafruit.com/product/1714, 2014. [Online; accessed May 15, 2018].

[3] AliExpress. DC-DC CC CV Buck Converter. https://www.aliexpress.com/item/DC-DC-CC-CV-Buck-Converter-Volt-Step-Down-12V-19V-24V-Car-Laptop-Power-Supply/32822603345.html?spm=2114.search0104.3.135.191f73dfqonWDU&ws_ab_test=searchweb0_0,searchweb201602_2_10152_10151_10065_10344_10130_10068_10324_10547_10342_10325_10546_10343_10340_10548_10341_10545_10696_10084_10083_10618_10307_10059_308_100031_10103_10624_10623_10622_10621_10620,searchweb201603_32,ppcSwitch_5&algo_expid=30a33cc9-3acf-4021-b514-6ba550085dd9-19&algo_pvid=30a33cc9-3acf-4021-b514-6ba550085dd9&priceBeautifyAB=0, 2018. [Online; accessed May 11, 2018].

[4] Cal Poly. Roborodentia - Cal Poly, San Luis Obispo. http://www.myurl.com, 2018. [Online; accessed May 11, 2018].

[5] Fabio Varesano. FreeIMU Magnetometer and Accelerometer Calibration GUI. http://www.varesano.net/blog/fabio/freeimu-magnetometer-and-accelerometer-calibration-gui-alpha-version-out, 2012. [Online; accessed May 11, 2018].

[6] R. R. Labbe. Kalman and Bayesian Filters in Python, Sep 2017.

[7] Leopoldo Armesto. How to generate a PPM signal with Arduino to control a

servo? http://robotica.webs.upv.es/en/how-to-generate-a-ppm-signal-with-arduino-to-control-a-servo/, 2015. [Online; accessed May 14, 2018].

[8] MakerGear. MakerGear M2 3D Printer. https://www.makergear.com/products/m2, 2018. [Online; accessed May 25, 2018].

[9] MechaMan. Working with L298N DC Motor Driver. http://fritzing.org/projects/working-with-l298n-dc-motor-driver, 2018. [Online; accessed May 13, 2018].

[10] R. Negenborn. *Robot Localization and Kalman Filters*. PhD thesis, Sep 2003.

[11] Nevada Mark. FAQ: Hard & Soft Iron Correction for Magnetometer Measurements. https://ez.analog.com/docs/DOC-2544, 2014. [Online; accessed May 11, 2018].

[12] D. H. Nguyen and B. Widrow. Neural Networks for Self-Learning Control Systems. *IEEE Control Systems Magazine*, Apr 1990.

[13] STMicroelectronics. LSM303DLHC Datasheet. http://www.st.com/resource/en/datasheet/DM00027543.pdf, 2013. [Online; accessed May 15, 2018].

[14] STMicroelectronics. STM32 Nucleo-64 User Manual. http://www.st.com/content/ccc/resource/technical/document/user_manual/98/2e/fa/4b/e0/82/43/b7/DM00105823.pdf/files/DM00105823.pdf/jcr:content/translations/en.DM00105823.pdf, 2017. [Online; accessed May 11, 2018].

[15] STMicroelectronics. VL53L0X.
http://www.st.com/en/imaging-and-photonics-solutions/vl53l0x.html, 2018. [Online; accessed May 15, 2018].

[16] STMicroelectronics. VL53L0X Datasheet.
http://www.st.com/content/ccc/resource/technical/document/datasheet/group3/b2/1e/33/77/c6/92/47/6b/DM00279086/files/DM00279086.pdf/jcr:content/translations/en.DM00279086.pdf, 2018. [Online; accessed May 12, 2018].

APPENDICES

Appendix A

MECHANICAL BILL OF MATERIALS

Table A.1: Mechanical Bill of Materials

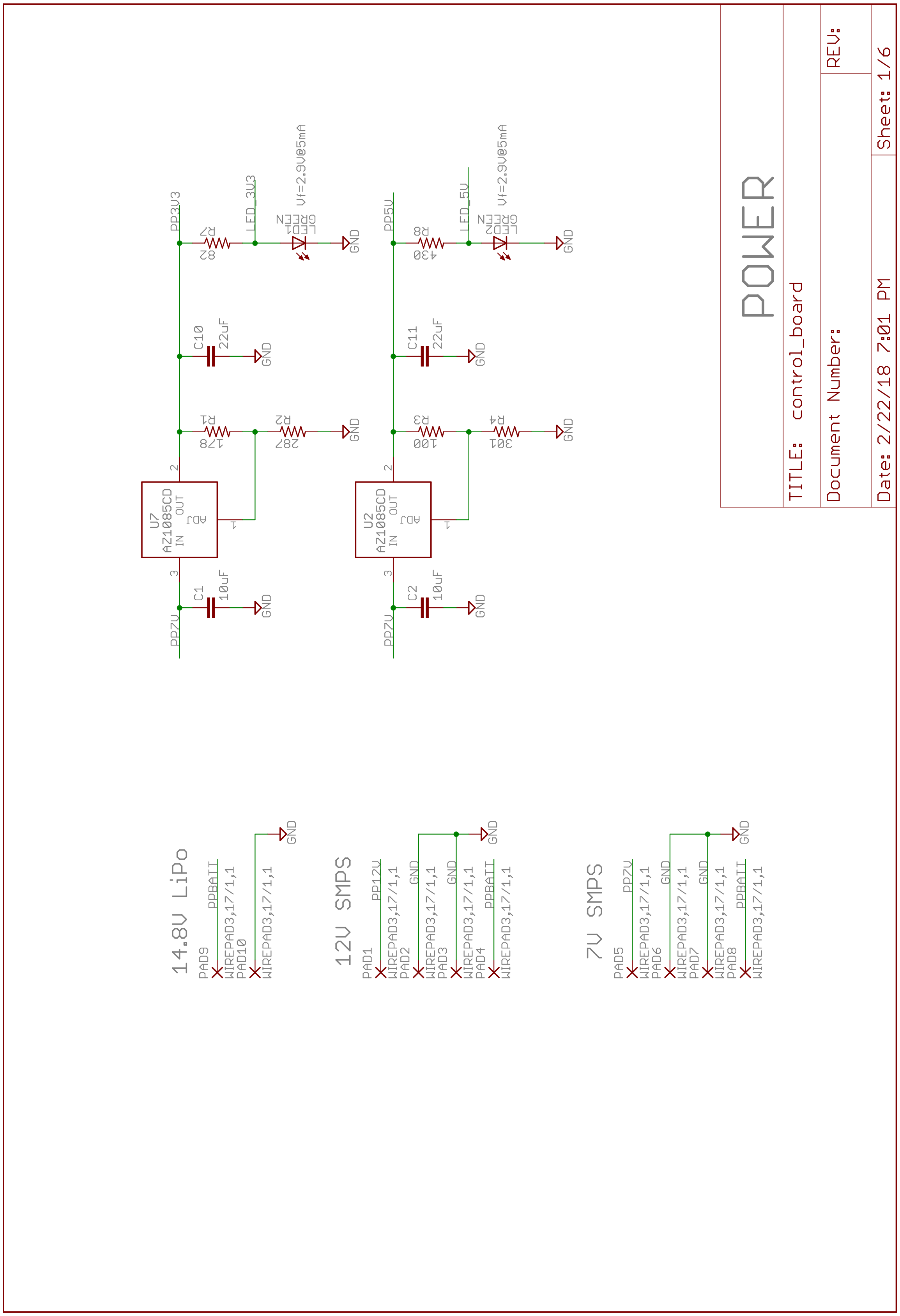
ITEM NO.	PART FILE NAME	QTY.
1	battery	1
2	battery_mount	1
3	blower	1
4	bottom_plate	1
5	buck_converter	2
6	button	1
7	button_mount	1
8	control_brd	1
9	dual_flywheel_cage_fixed	1
10	electrical_mount	1
11	electrical_mount_bracket_angled	2
12	hopper_bar	2
13	hopper_connector	2
14	imu	1
15	launcher_motor_small	2
16	m2-12	2
17	m2-8	12
18	m2-nut	14

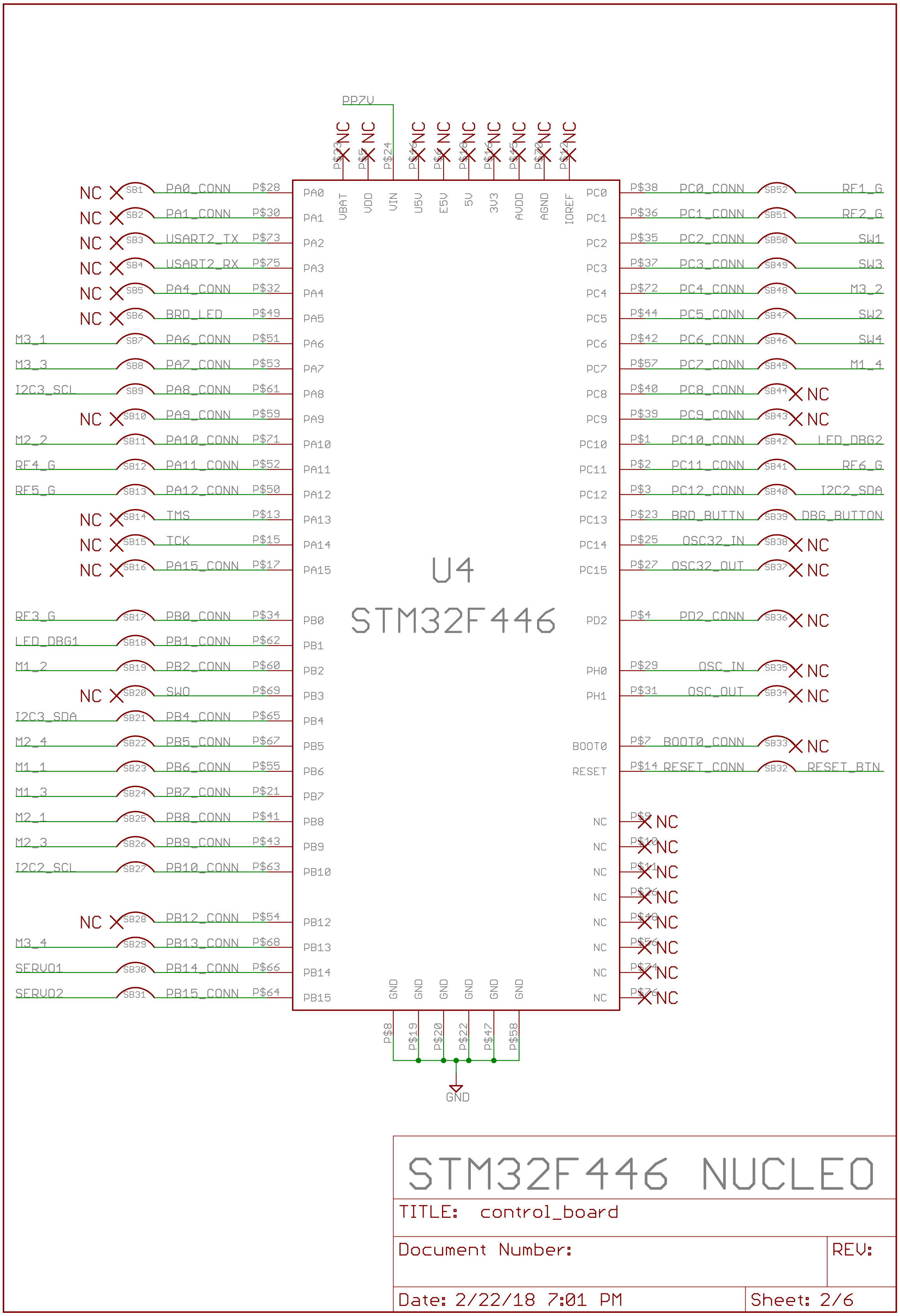
19	m3-12	86
20	m3-20	2
21	m3-25	4
22	m3-30	16
23	m3-35	4
24	m3-40	5
25	m3-45	12
26	m3-nut	115
27	m3-standoff-5	4
28	m4-12	8
29	m4-nut	8
30	mecanum_coupler	4
31	mecanum_wheel	4
32	motor_clamp	2
33	motor_driver	3
34	NUCLEO_F446RE	1
35	nylon_standoff	36
36	pusher_mount	1
37	rangefinder	4
38	rangefinder_mount	4
39	Raspberry Pi 3 Light Version	1
40	servo	1
41	servo_hopper	1
42	servo_hopper_gate	1
43	servo_hopper_mount_base_mini	1
44	servo_hopper_mount_base_right_mini	1

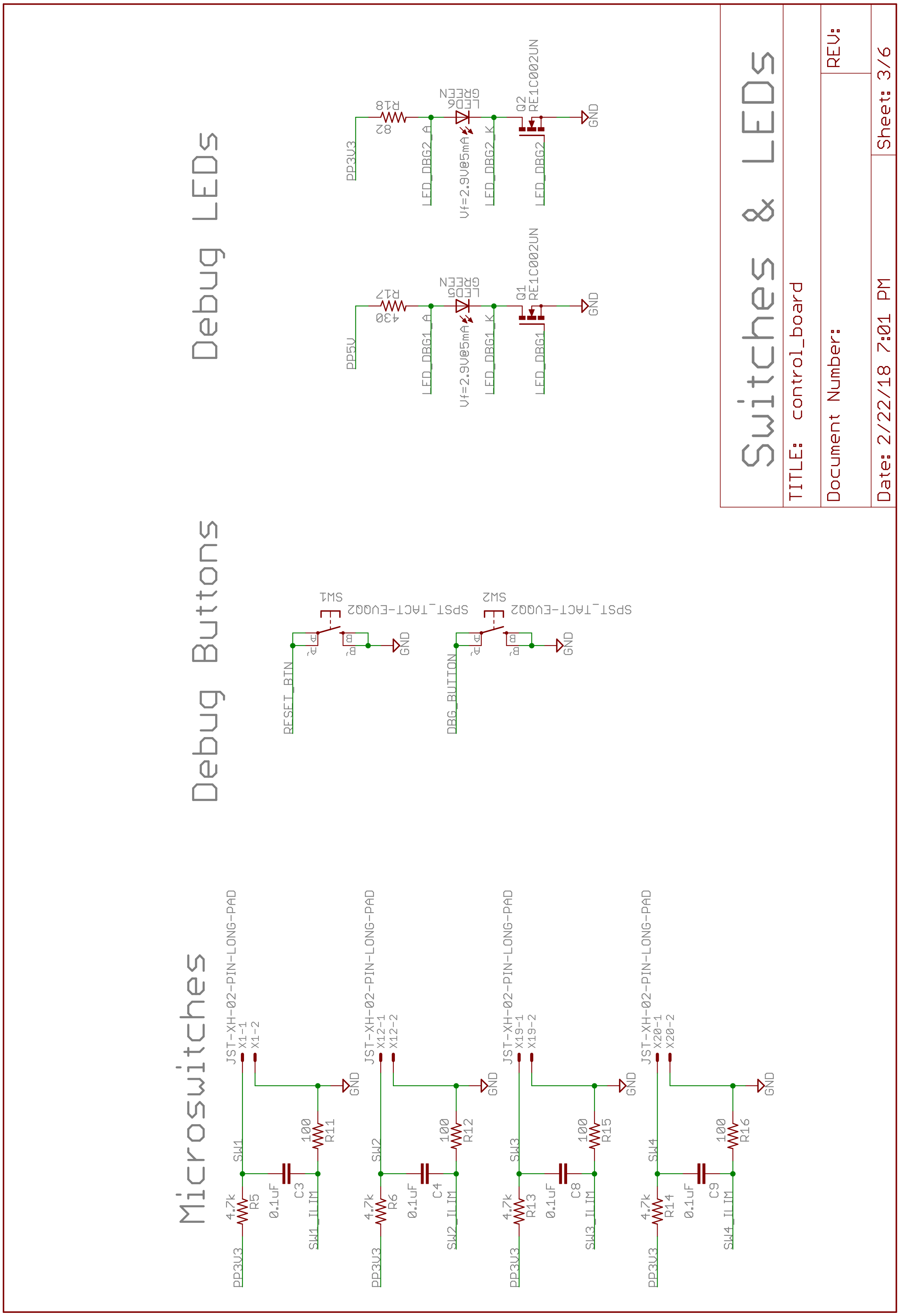
45	servo_hopper_mount_bracket	2
46	servo_hopper_servo_mount	1
47	servo_horn	1
48	switch	1
49	switch_mount	1
50	venturi_loader_fixed	1
51	wheel_grip_small	2
52	wheel_motor	4
53	wheel_motor_bracket	4
	TOTAL	394

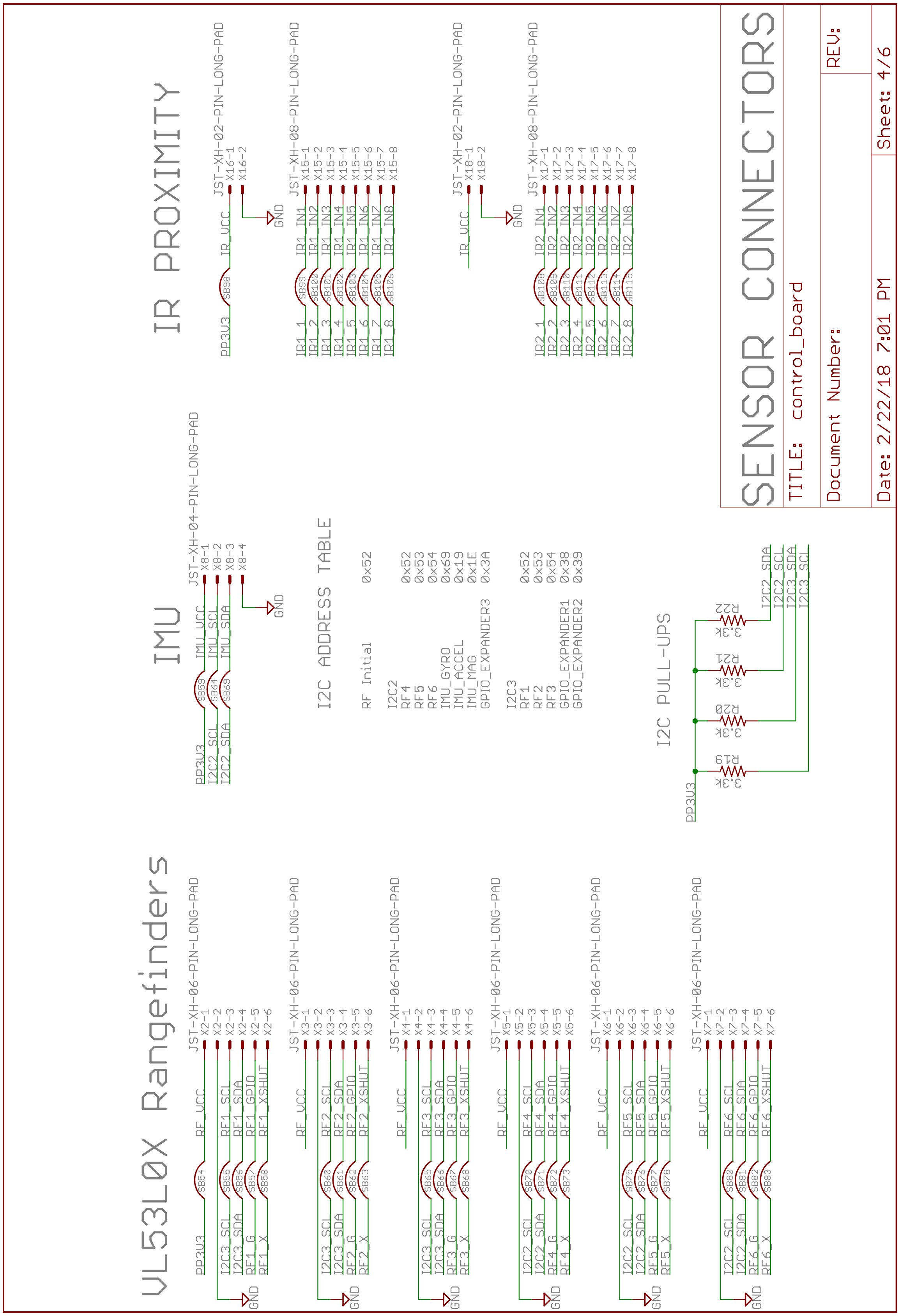
Appendix B

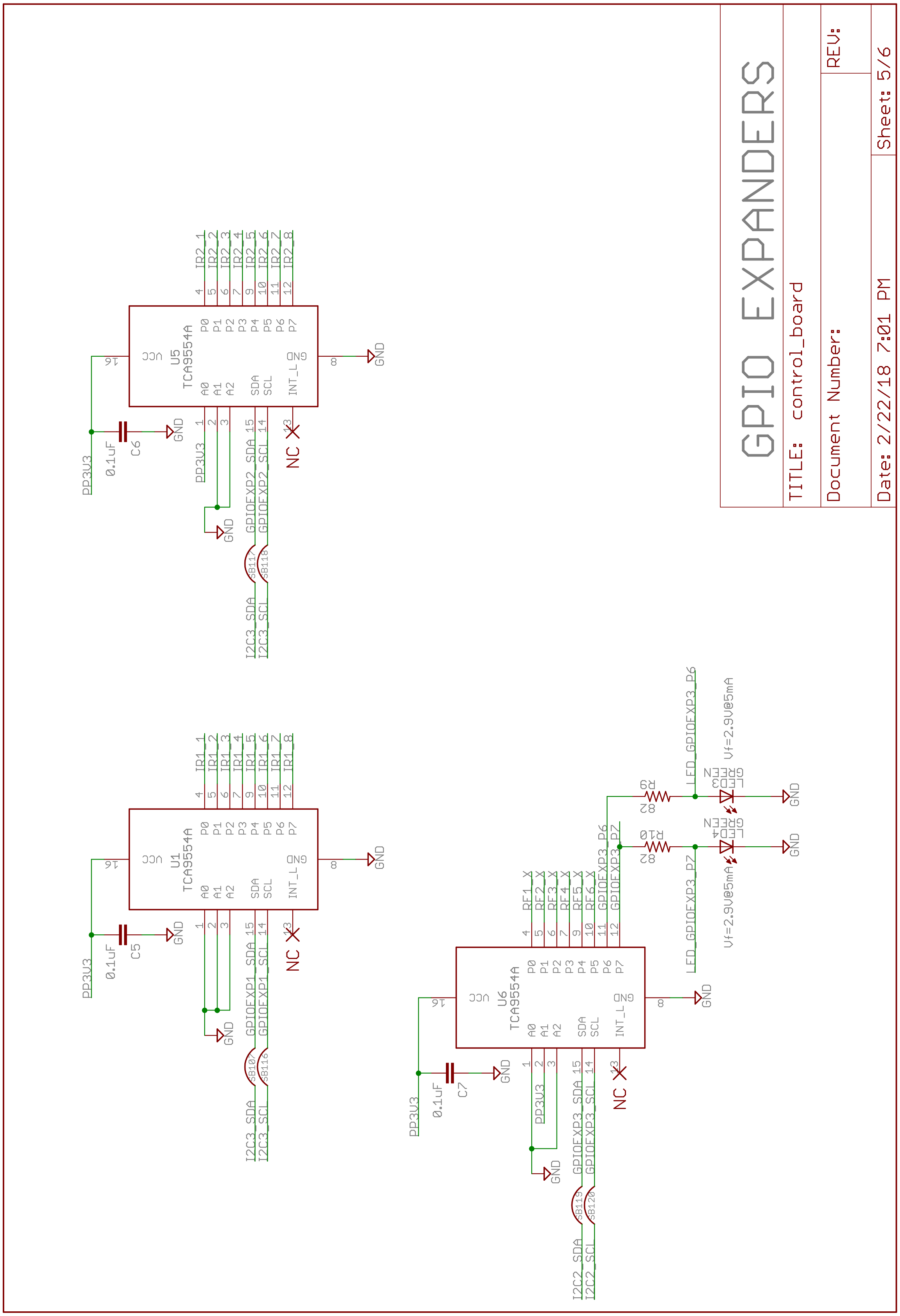
INTERCONNECT PCB SCHEMATIC


POWER
TITLE: control_board
Document Number:
REV:
Date: 2/22/18 7:01 PM
Sheet: 1/6
PP3V3
PP5V
PP7V
U7
AZ1085CD
U2
AZ1085CD
IN
OUT
ADJ
C1
10uF
C2
10uF
C10
22uF
C11
22uF
R1
178
R2
287
R3
100
R4
301
R7
82
R8
430
LED1
LED2
GREEN
LED_3V3
LED_5V
Vf=2.9V@5mA
GND
14.8V LiPo
12V SMPS
7V SMPS
PAD1
PAD2
PAD3
PAD4
PAD5
PAD6
PAD7
PAD8
PAD9
PAD10
WIREPAD3,17/1,1
PPBATT
PP12V


U4
STM32F446
PP7V
VBAT
VDD
VIN
V5V
E5V
5V
3V3
AVDD
AGND
IOREF
PA0_CONN
PA1_CONN
USART2_TX
USART2_RX
PA4_CONN
BRD_LED
PA6_CONN
PA7_CONN
PA8_CONN
PA9_CONN
PA10_CONN
PA11_CONN
PA12_CONN
TMS
TCK
PA15_CONN
PB0_CONN
PB1_CONN
PB2_CONN
SWO
PB4_CONN
PB5_CONN
PB6_CONN
PB7_CONN
PB8_CONN
PB9_CONN
PB10_CONN
PB12_CONN
PB13_CONN
PB14_CONN
PB15_CONN
M3_1
M3_3
I2C3_SCL
M2_2
RF4_G
RF5_G
RF3_G
LED_DBG1
M1_2
I2C3_SDA
M2_4
M1_1
M1_3
M2_1
M2_3
I2C2_SCL
M3_4
SERVO1
SERVO2
PC0_CONN
PC1_CONN
PC2_CONN
PC3_CONN
PC4_CONN
PC5_CONN
PC6_CONN
PC7_CONN
PC8_CONN
PC9_CONN
PC10_CONN
PC11_CONN
PC12_CONN
BRD_BUTTN
OSC32_IN
OSC32_OUT
PD2_CONN
OSC_IN
OSC_OUT
BOOT0_CONN
RESET_CONN
RF1_G
RF2_G
SW1
SW3
M3_2
SW2
SW4
M1_4
LED_DBG2
RF6_G
I2C2_SDA
DBG_BUTTON
RESET_BTN
BOOT0
RESET
NC
GND
STM32F446 NUCLEO
TITLE: control_board
Document Number:
REV:
Date: 2/22/18 7:01 PM
Sheet: 2/6


Microswitches
JST-XH-02-PIN-LONG-PAD
X1-1
X1-2
SW1
4.7k
R5
0.1uF
C3
100
R11
SW1_ILIM
PP3V3
GND
JST-XH-02-PIN-LONG-PAD
X12-1
X12-2
SW2
4.7k
R6
0.1uF
C4
100
R12
SW2_ILIM
PP3V3
GND
JST-XH-02-PIN-LONG-PAD
X19-1
X19-2
SW3
4.7k
R13
0.1uF
C8
100
R15
SW3_ILIM
PP3V3
GND
JST-XH-02-PIN-LONG-PAD
X20-1
X20-2
SW4
4.7k
R14
0.1uF
C9
100
R16
SW4_ILIM
PP3V3
GND
Debug Buttons
RESET_BTN
SW1
SPST_TACT-EVQQ2
GND
DBG_BUTTON
SW2
SPST_TACT-EVQQ2
GND
Debug LEDs
PP5V
R17
430
LED_DBG1_A
LED5
GREEN
Vf=2.9V@5mA
LED_DBG1_K
Q1
RE1C002UN
LED_DBG1
GND
PP3V3
R18
82
LED_DBG2_A
LED6
GREEN
Vf=2.9V@5mA
LED_DBG2_K
Q2
RE1C002UN
LED_DBG2
GND
Switches & LEDs
TITLE: control_board
Document Number:
REV:
Date: 2/22/18 7:01 PM
Sheet: 3/6


VL53L0X Rangefinders
IMU
IR PROXIMITY
I2C ADDRESS TABLE
RF Initial 0x52
I2C2
RF4 0x52
RF5 0x53
RF6 0x54
IMU_GYRO 0x69
IMU_ACCEL 0x19
IMU_MAG 0x1E
GPIO_EXPANDER3 0x3A
I2C3
RF1 0x52
RF2 0x53
RF3 0x54
GPIO_EXPANDER1 0x38
GPIO_EXPANDER2 0x39
I2C PULL-UPS
SENSOR CONNECTORS
TITLE: control_board
Document Number:
REV:
Date: 2/22/18 7:01 PM
Sheet: 4/6


GPIO EXPANDERS
TITLE: control_board
Document Number:
REV:
Date: 2/22/18 7:01 PM
Sheet: 5/6
U5
TCA9554A
U1
TCA9554A
U6
TCA9554A
C5
C6
C7
0.1uF
PP3V3
GND
NC
I2C3_SDA
I2C3_SCL
I2C2_SDA
I2C2_SCL
GPIOEXP1_SDA
GPIOEXP1_SCL
GPIOEXP2_SDA
GPIOEXP2_SCL
GPIOEXP3_SDA
GPIOEXP3_SCL
GPIOEXP3_P6
GPIOEXP3_P7
LED_GPIOEXP3_P6
LED_GPIOEXP3_P7
R9
R10
82
LED3
LED4
GREEN
Vf=2.9V@5mA


MOTOR DRIVERS
JST-XH-07-PIN-LONG-PAD
PP12V
PP5V
M1_1
M1_2
M1_3
M1_4
GND
PP12V
PP5V
M2_1
M2_2
M2_3
M2_4
GND
PP12V
PP5V
M3_1
M3_2
M3_3
M3_4
GND
SERVOS
JST-XH-03-PIN-LONG-PAD
PP5V
SERVO1_VCC
SERVO1
GND
PP5V
SERVO2_VCC
SERVO2
GND
MOTOR DRIVERS
TITLE: control_board
Document Number:
REV:
Date: 2/22/18 7:01 PM
Sheet: 6/6

Appendix C

INTERCONNECT PCB LAYOUT

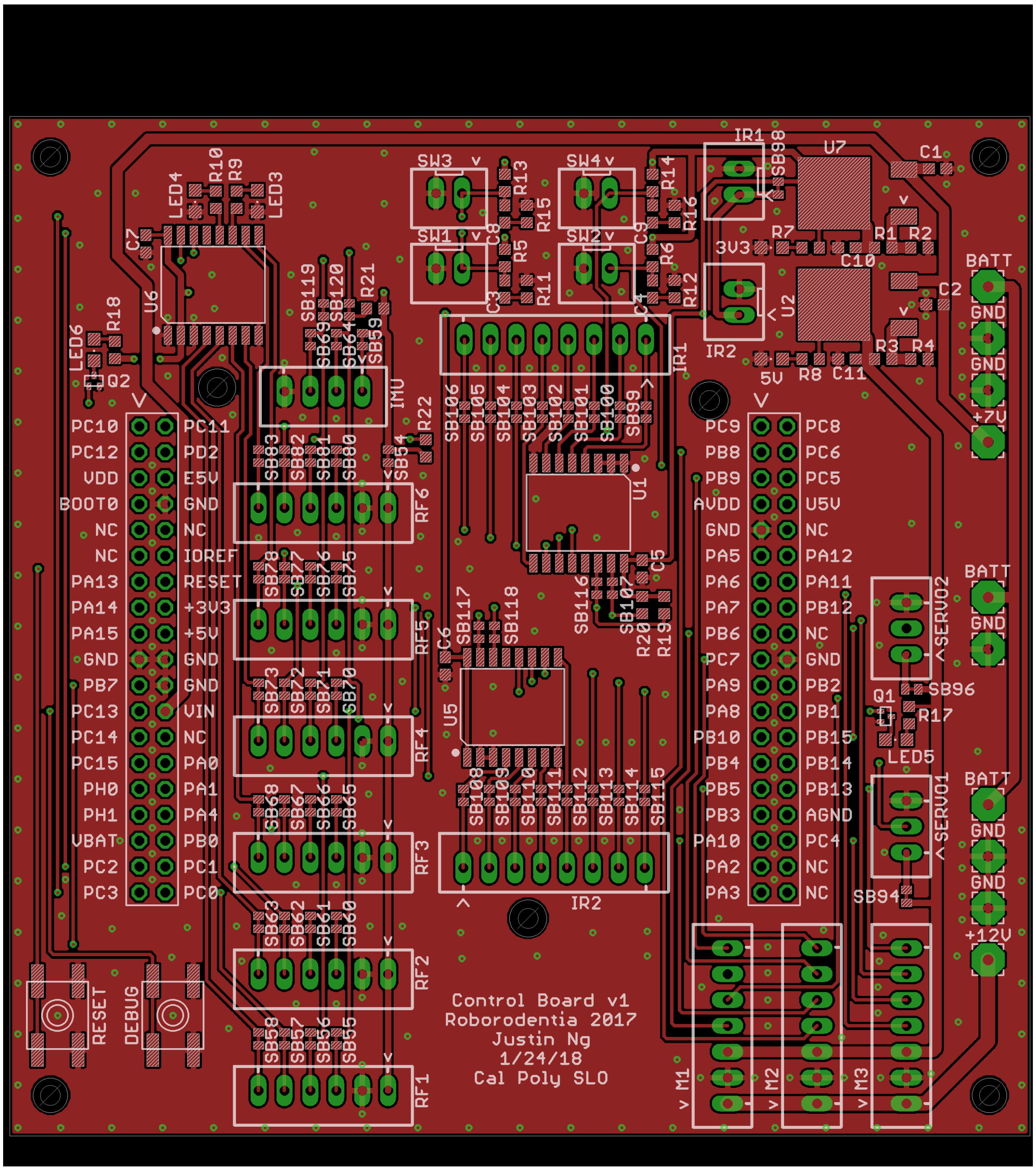


Figure C.1: Interconnect PCB Layout – Top Layer

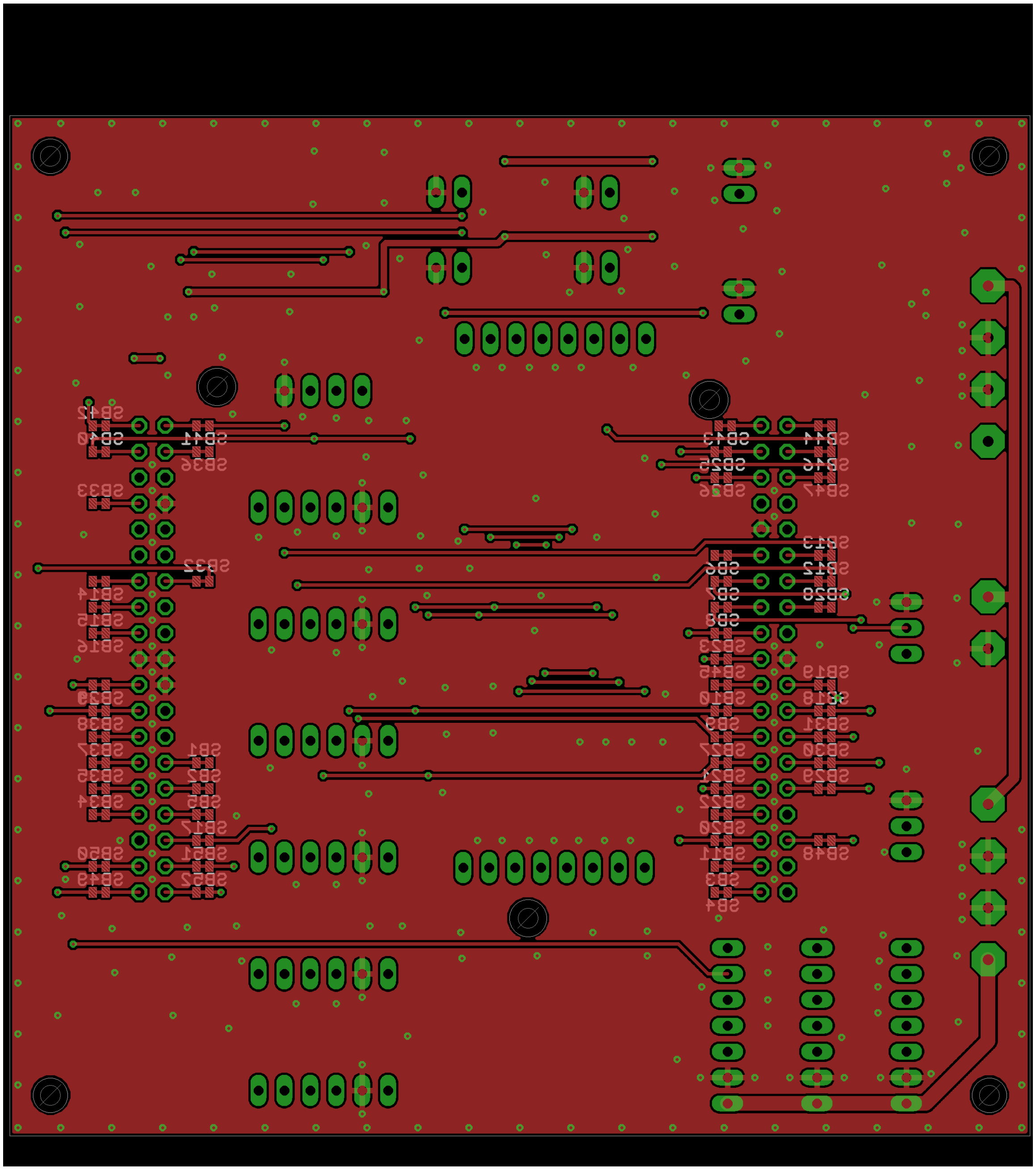


Figure C.2: Interconnect PCB Layout – Bottom Layer

Appendix D

INTERCONNECT PCB BILL OF MATERIALS

Table D.1: Interconnect PCB Bill of Materials

Qty	Value	Package	Parts	Description	DIGIKEY PN
2	AZ1085CD	TO252	U2, U7	AZ1085C	AZ1085CD-ADJTRG1DICT-ND
7	0.1uF	C0603	C3, C4, C5, C6, C7, C8, C9	CAPACITOR	445-5667-1-ND
2	10uF	C0603	C1, C2	CAPACITOR	490-13248-1-ND
2	22uF	C0603	C10, C11	CAPACITOR	490-10476-1-ND
3	TCA9554A	SOIC16	U1, U5, U6	I2C 8-bit GPIO Expander	296-45456-1-ND
6		JST-XH-02	X1, X12, X16, X18, X19, X20	JST XH Connector 2 Pin	455-2247-ND
2		JST-XH-03	X13, X14	JST XH Connector 3 Pin	455-2248-ND
1		JST-XH-04	X8	JST XH Connector 4 Pin	455-2249-ND
6		JST-XH-06	X2, X3, X4, X5, X6, X7	JST XH Connector 6 Pin	455-2271-ND
3		JST-XH-07	X9, X10, X11	JST XH Connector 7 Pin	455-2252-ND
2		JST-XH-08	X15, X17	JST XH Connector 8 Pin	455-2251-ND
6	GREEN	CHIPLED_0805	LED1, LED2, LED3, LED4, LED5, LED6	LED	732-4971-1-ND
2	RE1C002UN	SOT416FL	Q1, Q2	Logic-level N-FET	RE1C002UNTCLCT-ND
4	82	R0603	R7, R9, R10, R18	RESISTOR	311-82.0HRCT-ND

5	100	R0603	R3, R11, R12, R15, R16	RESISTOR	311-100HRCT-ND
1	178	R0603	R1	RESISTOR	311-178HRCT-ND
1	287	R0603	R2	RESISTOR	311-287HRCT-ND
1	301	R0603	R4	RESISTOR	311-301HRCT-ND
2	430	R0603	R8, R17	RESISTOR	311-430HRCT-ND
4	3.3k	R0603	R19, R20, R21, R22	RESISTOR	311-3.30KHRCT-ND
4	4.7k	R0603	R5, R6, R13, R14	RESISTOR	311-4.70KHRCT-ND
2	SPST	EVQ-Q2	SW1, SW2	SMT 6mm switch	P12955SCT-ND
1	STM32F446	NUCLEO	U4	STM32 NUCLEO-64	497-15882-ND
10		3,17/1,1	PAD1, PAD2, PAD3, PAD4, PAD5, PAD6, PAD7, PAD8, PAD9, PAD10	Wire PAD connect wire on PCB	n/a

105	BRIDGE	SB1, SB2, SB3, SB4, SB5, SB6, SB7, SB8, SB9, SB10, SB11, SB12, SB13, SB14, SB15, SB16, SB17, SB18, SB19, SB20, SB21, SB22, SB23, SB24, SB25, SB26, SB27, SB28, SB29, SB30, SB31, SB32, SB33, SB34, SB35, SB36, SB37, SB38, SB39, SB40, SB41, SB42, SB43, SB44, SB45, SB46, SB47, SB48, SB49, SB50, SB51, SB52, SB54, SB55, SB56, SB57, SB58, SB59, SB60, SB61, SB62, SB63, SB64, SB65, SB66, SB67, SB68, SB69, SB70, SB71, SB72, SB73, SB75, SB76, SB77, SB78, SB80, SB81, SB82, SB83, SB94, SB96, SB98, SB99, SB100, SB101, SB102, SB103, SB104, SB105, SB106, SB107, SB108, SB109, SB110, SB111, SB112, SB113, SB114, SB115, SB116, SB117, SB118, SB119, SB120	Solder bridge with knife-cuttable prebridged connection.	n/a

Appendix E

STM32CUBEMX REPORT

1. Description

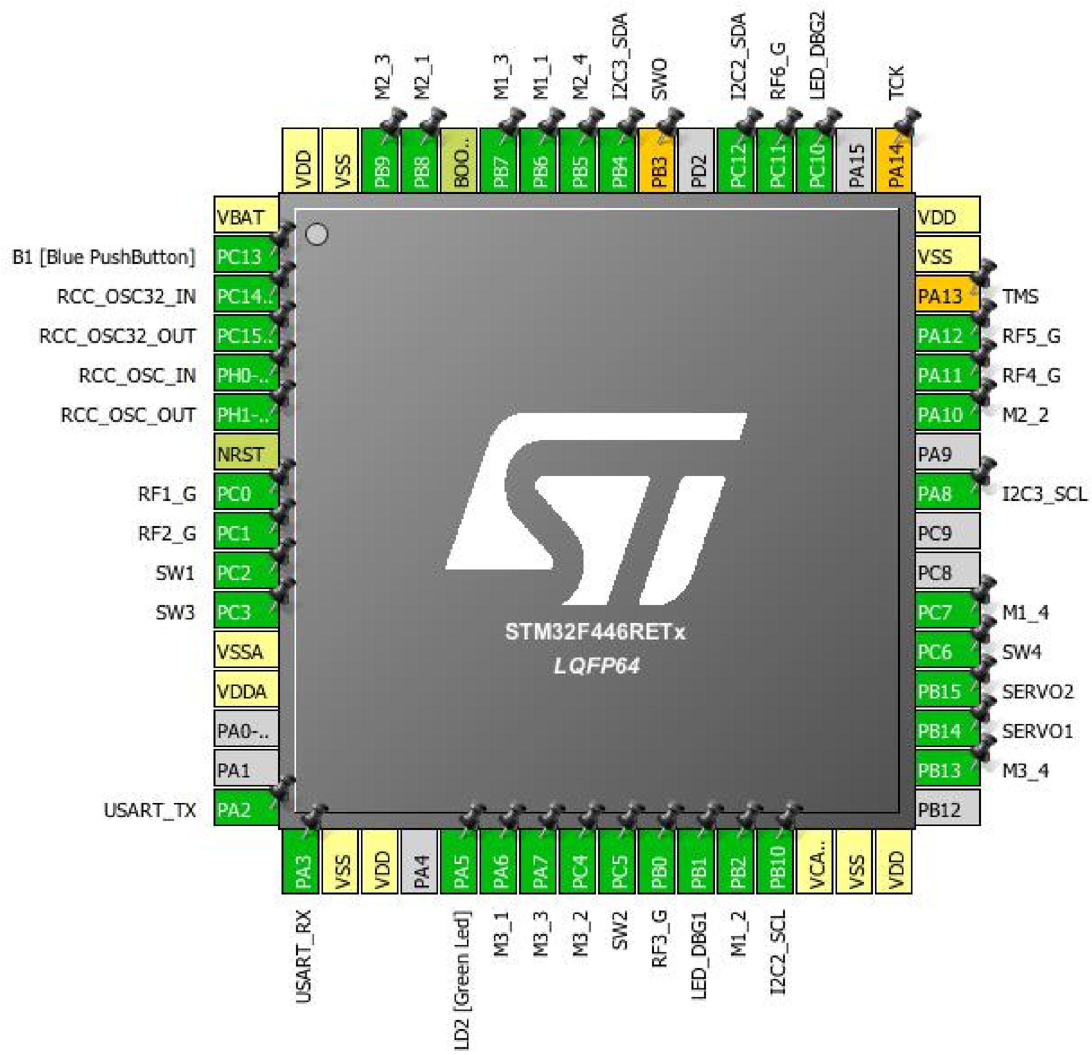
1.1. Project

Project Name	firmware_new
Board Name	NUCLEO-F446RE
Generated with:	STM32CubeMX 4.24.0
Date	05/19/2018

1.2. MCU

MCU Series	STM32F4
MCU Line	STM32F446
MCU name	STM32F446RETx
MCU Package	LQFP64
MCU Pin number	64

2. Pinout Configuration



3. Pins Configuration

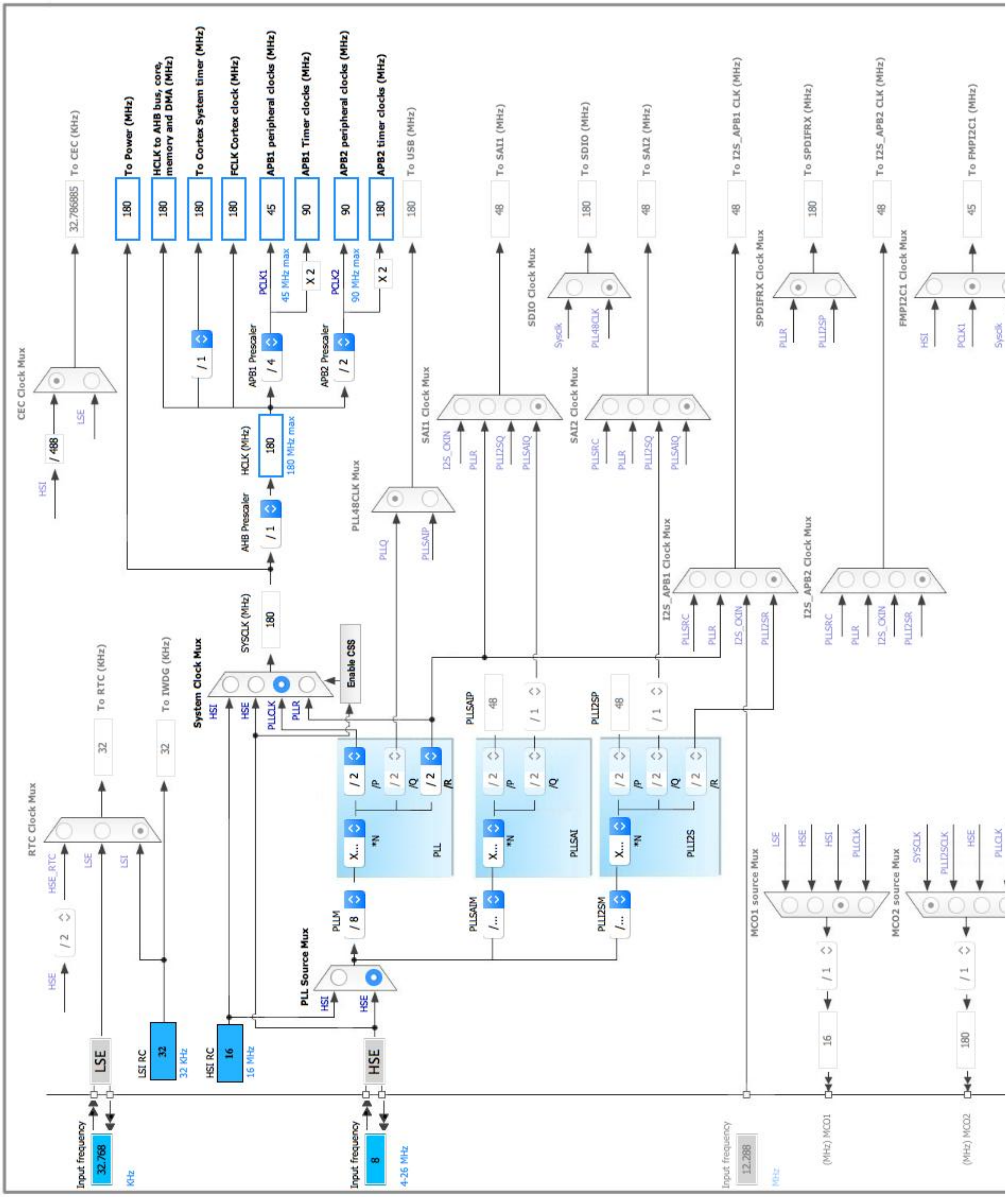
Pin Number LQFP64	Pin Name (function after reset)	Pin Type	Alternate Function(s)	Label
1	VBAT	Power		
2	PC13	I/O	GPIO_EXTI13	B1 [Blue PushButton]
3	PC14-OSC32_IN	I/O	RCC_OSC32_IN	
4	PC15-OSC32_OUT	I/O	RCC_OSC32_OUT	
5	PH0-OSC_IN	I/O	RCC_OSC_IN	
6	PH1-OSC_OUT	I/O	RCC_OSC_OUT	
7	NRST	Reset		
8	PC0	I/O	GPIO_EXTI0	RF1_G
9	PC1	I/O	GPIO_EXTI1	RF2_G
10	PC2 *	I/O	GPIO_Output	SW1
11	PC3 *	I/O	GPIO_Output	SW3
12	VSSA	Power		
13	VDDA	Power		
16	PA2	I/O	USART2_TX	USART_TX
17	PA3	I/O	USART2_RX	USART_RX
18	VSS	Power		
19	VDD	Power		
21	PA5 *	I/O	GPIO_Output	LD2 [Green Led]
22	PA6	I/O	TIM3_CH1	M3_1
23	PA7	I/O	TIM3_CH2	M3_3
24	PC4 *	I/O	GPIO_Output	M3_2
25	PC5 *	I/O	GPIO_Output	SW2
26	PB0 *	I/O	GPIO_Input	RF3_G
27	PB1 *	I/O	GPIO_Output	LED_DBG1
28	PB2 *	I/O	GPIO_Output	M1_2
29	PB10	I/O	I2C2_SCL	
30	VCAP_1	Power		
31	VSS	Power		
32	VDD	Power		
34	PB13 *	I/O	GPIO_Output	M3_4
35	PB14	I/O	TIM12_CH1	SERVO1
36	PB15	I/O	TIM12_CH2	SERVO2
37	PC6 *	I/O	GPIO_Output	SW4
38	PC7 *	I/O	GPIO_Output	M1_4
41	PA8	I/O	I2C3_SCL	
43	PA10 *	I/O	GPIO_Output	M2_2

Pin Number LQFP64	Pin Name (function after reset)	Pin Type	Alternate Function(s)	Label
44	PA11	I/O	GPIO_EXTI11	RF4_G
45	PA12	I/O	GPIO_EXTI12	RF5_G
46	PA13 **	I/O	SYS_JTMS-SWDIO	TMS
47	VSS	Power		
48	VDD	Power		
49	PA14 **	I/O	SYS_JTCK-SWCLK	TCK
51	PC10 *	I/O	GPIO_Output	LED_DBG2
52	PC11 *	I/O	GPIO_Input	RF6_G
53	PC12	I/O	I2C2_SDA	
55	PB3 **	I/O	SYS_JTDO-SWO	SWO
56	PB4	I/O	I2C3_SDA	
57	PB5 *	I/O	GPIO_Output	M2_4
58	PB6	I/O	TIM4_CH1	M1_1
59	PB7	I/O	TIM4_CH2	M1_3
60	BOOT0	Boot		
61	PB8	I/O	TIM4_CH3	M2_1
62	PB9	I/O	TIM4_CH4	M2_3
63	VSS	Power		
64	VDD	Power		

* The pin is affected with an I/O function

** The pin is affected with a peripheral function but no peripheral mode is activated

4. Clock Tree Configuration



5. IPs and Middleware Configuration

5.1. I2C2

I2C: I2C

5.1.1. Parameter Settings:

Master Features:	
I2C Speed Mode	**Fast Mode ***
I2C Clock Speed (Hz)	400000
Fast Mode Duty Cycle	Duty cycle Tlow/Thigh = 2
Slave Features:	
Clock No Stretch Mode	Disabled
Primary Address Length selection	7-bit
Dual Address Acknowledged	Disabled
Primary slave address	0
General Call address detection	Disabled

5.2. I2C3

I2C: I2C

5.2.1. Parameter Settings:

Master Features:	
I2C Speed Mode	**Fast Mode ***
I2C Clock Speed (Hz)	400000
Fast Mode Duty Cycle	Duty cycle Tlow/Thigh = 2
Slave Features:	
Clock No Stretch Mode	Disabled
Primary Address Length selection	7-bit
Dual Address Acknowledged	Disabled
Primary slave address	0
General Call address detection	Disabled

5.3. RCC

High Speed Clock (HSE): Crystal/Ceramic Resonator
Low Speed Clock (LSE) : Crystal/Ceramic Resonator

5.3.1. Parameter Settings:

System Parameters:	
VDD voltage (V)	3.3
Instruction Cache	Enabled
Prefetch Buffer	Enabled
Data Cache	Enabled
Flash Latency(WS)	5 WS (6 CPU cycle)
RCC Parameters:	
HSI Calibration Value	16
TIM Prescaler Selection	Disabled
HSE Startup Timout Value (ms)	100
LSE Startup Timout Value (ms)	5000
Power Parameters:	
Power Regulatror Voltage Scale	Power Regulator Voltage Scale 1
Power Over Drive	Enabled

5.4. SYS

Timebase Source: SysTick

5.5. TIM3

Clock Source : Internal Clock
Channel1: PWM Generation CH1
Channel2: PWM Generation CH2

5.5.1. Parameter Settings:

Counter Settings:	
Prescaler (PSC - 16 bits value)	**1 ***
Counter Mode	Up
Counter Period (AutoReload Register - 16 bits value)	**2047 ***
Internal Clock Division (CKD)	No Division

Trigger Output (TRGO) Parameters:

Master/Slave Mode	Disable (no sync between this TIM (Master) and its Slaves
Trigger Event Selection	Reset (UG bit from TIMx_EGR)

PWM Generation Channel 1:

Mode	PWM mode 1
Pulse (16 bits value)	0
Fast Mode	Disable
CH Polarity	High

PWM Generation Channel 2:

Mode	PWM mode 1
Pulse (16 bits value)	0
Fast Mode	Disable
CH Polarity	High

5.6. TIM4

mode: Clock Source
Channel1: PWM Generation CH1
Channel2: PWM Generation CH2
Channel3: PWM Generation CH3
Channel4: PWM Generation CH4

5.6.1. Parameter Settings:

Counter Settings:

Prescaler (PSC - 16 bits value)	**1 ***
Counter Mode	Up
Counter Period (AutoReload Register - 16 bits value)	**2047 ***
Internal Clock Division (CKD)	No Division

Trigger Output (TRGO) Parameters:

Master/Slave Mode	Disable (no sync between this TIM (Master) and its Slaves
Trigger Event Selection	Reset (UG bit from TIMx_EGR)

PWM Generation Channel 1:

Mode	PWM mode 1
Pulse (16 bits value)	0
Fast Mode	Disable
CH Polarity	High

PWM Generation Channel 2:

Mode	PWM mode 1

Pulse (16 bits value)	0
Fast Mode	Disable
CH Polarity	High

PWM Generation Channel 3:

Mode	PWM mode 1
Pulse (16 bits value)	0
Fast Mode	Disable
CH Polarity	High

PWM Generation Channel 4:

Mode	PWM mode 1
Pulse (16 bits value)	0
Fast Mode	Disable
CH Polarity	High

5.7. TIM5

mode: Clock Source

5.7.1. Parameter Settings:

Counter Settings:

Prescaler (PSC - 16 bits value)	**9000 ***
Counter Mode	Up
Counter Period (AutoReload Register - 32 bits value)	**0xFFFFFFFF ***
Internal Clock Division (CKD)	No Division

Trigger Output (TRGO) Parameters:

Master/Slave Mode	Disable (no sync between this TIM (Master) and its Slaves
Trigger Event Selection	Reset (UG bit from TIMx_EGR)

5.8. TIM12

mode: Clock Source

Channel1: PWM Generation CH1

Channel2: PWM Generation CH2

5.8.1. Parameter Settings:

Counter Settings:

Prescaler (PSC - 16 bits value)	**354 ***
Counter Mode	Up
Counter Period (AutoReload Register - 16 bits value)	**3178 ***
Internal Clock Division (CKD)	No Division
PWM Generation Channel 1:	
Mode	PWM mode 1
Pulse (16 bits value)	**63 ***
Fast Mode	Disable
CH Polarity	High
PWM Generation Channel 2:	
Mode	PWM mode 1
Pulse (16 bits value)	**63 ***
Fast Mode	Disable
CH Polarity	High

5.9. USART2

Mode: Asynchronous

5.9.1. Parameter Settings:

Basic Parameters:	
Baud Rate	**921600 ***
Word Length	8 Bits (including Parity)
Parity	None
Stop Bits	1
Advanced Parameters:	
Data Direction	Receive and Transmit
Over Sampling	16 Samples

*** User modified value**

6. System Configuration

6.1. GPIO configuration

IP	Pin	Signal	GPIO mode	GPIO pull/up pull down	Max Speed	User Label
I2C2	PB10	I2C2_SCL	Alternate Function Open Drain	Pull-up	**Very High** *	
	PC12	I2C2_SDA	Alternate Function Open Drain	Pull-up	**Very High** *	
I2C3	PA8	I2C3_SCL	Alternate Function Open Drain	Pull-up	**Very High** *	
	PB4	I2C3_SDA	Alternate Function Open Drain	Pull-up	**Very High** *	
RCC	PC14-OSC32_IN	RCC_OSC32_IN	n/a	n/a	n/a	
	PC15-OSC32_OUT	RCC_OSC32_OUT	n/a	n/a	n/a	
	PH0-OSC_IN	RCC_OSC_IN	n/a	n/a	n/a	
	PH1-OSC_OUT	RCC_OSC_OUT	n/a	n/a	n/a	
TIM3	PA6	TIM3_CH1	Alternate Function Push Pull	No pull-up and no pull-down	Low	M3_1
	PA7	TIM3_CH2	Alternate Function Push Pull	No pull-up and no pull-down	Low	M3_3
TIM4	PB6	TIM4_CH1	Alternate Function Push Pull	No pull-up and no pull-down	Low	M1_1
	PB7	TIM4_CH2	Alternate Function Push Pull	No pull-up and no pull-down	Low	M1_3
	PB8	TIM4_CH3	Alternate Function Push Pull	No pull-up and no pull-down	Low	M2_1
	PB9	TIM4_CH4	Alternate Function Push Pull	No pull-up and no pull-down	Low	M2_3
TIM12	PB14	TIM12_CH1	Alternate Function Push Pull	No pull-up and no pull-down	Low	SERVO1
	PB15	TIM12_CH2	Alternate Function Push Pull	No pull-up and no pull-down	Low	SERVO2
USART2	PA2	USART2_TX	Alternate Function Push Pull	Pull-up	**Very High** *	USART_TX
	PA3	USART2_RX	Alternate Function Push Pull	Pull-up	**Very High** *	USART_RX
Single Mapped Signals	PA13	SYS_JTMS-SWDIO	n/a	n/a	n/a	TMS
	PA14	SYS_JTCK-SWCLK	n/a	n/a	n/a	TCK
	PB3	SYS_JTDO-	n/a	n/a	n/a	SWO

IP	Pin	Signal	GPIO mode	GPIO pull/up pull down	Max Speed	User Label
		SWO				
GPIO	PC13	GPIO_EXTI13	**External Interrupt Mode with Falling edge trigger detection**	No pull-up and no pull-down	n/a	B1 [Blue PushButton]
	PC0	GPIO_EXTI0	**External Interrupt Mode with Falling edge trigger detection**	**Pull-up ***	n/a	RF1_G
	PC1	GPIO_EXTI1	**External Interrupt Mode with Falling edge trigger detection**	**Pull-up ***	n/a	RF2_G
	PC2	GPIO_Output	Output Push Pull	No pull-up and no pull-down	Low	SW1
	PC3	GPIO_Output	Output Push Pull	No pull-up and no pull-down	Low	SW3
	PA5	GPIO_Output	Output Push Pull	No pull-up and no pull-down	Low	LD2 [Green Led]
	PC4	GPIO_Output	Output Push Pull	No pull-up and no pull-down	Low	M3_2
	PC5	GPIO_Output	Output Push Pull	No pull-up and no pull-down	Low	SW2
	PB0	GPIO_Input	Input mode	**Pull-up ***	n/a	RF3_G
	PB1	GPIO_Output	Output Push Pull	No pull-up and no pull-down	Low	LED_DBG1
	PB2	GPIO_Output	Output Push Pull	No pull-up and no pull-down	Low	M1_2
	PB13	GPIO_Output	Output Push Pull	No pull-up and no pull-down	Low	M3_4
	PC6	GPIO_Output	Output Push Pull	No pull-up and no pull-down	Low	SW4
	PC7	GPIO_Output	Output Push Pull	No pull-up and no pull-down	Low	M1_4
	PA10	GPIO_Output	Output Push Pull	No pull-up and no pull-down	Low	M2_2
	PA11	GPIO_EXTI11	**External Interrupt Mode with Falling edge trigger detection**	**Pull-up ***	n/a	RF4_G
	PA12	GPIO_EXTI12	**External Interrupt Mode with Falling edge trigger detection**	**Pull-up ***	n/a	RF5_G
	PC10	GPIO_Output	Output Push Pull	No pull-up and no pull-down	Low	LED_DBG2
	PC11	GPIO_Input	Input mode	No pull-up and no pull-down	n/a	RF6_G
	PB5	GPIO_Output	Output Push Pull	No pull-up and no pull-down	Low	M2_4

6.2. DMA configuration

DMA request	Stream	Direction	Priority
I2C2_RX	DMA1_Stream2	Peripheral To Memory	Low
I2C2_TX	DMA1_Stream7	Memory To Peripheral	**High ***
I2C3_RX	DMA1_Stream1	Peripheral To Memory	Low
I2C3_TX	DMA1_Stream4	Memory To Peripheral	**High ***
USART2_RX	DMA1_Stream5	Peripheral To Memory	**Very High ***
USART2_TX	DMA1_Stream6	Memory To Peripheral	**Very High ***

I2C2_RX: DMA1_Stream2 DMA request Settings:

Mode: Normal
Use fifo: Disable
Peripheral Increment: Disable
Memory Increment: **Enable ***
Peripheral Data Width: Byte
Memory Data Width: Byte

I2C2_TX: DMA1_Stream7 DMA request Settings:

Mode: Normal
Use fifo: Disable
Peripheral Increment: Disable
Memory Increment: **Enable ***
Peripheral Data Width: Byte
Memory Data Width: Byte

I2C3_RX: DMA1_Stream1 DMA request Settings:

Mode: Normal
Use fifo: Disable
Peripheral Increment: Disable
Memory Increment: **Enable ***
Peripheral Data Width: Byte
Memory Data Width: Byte

I2C3_TX: DMA1_Stream4 DMA request Settings:

Mode:	Normal
Use fifo:	Disable
Peripheral Increment:	Disable
Memory Increment:	**Enable ***
Peripheral Data Width:	Byte
Memory Data Width:	Byte

USART2_RX: DMA1_Stream5 DMA request Settings:

Mode:	**Circular ***
Use fifo:	Disable
Peripheral Increment:	Disable
Memory Increment:	**Enable ***
Peripheral Data Width:	Byte
Memory Data Width:	Byte

USART2_TX: DMA1_Stream6 DMA request Settings:

Mode:	Normal
Use fifo:	Disable
Peripheral Increment:	Disable
Memory Increment:	**Enable ***
Peripheral Data Width:	Byte
Memory Data Width:	Byte

6.3. NVIC configuration

Interrupt Table	Enable	Preenmption Priority	SubPriority
Non maskable interrupt	true	0	0
Hard fault interrupt	true	0	0
Memory management fault	true	0	0
Pre-fetch fault, memory access fault	true	0	0
Undefined instruction or illegal state	true	0	0
System service call via SWI instruction	true	0	0
Debug monitor	true	0	0
Pendable request for system service	true	0	0
System tick timer	true	0	0
DMA1 stream1 global interrupt	true	0	0
DMA1 stream2 global interrupt	true	0	0
DMA1 stream4 global interrupt	true	0	0
DMA1 stream5 global interrupt	true	0	0
DMA1 stream6 global interrupt	true	0	0
I2C2 event interrupt	true	0	0
I2C2 error interrupt	true	0	0
USART2 global interrupt	true	0	0
DMA1 stream7 global interrupt	true	0	0
I2C3 event interrupt	true	0	0
I2C3 error interrupt	true	0	0
PVD interrupt through EXTI line 16	unused		
Flash global interrupt	unused		
RCC global interrupt	unused		
EXTI line 0 interrupt	unused		
EXTI line 1 interrupt	unused		
TIM3 global interrupt	unused		
TIM4 global interrupt	unused		
EXTI line[15:10] interrupts	unused		
TIM8 break interrupt and TIM12 global interrupt	unused		
TIM5 global interrupt	unused		
FPU global interrupt	unused		

*** User modified value**

7. Power Consumption Calculator report

7.1. Microcontroller Selection

Series	STM32F4
Line	STM32F446
MCU	STM32F446RETx
Datasheet	027107_Rev6

7.2. Parameter Selection

Temperature	25
Vdd	3.3

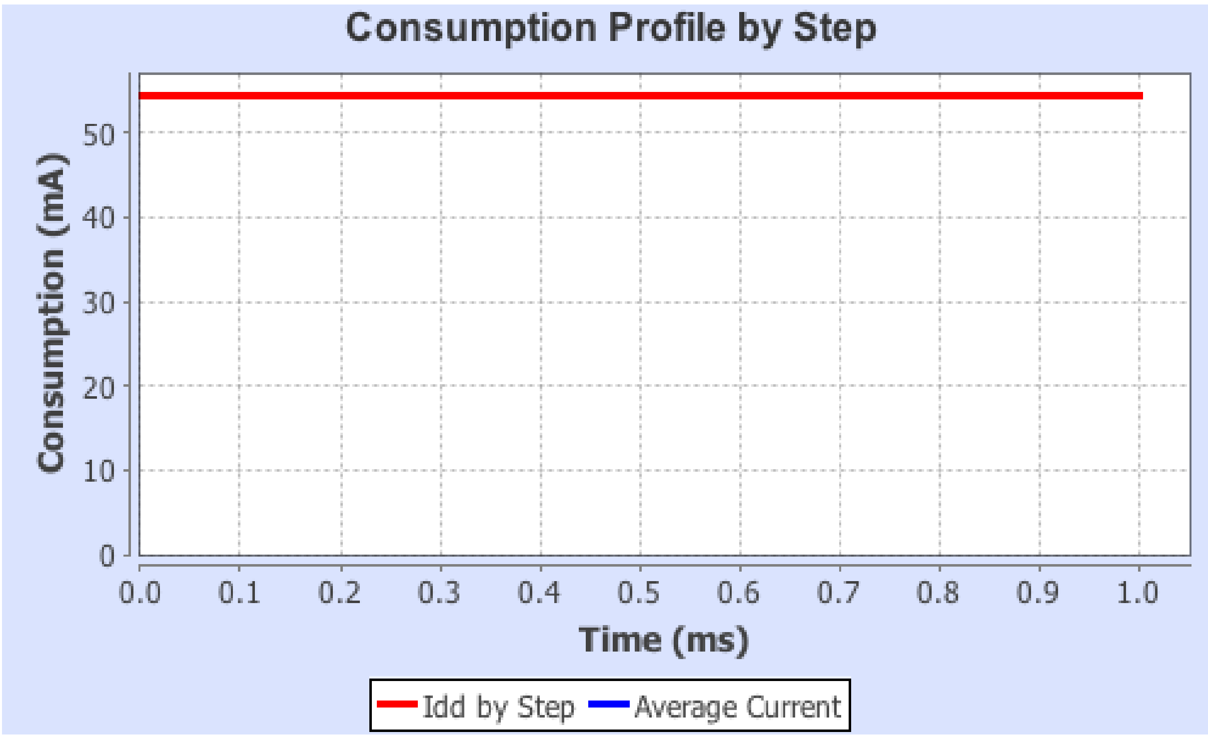
7.3. Sequence

Step	Step1
Mode	RUN
Vdd	3.3
Voltage Source	Vbus
Range	Scale1-High
Fetch Type	RAM/FLASH/REGON/ART/PREFETCH
Clock Configuration	HSE PLL
Clock Source Frequency	4 MHz
CPU Frequency	180 MHz
Peripherals	DMA1 DMA2 GPIOA GPIOB GPIOC I2C2 I2C3 SYS TIM1 TIM2 TIM3 TIM4 TIM12 USART2
Additional Cons.	0 mA
Average Current	54.31 mA
Duration	1 ms
DMIPS	225.0
Ta Max	96.76
Category	In DS Table

7.4. RESULTS

Sequence Time	1 ms	Average Current	54.31 mA
Battery Life	0	Average DMIPS	225.0 DMIPS

7.5. Chart



8. Software Project

8.1. Project Settings

Name	Value
Project Name	firmware_new
Project Folder	/Users/justinng/Documents/Github/roborodentia2017/firmware_new
Toolchain / IDE	Makefile
Firmware Package Name and Version	STM32Cube FW_F4 V1.18.0

8.2. Code Generation Settings

Name	Value
STM32Cube Firmware Library Package	Copy all used libraries into the project folder
Generate peripheral initialization as a pair of '.c/.h' files	Yes
Backup previously generated files when re-generating	No
Delete previously generated files when not re-generated	Yes
Set all free pins as analog (to optimize the power consumption)	No

9. Software Pack Report