# Educational Coordinated Robotics

By

Kan, John

Lu, Louis

Null, W David

Final Report for ECE 445, Senior Design, Fall 2018

TA: Dongwei Shi

11 December 2018

Project No. 6

# Abstract

The SeaPerch underwater ROV (Remotely Operated Vehicle) competition is focused on introducing middle and high-school students to STEM fields through Naval related challenges. The Office of Naval Research is the primary sponsor of this national program which has operated since 2011. Through entering this competition, students build a very basic robotic system out of PVC pipe, DC motors, and Ethernet cable. The Navy recently began to seek proposals for new educational technology platforms and curriculum to add an advanced level to the competition in order to introduce students to emerging robotic trends. Our team is connected with a few of the researchers leading this new SeaPerch initiative at Naval Research Laboratory in Stennis Space Center, Mississippi. At its core, our project seeks to meet this need through a modular prototype system capable of giving students access to advanced sensors, motor control, and robotic coordination technology. Coordinated technologies are currently at the forefront of research efforts in the field of robotics. They enable novel and effective solutions including but not limited to search and rescue operations, distributed sensing applications, and agricultural tasks. However, communication between multiple robots is difficult to implement in educational environments because of the cost of materials and complexity of the systems. The system we designed includes a PCB and software libraries that integrate with existing off the shelf educational platforms, and enable students to develop their own network of robots with the ability to communicate.

# Contents

# 1 Introduction

## 1.1 Purpose

The SeaPerch underwater ROV (Remotely Operated Vehicle) competition is focused on introducing middle and high-school students to STEM fields through Naval related challenges [1]. The Office of Naval Research is the primary sponsor of this national program which has operated since 2011. Through entering this competition, students build a very basic robotic system out of PVC pipe, DC motors, and Ethernet cable. The Navy recently began to seek proposals for new educational technology platforms and curriculum to add an advanced level to the competition in order to introduce students to emerging robotic trends. Our team is connected with a few of the researchers leading this new SeaPerch initiative at Naval Research Laboratory in Stennis Space Center, Mississippi. At its core, our project seeks to meet this need through a modular prototype system capable of giving students access to advanced sensors, motor control, and robotic coordination technology.

## 1.2 Functionality

Our primary goal was to develop a PCB and accompanying software that enables a student to easily experiment with a network of coordinated robots. Our proposed and completed project fulfilled the high-level requirements of the project. Namely, the three goals identified were:
(1) There must be three designed and built robotic systems capable of actuation and environmental sensing.
(2) Each robotic system must be capable of connecting to a communication network with other robotic systems in order to transmit data and receive commands.
(3) Each robotic system must contain at least one marked code package which can be modified for customized control for the coordinated robotic network. Figure 1 shows the working product.

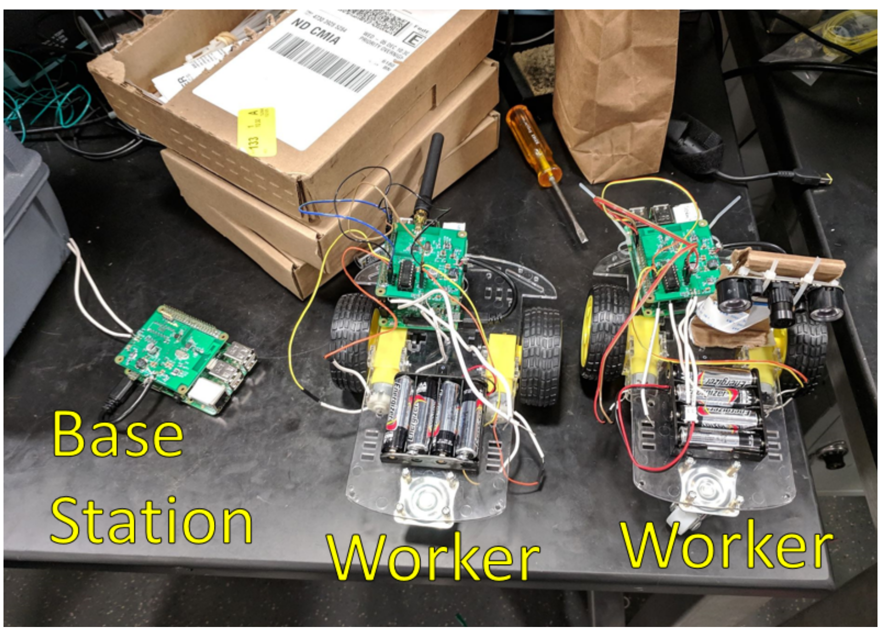


Figure 1: Working product

### 1.3 Subsystem Overview

The design process to meet the high level requirements began with the identification of various smaller requirement blocks. At the high level, the robots needed to communicate. The need for communication motivated the development of a radio frequency (RF) network block to be included in the overall design. Additionally, the robots required basic sensing to interact with the outside environment, which provided the impetus for a sensing unit which would include some sort of optical, accelerometer, and gyroscopic data functions. The actuation requirement motivated a motor drive and control unit. Finally, in order to power all of these systems and allow them to work properly and interact, a power system was developed. Thus, four main design blocks were identified and constructed to meet the specifications laid out by the high-level requirements: RF, sensing (to include inertial and optical data inputs), motor control, and power management. Additionally, these design blocks must all interact with the Raspberry Pi. The overall flow and interconnection between the various modules are illustrated in Figure 2. Note that a battery unit was also created in order to provide the energy for the entire system.

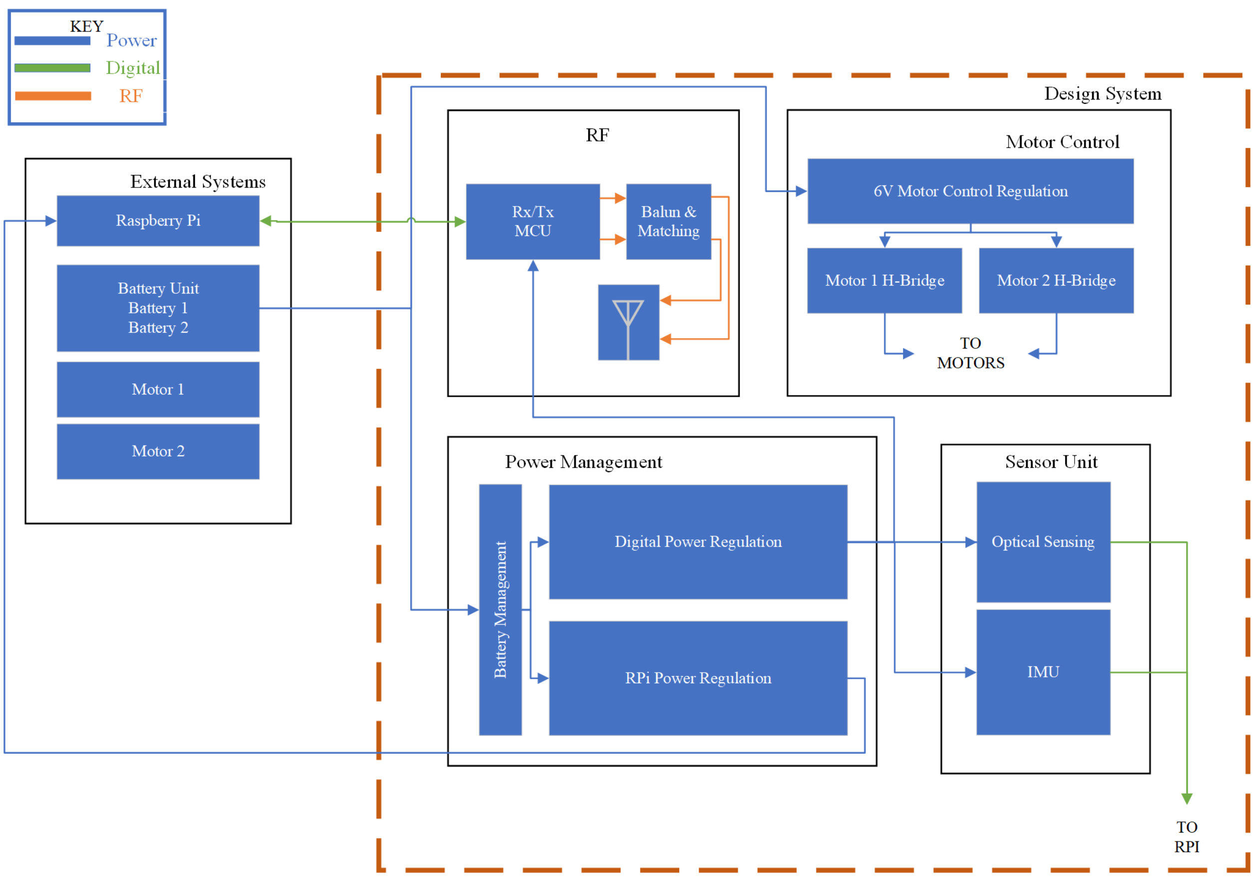


Figure 2: Single Robot Hardware Block Diagram

## 2 Design

## 2.1 Power Electronics

The power electronics had the greatest amount of design work necessary in order to determine which chips and components to utilize. To completely understand the system, absolute power requirements were determined. First, we accounted for the fact that the digital circuitry had a higher voltage sensitivity than the analog circuits. In concordance with this, the fact that the motor driver had a high potential for current spikes motivated the need to decouple the power sources for the motors, and the rest of the subsystems. Furthermore, we developed a regulated power circuit in order to protect the highly sensitive digital components of our system. This design is reflected in Figure 2.

The regulated voltage circuit was needed to not only power the Raspberry Pi, but also the sensing and RF ICs on the board. We calculated the maximum current draw of the Raspberry Pi plus a camera to be 1.25 A. Additionally, the maximum current draw for both the IMU and the RF chips was no more than 0.5 A[2]. With these upper bounds, the necessary maximum current draw required for this circuit is approximately 1.75 A. From these rough estimates, we designed a dc-dc converter capable of producing a 5V output with a 2A current maximum. In our implementation, we used the LTC3624 buck converter, capable of producing a regulated 5V output from an input in the range of 5-17V, with feedback, and a maximum of 2A.

The choice of a buck converter, the LTC3624, was not the original course or preparation in the design process. Our original alternative design used the LT3112, a 5V 2A buck-boost regulator. This design choice resulted from two different uncertainties: (1) power requirements for the board and (2) the cost of the chip. The LT3112 is approximately $9 per chip. Additionally, the power requirements themselves were not fully evaluated, and because of this, while the chip itself may have been able to provide a 5V, 2A output, the modes of operation, as well as the components required to be utilized in conjunction with the IC not only had to be large in physical size and component rating, but also had a high cost. Because of marginal returns on a wide input voltage for an increased cost, the design required a new DC-DC regulator.

Utilizing the LTC3624 required a few more design components. To provide power, the output inductor needed to be properly designed. In a buck converter, which has the conversion ratio $\frac{V_{out}}{V_{in}}$

$$\frac{V_{out}}{V_{out}} = D \tag{1}$$

where D is the duty ratio, the inductor must be designed so that the the specified output can be achieved. Additionally, knowing that the inductor has a voltage-current relationship of

$$V_L = L\frac{di_L}{dt} \tag{2}$$

The ripple across the inductor, found by utilizing only the first section of the duty ratio, is then found to be

$$\Delta i = \frac{V_L}{L}DT \tag{3}$$

$$= \frac{V_L * D}{L * f_{sw}} \tag{4}$$

indicating that the ripple current is based upon the voltage across the inductor, the inductance of the material, the duty ratio, and the switching frequency. Note that for a buck converter, the general voltage relationship for $V_L$ is $V_L = V_1 - V_2$. The key point from this derivation is that the inductor value is based upon maximum current ripple, input and output voltages, switching frequency and duty ratio. Next, the current ripple was chosen. A good rule of thumb is to give a current ripple of approximately 10% of maximum current [3]. Thus, the maximum current ripple is around 10%, making L to be calculated as:

$$L \geq \frac{(V_{in} - V_{out}) * V_{out}}{\Delta i * f_{sw} * V_{in}} \tag{5}$$

$$\geq \frac{(7-5)*5}{0.2*1MHz*7} \tag{6}$$

$$L \geq 0.793\mu H \tag{7}$$

A standard part, an inductance of 3.3$\mu$H was chosen. Recalculating the ripple current, it is found that the ripple current becomes approximately 0.192 A, which is within the 10% boundaries selected. Thus, we chose a 3.3$\mu$H inductor able to withstand up to 2A for this circuit.

One final design aspect to this buck converter is the input and output capacitances, which were eventually selected to be around 47 $\mu$F. When designing DC-DC converters, it is always necessary to have a large bulk capacitance, as well as some smaller high-frequency filtering capacitance. After evaluating the ripple, the capacitance was chosen to be 47 $\mu$F on the output and 20 $\mu$F capacitors were placed on the input. This ensures that in the steady state the voltage has low ripple and is stable.

A similar process was followed for linear regulator, however, because a linear regulator essentially a resistor divider that is controlled using a transistor. In this case, the application notes [4] were followed, and a resistive combination that had a ratio of $\frac{3.3}{5}$ was chosen in order to match output and input voltages. Note that this is quite inefficient – a resistor divider only can have efficiencies that are a ratio of output power to input power (around 60% for this particular LDO).

Finally, the entire system (Figure 3) was simulated (Figure 4), and a simple resistor divider network was added to the shutdown pins of both LTC3624 and LT3065 to ensure that a constant voltage threshold exists at the input of the power converters before the power is then transferred down the line. This provides another factor of protection in the circuit for the more sensitive electronics, namely the Raspberry Pi, IMU, and RF transceiver. Thus, the power system was developed and simulated for this project.

### 2.1.1 Simulation

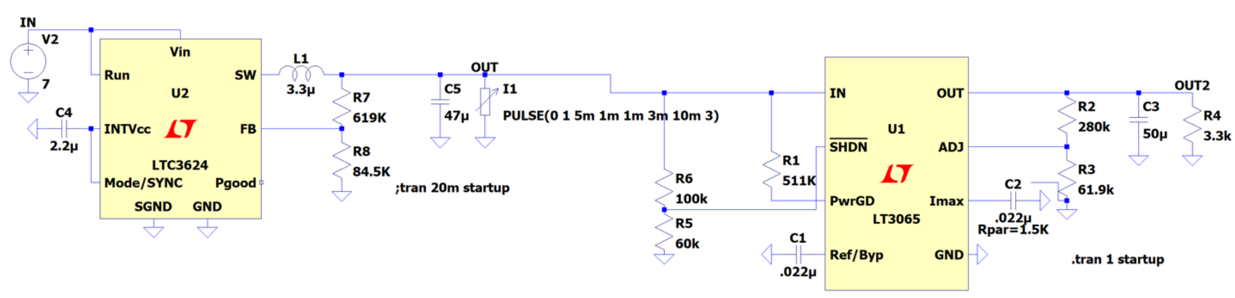


Figure 3: Full System Simulation Schematic

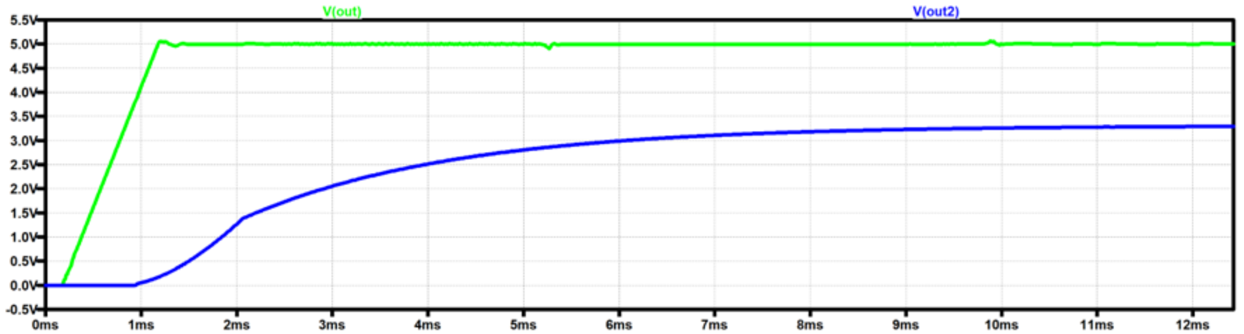


Figure 4: Full System Simulation

### 2.1.2 Motivation and Chip Selection

The central item of this board for communication is a Bluetooth low-energy (BLE) 2.4 GHz MCU, the NRF24L01+. A few different requirements provided the motivation for the selection of this part. One criterion was ease of use and implementation. The NRF24L01 has a simple matching network and overall package dimension. A second criterion for selecting this chip was availability of documentation. This chip also is widely used in the ISM (Industrial Scientific and Medical) band and has a dearth of documentation available for our usage. A third criterion for chip selection was that the radio had to be able to operate within distances of 100 m, as well as transmit position and direction data. An additional requirement was that this chip was required to be used in a "mesh" sort of network. Because the chip utilizes BLE, it is able to connect to a number of different devices and to form a "mesh" topology. Furthermore, it has the ability to transfer up to 2MB/s, and with a transmitting time of 6ms, can transfer around 40 bytes [2] of data per transmit, which is enough to transmit position and instruction data. Finally, the power consumption of the chip had to be fairly low, in order to accommodate power losses elsewhere in the circuit. Because the chip utilizes 3.3V and the maximum transmission amperage is around 20 mA, the power is calculated as:

$$\begin{aligned} P &= I * V \\ P &= 20[mA] * 3.3[V] \\ P &\approx 66mW \end{aligned} \tag{8}$$

Thus, with all these factors considered, the NRF24L01+ was selected. Additionally, the cost per chip was only \$3.5, which is considerably less expensive than all other Nordic RF chips, as seen on various electronic component retailer catalogues.

### 2.1.3 Circuit Design

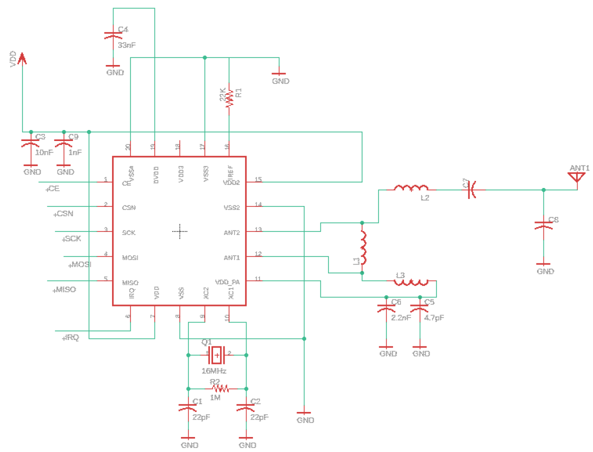

Figure 5: RF Module

The circuit design section of this module followed the application notes as suggested by documentation of the chip. Note that the final circuit appears in 5. Because of the uncertainty of the values necessary for the matching network, the inductors for L1, L2, L3, C5, C6, and C8 were left unlabeled. Initial simulations followed the component values that were suggested in the application notes [2]. Additionally, further analysis was done in order to confirm the results of the simulation. Simulations indicated that at 2.4GHz, there was approximately

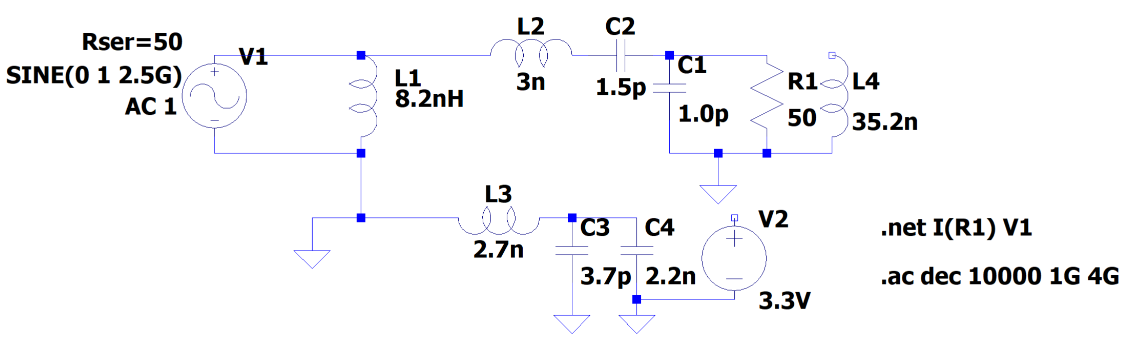


Figure 6: Matching Network

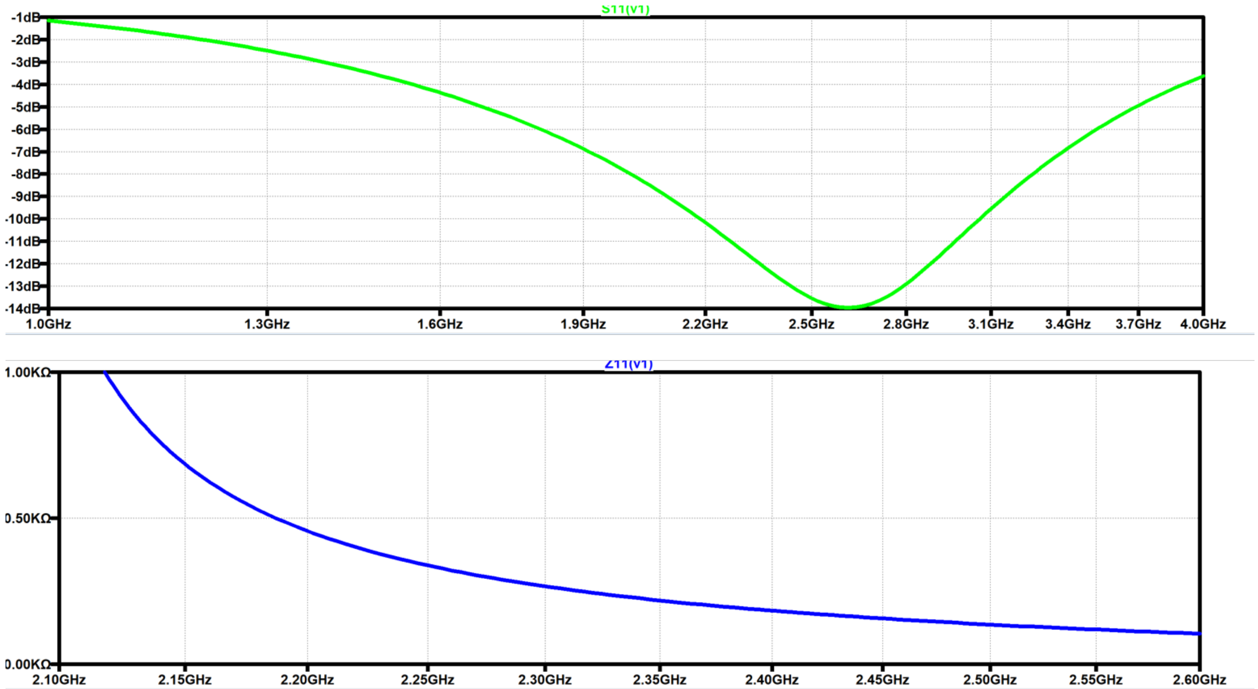


Figure 7: Matching Network Simulation Output

## 2.2 Sensors

### 2.2.1 Camera Distance Estimation

To acquire the position of each robot, we employed camera distance estimation. We placed an AprilTag[5] with a known size at an absolute position. Then, using a calibrated Raspberry Pi Camera module, we can infer the robot's pose estimate relative to the April Tag.

### 2.2.2 Inertial Measurement Unit (IMU)

One of the key sensors of our robot is the inertial measurement unit (IMU). The on board IMU measures various positional and rotational data such as acceleration and rotational changes. We integrated this data to calculate the robots' velocity and relative position estimates. The schematic is in Figure 8 s

Sensor Unit:

| Requirement | Verification |
|---|---|
| Overall:<br>1) Robot can determine position within 10 cm of a 3m x 3m square<br>Camera Unit (software)<br>1) Camera can distinguish position of 4 cm x 4 cm square marker<br>IMU<br>1) IMU must be able to give acceleration data of at least three axes (accuracy of +/- 1 mg of accuracy for acceleration, 1 degree/s<br>2) Data must be relayed back to the base station in under 1 millisecond | 1) Accuracy of DSP algorithm will be determined in software tests - if there are 100 pixels, with 4 of them being the marker, the algorithm should identify it as the marker<br>2) Camera unit<br>a) Camera unit will be tested using DSP algorithm<br>3) IMU<br>a) 2 axis will be tested for acceleration precision<br>b) Data will be transmitted and received in real time by individual module and other host devices<br>i) Software test for how long data takes will be developed |

Figure 8: IMU Circuit

## 2.3 Motors

The requirements for our motors are fairly simple. We needed motors that could carry our payload and also draw relatively little power. We ended up with two brushed DC motors for each robot, each drawing roughly 100mA at 75% operating duty cycle. This was perfect because each motor draws reasonable amount of current, and can drive our robots effortlessly.

## 2.4 Subsystem Diagrams and Schematics

Figure 9 shows the final board schematic of our Pi-Hat product. It fits and connects with Raspberry Pi's GPIO pins directly on top.

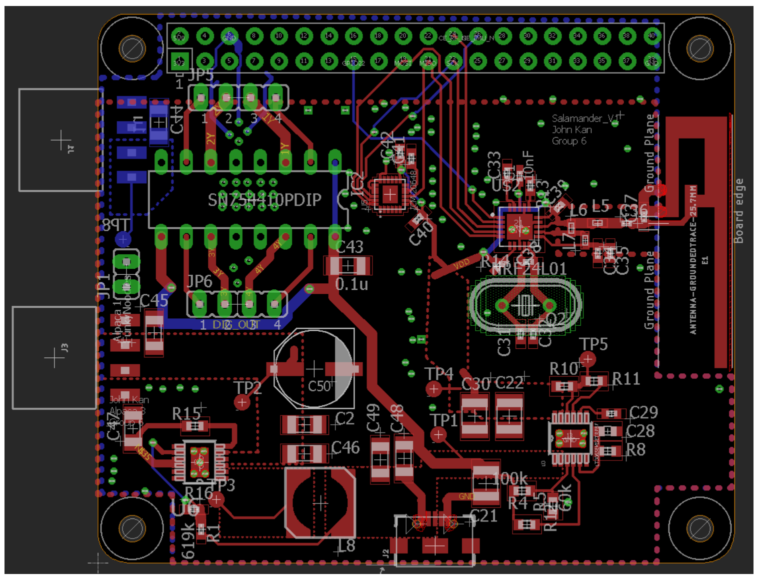


Figure 9: Single Robot Hardware Block Diagram

# 3 Costs and Schedule

## 3.1 Costs

### 3.1.1 Labor

Assuming each of gets paid 30 dollars an hour, and we would on average each spend 27 hours a week. For there are 16 weeks this semester, then the labor of all three of us will be

$$30 \times 27 \times 16 \times 3 = \$38,880$$

### 3.1.2 Parts

Table 1: Part Costs

| Part | Count | Cost for Each |
|---|---|---|
| NRF24L01+ | 1 | \$3.53 |
| LT3065 | 1 | \$4.00 |
| LTC3624 | 1 | \$4.00 |
| RCL Components | 50 | \$0.005 |
| SN75441NE | 1 | \$1.95 |
| Motors | 2 | \$1.00 |
| Robot Kit | 1 | \$15.00 |
| IMU | 1 | \$9.67 |
| Camera | 1 | \$30.00 |
| PCB Orders | 1 | \$1.00 |
| Total | 1 | \$67.78 |

Table 1 illustrates the cost of each parts for a single robot. Each robot will cost \$109.22 to make; since we are making at least three, and ideally four to five. Our project parts in total will cost \$546.10 for five robots.

## 3.2 Schedule

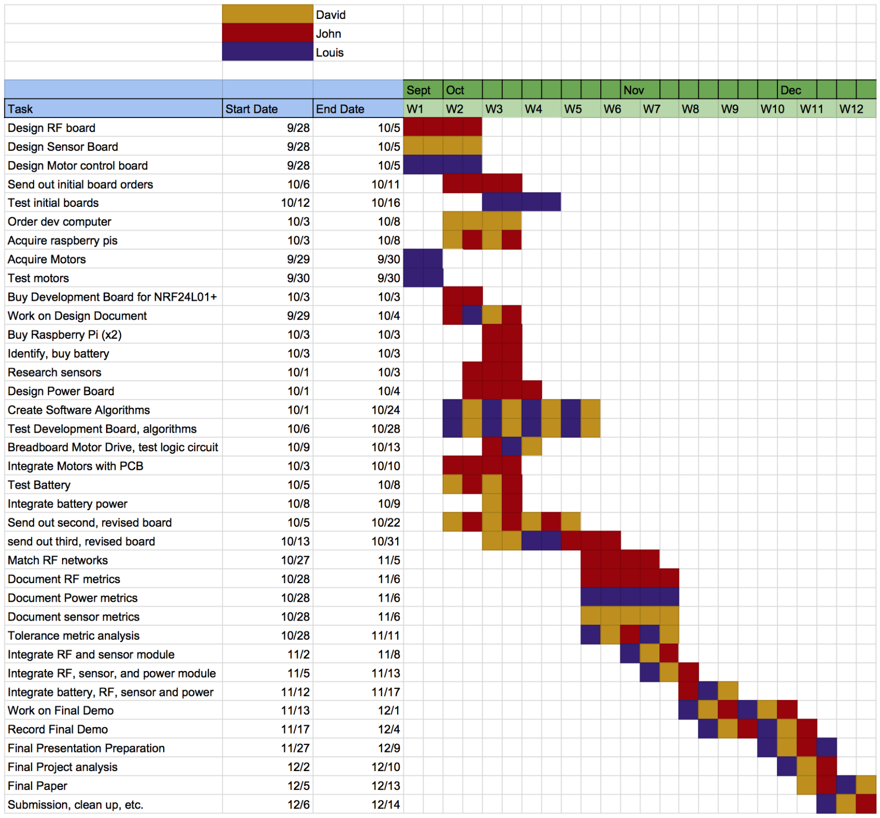


Figure 10: Fall 2018 445 Gantt Chart

Figure 10 shows our the schedule for each task.

# 4 Requirements and Verification

### 4.0.1 Base Station

Table 2: RV Table for Base Station

| **Requirements** | **Verification** |
|---|---|
| The user needs to connect to the base station robot through local network | Connect the Pi to a WiFi network, verify commands are issued to the Pi through SSH |

We are able to connect to our base station and hence satisfy the requirements in Table 2

### 4.0.2 Radio Frequency (RF) Communication Module

Table 3: RV Table for Battery Unit

| Requirements | Verification |
|---|---|
| Module must be able to transmit data to another device | Check if data can be received on another device |
| Must be able to send at 1Mbps | send a large file over and check time required |
| Antenna must provide general directivity between each device | Verify with robots in different directions and orientations |
| Allow multicast and multiple device communication | Send bytes from one robot to multiple others and see if they all received it |
| (1) RF circuit must be able to transmit data – and be matched for each module<br>(2) Antenna should have a 50 Ohm input impedance<br>(3) Matching circuit should bring input impedance of transceiver IC to 50 Ohm | (1) Use Network Analyzer to ensure that the s-parameters (and by extension, input impedance) match – if not matched, adjust the values of the matching network until s-parameters provide maximum power out<br>(2) Check input impedance of antenna using a network analyzer, find s-parameters, and note input impedance |
| Robot needs to relay message between two robots if necessary | Using software to kill a link between two robot, and test to see if the third robot can relay messages. |

To satisfy the requirements shown in Table 3, we check if two robots equipped with our RF chips can transmit and receive data. We also check if one robot can send data to multiple robots.

We check if the robots can still communicate with each other using an intermediate node by first gradually moving one robot away from another, until it cannot be reached. Then we move the third robot between to see if the two original robots can still communicate with each other. This test failed, we think it was a problem with the software driver that claimed to be able to achieve this functionality, but ultimately, did not.

In order to verify the power loss and the input impedance, the Network Analyzer was utilized. The simulation results noted in Figure 7 were tested via the verification procedures, and the S-parameters were found for three different inductance values used to find the optimal input impedance. Additionally, the input impedances, derived from the S-parameters utilizing the relation

$$Z_{in} = 50 * (1 - s_{11})/(1 + s_{11})$$

were also noted [6]. Table 4 indicates the s-parameter and matching input impedance and Figure 11 indicates the output impedance at 2.4 GHz for various inductor values. Note that none of these values were able to fully match 50Ω. This figure indicates that we were not able to meet the particular requirement, however, the S parameters indicated that the gain was consistent and still allowed most of the power to be output. Note that Figure 12 indicates that there is an approximate loss of 10 dBm. Note that each of the peaks has an approximate output of -20 dBm, with a -10 dBm offset. Hence, a 10% power loss indicates that the converter was still efficient enough to transmit information.

Table 4: Real S-Parameters and Impedances at 2.4GHz

| **Inductance[nH]** | $S_{11}[dB]$ | **Impedance** $[\Omega]$ |
|---|---|---|
| 3.3 | -15.649 | 69.76407889 |
| 3.9 | -7.2975 | 125.945844 |
| 4.3 | -4.00 | 220.97138638 |

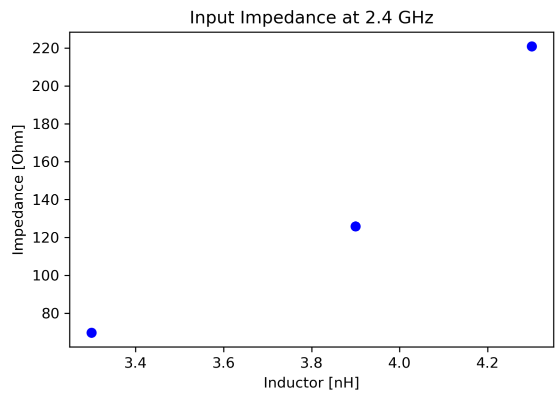


Figure 11: Impedance at 2.4 GHz

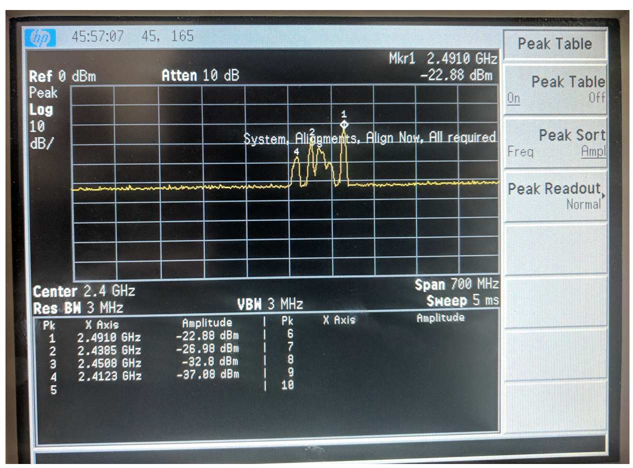


Figure 12: Full System Simulation

### 4.0.3 Power Management

Table 5: RV Table for Power Management

| Requirements | Verification |
|---|---|
| (1) Distribute 3.3V (+/- 5%) power to sensor and RF modules<br>(2) Ensure all rise times are less than 50 ms (3) Ensure that shutdown will occur within 50 ms | (1) Verify voltage output and ripple with oscilloscope readings<br>(2) Verify with an oscilloscope that the rise time at start up is 50 ms – if not in 50 ms time frame, turn off<br>(3) Induce a shutdown (pull out power source) and note the shutdown time |
| (1) Distribute 5.1V (+/- 5%) power to Raspberry Pi (2) Ensure all rise times are less than 50 ms (3) Ensure that shutdown will occur within 50ms | (1) Verify voltage output and ripple with oscilloscope readings<br>(2) Verify with an oscilloscope that the rise time at start up is 50 ms – if not in 50 ms time frame, turn off<br>(3) Induce a shutdown (pull out power source) and note the shutdown time |

Note that Figure 13 indicates that the turn on times and ripple voltages for the 5V and 3.3V systems were verified.

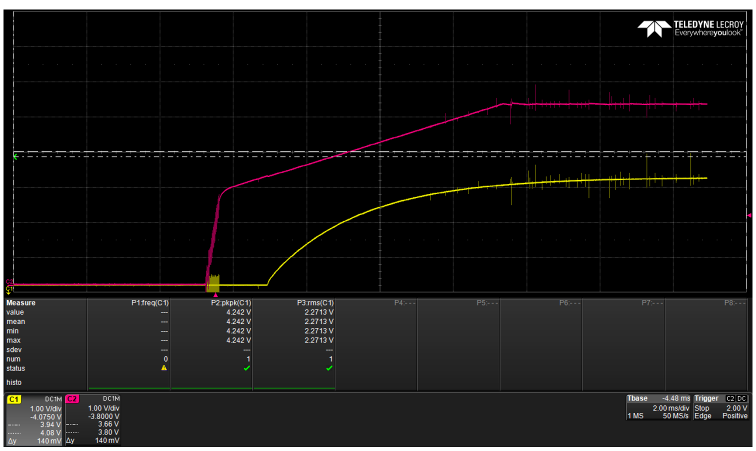


Figure 13: Power Management verification

### 4.0.4 Inertial Measurement Unit

Table 6: RV Table for IMU Unit

| **Requirements** | **Verification** |
|---|---|
| Detect acceleration and orientation of robot and relay data back to Raspberry Pi on two axes up to 1 g of force | Run programs on Pi to take IMU reading from GPIO and verify position and acceleration estimates |

To measure the accuracy of our IMU chip, we took our robot on an elevator ride, from the second floor of the ECE building to the first floor, up to the fifth floor, and then back down to the second floor. We placed the robot inside the elevator while it records its acceleration data shown in Figure 14. From this elevator experiment, we estimated that the floor level of the ECE building is roughly 19 meters tall.

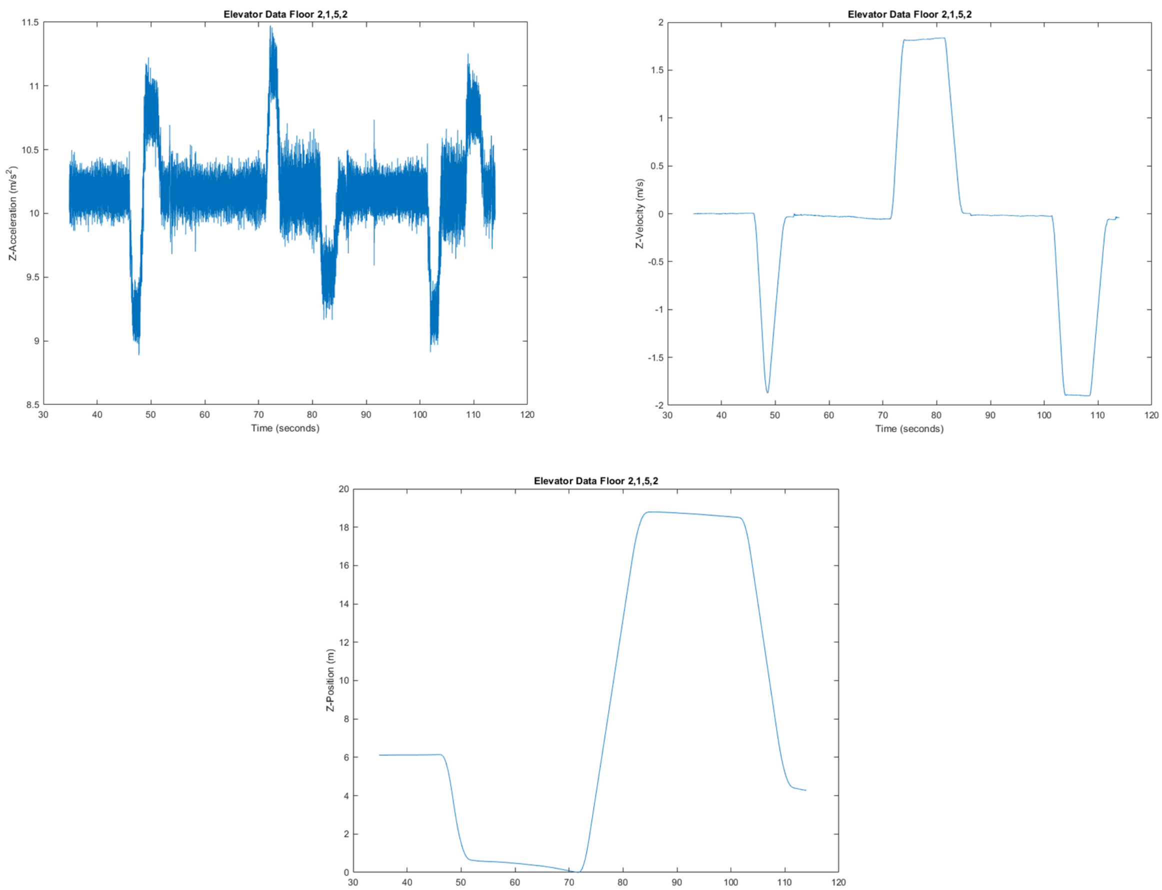


Figure 14: IMU raw acceleration data interpreted to velocity and position information. Top left is raw acceleration data, top right is integrated velocity data, and botom is position data.

### 4.0.5 Camera Distance Estimation

Table 7: RV Table for Camera Distance Calculation

| **Requirements** | **Verification** |
|---|---|
| Camera on robot needs to calculate its distance to any arbitrary marker. | Place the marker at a known distance away from the camera and verify that the camera estimates a reasonable value. |
| Rotate on its spot to calculate distances with multiple markers. | See below |
| Infer its absolute position given distances to several markers. | Two markers will be setup at known distances from the robot in two directions, the robot will rotate and infer its absolute location. |

To verify requirements for our camera distance detection, we placed a tape measure right next to the camera, and our April Tag at various points directly away (along the Z axis) from the camera and calculated its percent error in estimation.

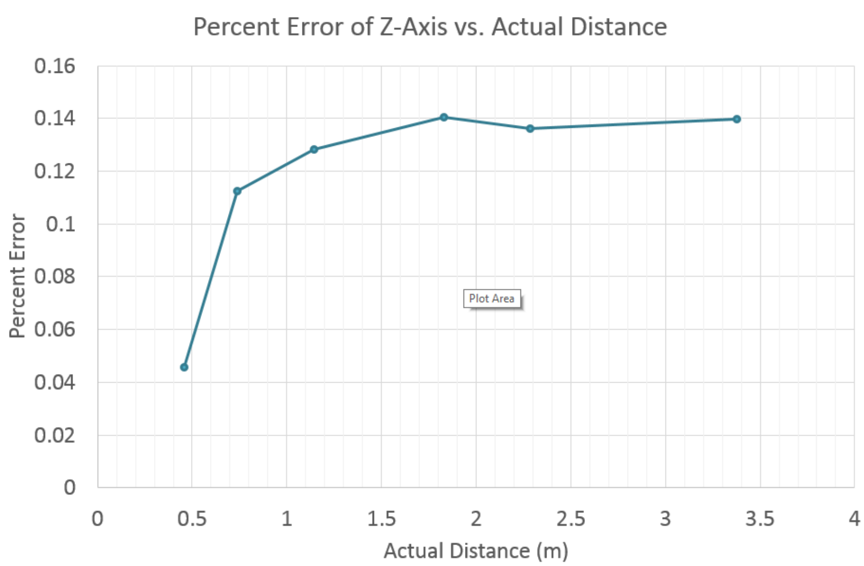


Figure 15: Z-Axis percent error for camera distance estimation

Figure 15 shows Z-axis percent error for camera distance estimation. The error is about 5% when the April Tag is about 1m away. However, this error grows and caps off at around 15% for farther distances.

### 4.0.6 Motor Control (5 points)

Table 8: RV Table for Motors

| Requirements | Verification |
|---|---|
| Control Motors using two H-bridge drives | Feed various PWM duty cycles into the SN754410NE chip and observe motor speeds driven by a power supply. |
| Motors need to provide enough torque for the robot | Install motors on the robot and see if the robot moves |
| Motors need to operate at 5V (±10%) | Use a power supply to test that motor work at 5V |

To verify the motor control and motor units, the soft PWM of the Raspberry Pi was utilized and the output was tracked using an oscilloscope (Figure 16). Additionally, the motor operation and control was tested (Tab. 9) to ensure that the motors could operate at around 5 V. Note the output waveforms in Figures 16 and 17 all indicate the ability to operate at 5V. Hence, the motor requirements were verified.

Table 9: Motor Control Operation Levels

| Duty% | Input Voltage [V] | Input Current [mA] |
|---|---|---|
| 50 | 5 | 210 |
| 60 | 5 | 200 |
| 70 | 5 | 200 |
| 80 | 5 | 180 |

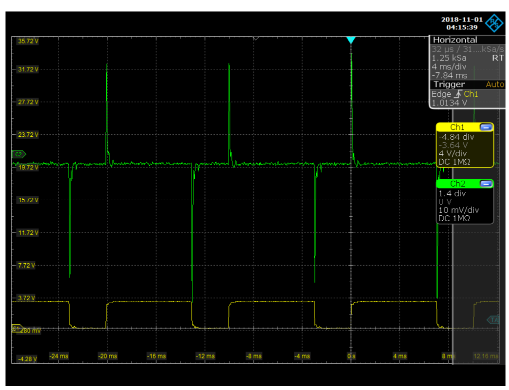


Figure 16: PWM and Motor Currents

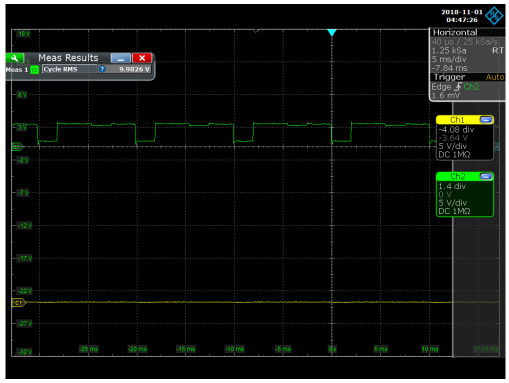


Figure 17: Current Output with PWM Signal In

### 4.0.7 Battery Unit

Table 10: RV Table for Battery Unit

| Requirements | Verification |
|---|---|
| Supply enough power (8W) for approximately 30 minutes (4 Wh) | (1) Use a resistive load to discharge batteries until completely drained<br>(2) Analyze power measurements of voltage and current using an oscilloscope to battery<br>(3) Note the point at which the voltage drops lower than 20% of rated voltage<br>(4) Verify that the time to discharge was 30 minutes |

Our 4 pack AA batteries can provide 6V continuously, as verified by multimeter reading. From our experimental results, our batteries can provide roughly 2 hours of runtime on our robots while it is mostly idle and intermittently doing some computer vision algorithms and transmitting data.

### 4.0.8 Software System

Table 11: RV Table for Battery Unit

| Requirements | Verification |
|---|---|
| Interface with RF module | Send and receive bytes to and from another robot |
| Take in raw data from IMU and camera units | Sanity check on values read in |
| Output correct soft PWM signal from GPIO pins at 100Hz | Use oscilloscope reading to verify |

To verify our software system is fully integrated, we ran all the software modules together at the same time and see if the robot can respond to our remote commands. Our software runs continuously and communicates seamlessly with other robots.

### 4.1 Completeness of Requirements

All of our requirements are met except for one. We could not get our RF chips to relay information between robots. We use an online software library that claims to have implemented this function for us, when finally verifying this requirement on our own robots, we did not get it to work as desire. This means that when one robot goes out of range with the master base station, they are still able to maintain connection through an intermediary robot that relays information. Though we did not get this to work, it is only a minor requirement that is not crucial to our final working of the project.

## 5 Conclusion

### 5.1 Accomplishments

Our project has turned out exceptionally well. We satisfied all of our major requirements. Each of our hardware and software sections integrate well together. Specifically, our power section can sufficiently provide power to both the Raspberry Pi and various chips on our PCB. Our RF chip and our matching network function well together to reliably transfer data between each devices. Our IMU sensor is able to collect fairly accurate data that allows for real world measurements. Finally, our motors are able to function according to commands and drive our robots.

### 5.2 Uncertainties

Soldering our boards has been fairly challenging since our electrical parts were all very small. We had to learn how to use soldering paste and the reflow oven. Furthermore, we went through five iterations of different PCB designs to finally have everything working together on one single contiguous board that fits on top of the Raspberry Pi.

Close to the demo deadline, we realized from sources in the relevant industry that our original IMU chip ICM20648 had troubles responding to its software drivers. So we switched to an older chip from the same company, the MPU9250, which fortunately has the exact pin size and configurations with the previous chip. The MPU9250 is much easier to use and worked as expected immediately after we soldered it on.

### 5.3 Future Work and Alternatives

Since we only had in total a few weeks to work on this project, we only implemented a subset of our complete envisioned features.

A key aspect of our project is the inclusion of various sensors. In our limited time frame, we only implemented two sensors, the IMU and the camera distance estimation modules. We hope to integrate more sensors and actuators such as IR sensors or mechanical claws to grab objects.

In addition to sensors, we also would like to spend more time to produce a PCB that optimizes our RF transmission range. Currently, it is unstable - various PCB has different transmission distances that vary by quite a margin. To make our product as robust as possible, we need improve the quality of our RF transmission range.

In our current design, we employed software PWM and brushed DC motors for moving the robot around.

This method is not very precise due to large inaccuracies of software PWM and mechanical imperfections of brushed motors. We can implement the much more accurate servos to drive our robots if we have more time.

Although we already employ ROS to make our software more modular, we would like to build a much more advanced and user-friendly software framework so our users can easily expand on our original functionality.

## 5.4 Ethics and Safety

Since we used RF bands for communications, we made sure to not violate FCC regulations on the frequency and transmitter power. Because we bought an MCU using a well-established Bluetooth technology specified in IEEE 802.15 [7], our RF usage is legal, safe, and will conform to the IEEE Code of Ethics Article 1.

We used NiMH nine-volt batteries to power our systems. NiMH batteries, according to Energizers guide [8], are cheap, made of environmentally friendly materials, only contain mild toxins, and are recyclable.

Additionally, this project had a variety of moving parts. In order to comply with the IEEE code of ethics part 9, "to avoid injuring others"[9], our robots can be shut down easily by turning off the power.

Additionally, each robot had a number of sensors that will collect data about the outside world. All data can be used in a way that is harmful to other people, and since we used cameras for object detection and local position sensing, this camera may have data that could be used in a malicious manner. To conform to Article 9 of the IEEE Code of Ethics,"to avoid injuring others...[or] their reputation..."[9], we did not permanently keep these camera files.

# References

[1] "SeaPerch ROV Competition," Web page, accessed September 2018. [Online]. Available: https://www.seaperch.org/index

[2] "nRF24L01+ Single Chip 2.4GHz Transceiver," Nordic Semiconductor, accessed September 2018. [Online]. Available: https://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24L01

[3] "LTC3624," Linear Technology, accessed November 2018. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/36242fd.pdf

[4] "LT3065," Analog Devices, accessed November 2018. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/LT3065.pdf

[5] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2011, pp. 3400–3407.

[6] "Performing Impedance Analysis with the E5061B ENA Vector Network Analyze," Keysight, accessed September 2018. [Online]. Available: http://literature.cdn.keysight.com/litweb/pdf/5991-0213EN.pdf

[7] "IEEE 802.15 Wpan Document Archive," Web page, accessed September 2018. [Online]. Available: http://grouper.ieee.org/groups/802/15/pub/Download.html

[8] "Nickel Metal Hydride (nimh) Handbook and Application Manual," Web page, accessed September 2018. [Online]. Available: http://data.energizer.com/pdfs/nickelmetalhydride_appman.pdf

[9] "IEEE Code of Ethics," Webpage, IEEE, accessed September 2018. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html