Electric Longboard Safety Suite

ECE 445 Final Report - Spring 2021

Project # 17

Pouya Akbarzadeh, Alexander Krysl, Patrick Stach

Professor: Jonathon Schuh

TA: Evan Widloski

Abstract

The following document details the motivations, development, and results of the Safety Suite for Electric Longboards project. Discussed here is 1) how commercially available electric longboards fail to discern whether the user is on the board or whether the motorized wheels have experienced loss of traction, 2) the design & development process over the course of this project to mitigate the aforementioned hazards, and 3) the final results of the efforts & innovations made in the pursuit of personal electric vehicle safety.

Contents

1 Introduction

1.1 Overview

Electric Skateboards and Longboards have skyrocketed in popularity for personal transportation in urban cities & towns [1]. Their nimble and speedy characteristics allow users to easily navigate several miles of congested vehicle or foot traffic yet are small and lightweight enough to be carried around indoors. Despite their value as a useful transportation and recreational tool, it is clear from the relatively simple motor and user interface designs that nearly all consumer electric skateboards and longboards lack seemingly paramount safety features. Note that we will use the terms "electric skateboard" and "electric longboard" interchangeably for the rest of this document.

1.2 Problem

To begin, no consumer electric longboard attempts to discern whether the user is physically on the longboard or has jumped/fallen off. This unfortunately allows for highly dangerous scenarios where a user falls off the electric longboard while inadvertently maintaining the throttle. These electric longboards may then accelerate out of control — possibly toward pedestrians — having lost the weight of the user. Next, no consumer electric longboard includes mechanisms that attempt to mitigate wheel-slip, which is the result of the user providing too much throttle input, more than what the electric skateboard wheels' traction can handle. Examples where this can occur include when traveling over uneven/gravel-like terrain, performing extremely hard turns, or even applying severe braking throttle while going at speed, and more. In each of these cases, the electronic speed controllers will allow the powered wheels on the electric skateboard to overcome static friction, continually "slipping" across the ground. Wheel-slip is a significant safety risk as it dissolves the user's control of the board. It may upset the balance of the user, it may cause the electric longboard to slide out from under the user, and it may prevent the user from braking/slowing down the electric skateboard appropriately risking a collision. While the statistics of electric longboards have not been formally studied, in the case of electric bicycles (another, more popular form of personal electric vehicle), roughly 30% of accidents were a result of wheel-slip — the most common among accident types by a significant margin [2].

1.3 Solution

To help remedy these safety concerns, we have implemented a twofold plan. Firstly, we have integrated weight sensors within the deck of the longboard. Second, we have custom-developed wheel-revolution sensors across both front, unpowered skateboard wheels. For the rear, motorized wheels, our motor controllers can be queried for rotational velocity information. We then perform calculations on the sensor data to identify whenever wheel slip or user-ejection has occurred; when it is identified, we manipulate the throttle commands to the motor controllers to allow us to reduce motor power to the responsible wheel(s). Whenever we identify that the user has fallen off of the board, we ensure that the electric longboard is not allowed to accelerate more than a typical walking pace. Further, whenever we identify that the rear, powered wheels are spinning considerably faster than the front, unpowered wheels, we actively attempt to regain traction by disengaging/dampening the throttle command to the offending motor(s) until the difference is minimized.

1.4 Requirements

The following is a list of the high-level requirements we declared at the beginning of our project:

1. Our electric longboard is able to sense if the weight atop the board is significantly less than that of a normal user (a threshold value of 40 pounds); when sensed for 0.5 seconds or more, motor power is reduced to a maximum speed of 5 mph or less.

2. Our electric longboard is able to measure/record the rotational speed of each of the front, unpowered wheels on the longboard to a tolerance of $\pm$ 5% of the true value.

3. Our electric longboard will sense whether the powered wheels are slipping in the opposite direction of the unpowered wheels (as would be the case under extreme braking); when sensed, our electric longboard will automatically dampen the reverse-throttle value until traction is recovered.

At the end of this project, we have demonstrated that our custom electric longboard meets all of the above requirements. This is further discussed in Section 3.

2 Design

- The **Board Control Subsystem** is responsible for receiving transmissions from the remote, taking data from the sensing subsystem, and commanding the electronic speed controllers. Based on these inputs, the board control subsystem will command the electronic speed controllers to power the motors accordingly.

- The **Board Drivetrain & Power Subsystem** is responsible for supplying power to all electronic components on the board, and provides the means to control the rear motors on the electric skateboard. This was accomplished by using open-source or purchasing commercial parts.

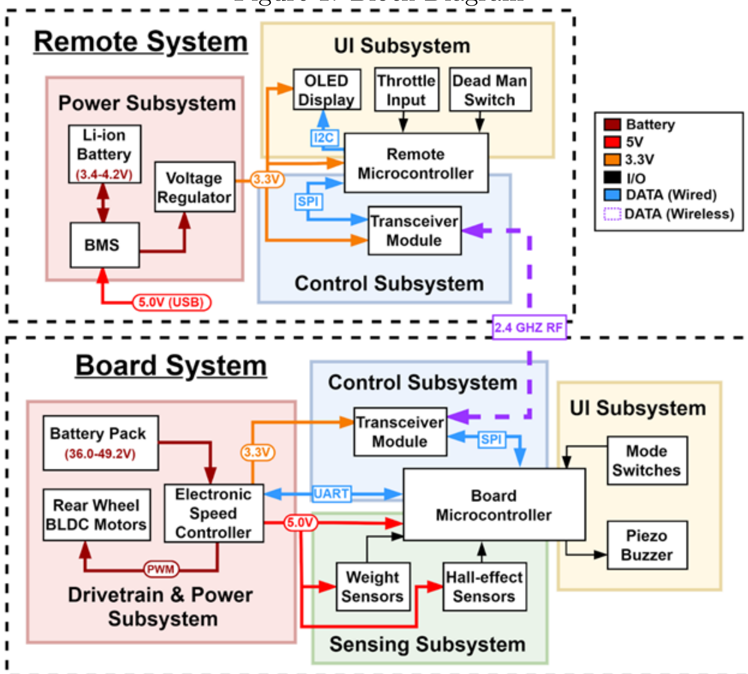
- The **Board User Interface Subsystem** is responsible for allowing the user to toggle on and off the various features of the electric longboard, such as traction control, as well as audibly notifying the user of unexpected conditions, such as if the remote is no longer communicating.

- The **Remote Control Subsystem** is responsible for passing the user interface input data to the Board Control Subsystem; this specifically includes the input throttle thumbwheel and the throttle/deadman switch. The Remote Control Subsystem must package this information into wireless radio communication messages to the Board Control Subsystem.

- The **Remote User Interface Subsystem** is responsible for allowing the user to control the electric skateboard, which may be adjusted by means of our safety suite.

- The **Remote Power Subsystem** is responsible for supplying power to all electronic components on the remote, most importantly the microcontroller and the transceiver module.

Figure 1: Block Diagram



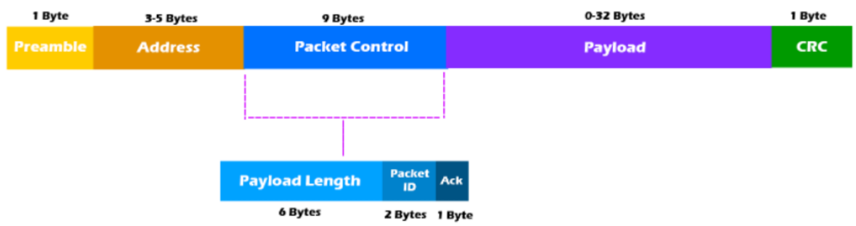
2.1 Remote & User Interface

While approaching the remote design, we had considered two options - whether to heavily modify a Nintendo Wii Nunchuck or to design our own from the ground up. From our initial remote designs, we quickly deemed that our trigger and thumbwheel design was much more suitable to electric longboard throttle control than an extended thumbstick. The X-Axis (left & right) motion of the Nunchuck's thumbstick would be not utilized in our project and may confuse inexperienced riders.

To read the throttle wanted from the user, we used a hall sensor & magnet design. One magnet placed inside the thumbwheel, and two hall sensors are integrated into the remote housing. The primary hall sensor is centered along with the magnet and is used to determine the magnitude of the user's throttle input. The secondary hall sensor is offset towards one direction and is used to determine the direction of the user's throttle input. With this design, we could achieve higher precision throttle input as compared to the Nunchuck option. Additionally, a tactile trigger was included for the user to press to engage the motors. Should a user inadvertently drop the remote, the trigger would be released and the throttle would automatically be disengaged.

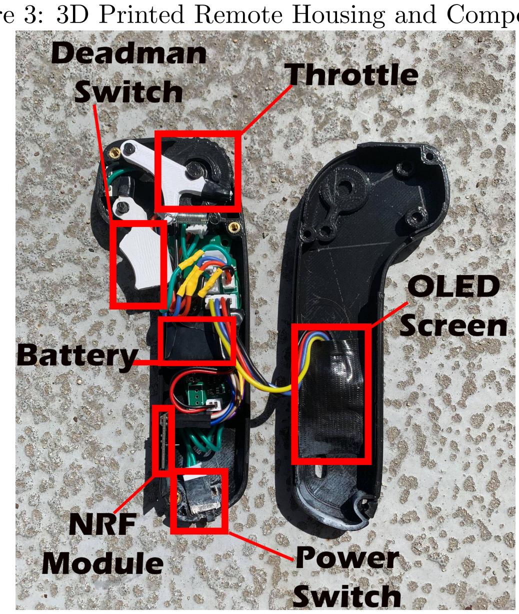
For wireless communication from the remote to the longboard, our module of choice was the NRF24L01+. This particular module uses the 'Enhanced ShockBurst Protocol' [3], which offers features such as Cyclic Redundancy Checks, variable payload sizes, and its ability to send/receive acknowledge messages. All these features allowed us the remote to communicate to the longboard and back and ensured that any errors in the transmission are detected and ignored. A layout of the message structure can be found below.

Figure 2: Enhanced ShockBurst Protocol message structure



Our original remote enclosure designs were roughly based on the open-source remote design called the Firefly Eskate Remote [4]. As seen in the picture below, it is clear that we were working with a very small amount of space. Fitting in all the components for the remote was challenging; however, shaping the PCB like the remote, and increasing the length of the remote allowed us to more comfortably fit all of our components inside.

Figure 3: 3D Printed Remote Housing and Components

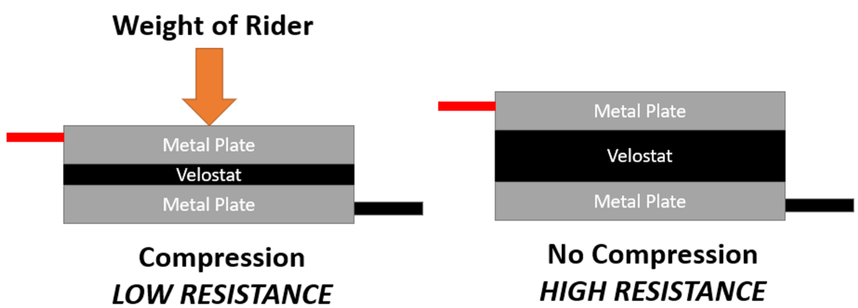


Another reason we decided to go on the route of 3D printing our remote was due to the fact that we wanted to fit an OLED screen in our controller as well. This would have been almost impossible due to the shape of the Wii Nunchuck. The OLED screen allowed us to display plethora of information in an easy manner. As part of the nature of making our remote completely wireless we need to use some sort of battery to power the components on the board. We chose a 500mAh li-ion battery as it has a high energy density in order to minimize the space it takes up on the remote, in addition to allowing for recharging. There are four sub-circuits on our PCB as part of our battery management system. This includes the battery charging, load sharing, under-voltage monitoring, and 3.3V regulator. The battery charging circuit safely charges our li-ion battery when 5V USB power is applied. The load sharing circuit will "OR" the battery and USB supply so that when USB power is applied: the battery will strictly charge and not source any current, while the USB power will also power the remote circuitry. The under-voltage monitor detects whether the li-ion battery voltage has dropped below 3.4V and when so will disable the 3.3V regulator, cutting power to the remote. The 3.3V regulator converts whatever supply voltage outputted from the load sharing circuit to regulated 3.3V power.

2.2 User Ejection Detection

To detect whether the user is actively atop the board, we have integrated weight sensing pads on the topside of the longboard deck. While the placement of the pressure pads has changed from our original plan, the fundamental design of the pads has not. To sense weight, we have utilized the velostat material. Velostat has the property that its conductance nontrivially increases with pressure. Our design simply sandwiches velostat between two thin, flexible metal sheets. The metal sheets are conductive, and as the user applies pressure to them by standing atop the board, the electric resistance of the pad decreases considerably.

Figure 4: Visual representation of weight sensing pads with and without compression



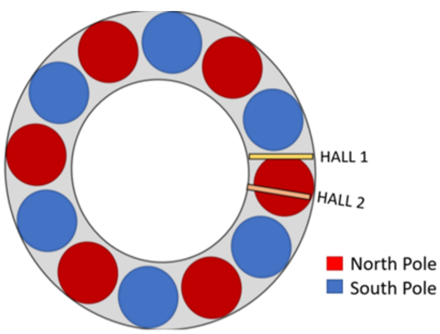
To implement this as an analog signal for our board microcontroller, a voltage divider circuit is utilized with a constant pull-up resistor and the pressure pad as the pull down resistor.

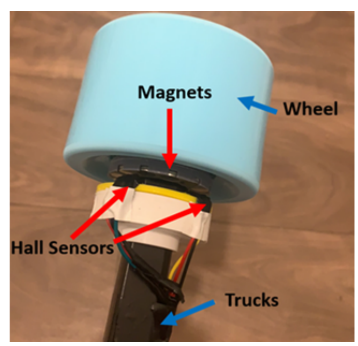
From initial observation of the velostat resistance under pressure, we found that un-pressed the sensors read around 10k ohms and pressed they read 20 ohms. The geometric mean of these two numbers is what will be used for the pull-up resistor, and this is roughly 440 ohms. We used a 470 ohm resistor as this was a value we already had on hand. After assembly and testing on the deck, we found that the low threshold of resistance is actually closer to 5 ohms, but using a 470 ohm resistor as calculated above worked well enough for our purposes. Our board microcontroller reads the output voltage under an 10-bit analog-todigital converter to gauge the weight atop the board. From testing, we determined that our ADC reads roughly 650/1023 (3.17V) with no user and 5/1023 (0.02V) with a user on the board. From this data, the final threshold dividing value was set to 300/1023 (1.47V). Our initial plan was to implement weight sensors within the trucks of the longboard. However, due to the pressure from the bolts that attach the trucks to the deck, we were unable to properly mount the trucks without saturating our pressure pads. Thus, we had to revamp our original idea. We found much greater success by placing the pads on top of the board, as seen in the picture below. This allowed for proper analog readings as well as higher accuracy by covering more surface area on the board.

2.3 Front Wheel RPM Sensing

Brushless DC (BLDC) motors utilize Hall sensors to detect the position and polarity of the magnetic field from the permanent magnet in the rotor. However, the front wheels are unpowered and thus have no magnetic field to detect. In order to circumvent this, we decided to emulate the design of a BLDC motor. We started by modifying the wheel to act as a rotor with permanent magnets inside, while placing latched hall sensors are mounted on the trucks, which is analogous to what would be the stator. This was done by modifying and 3D printing CAD files available online [5, 6]. Pictured below is a visual representation of this.

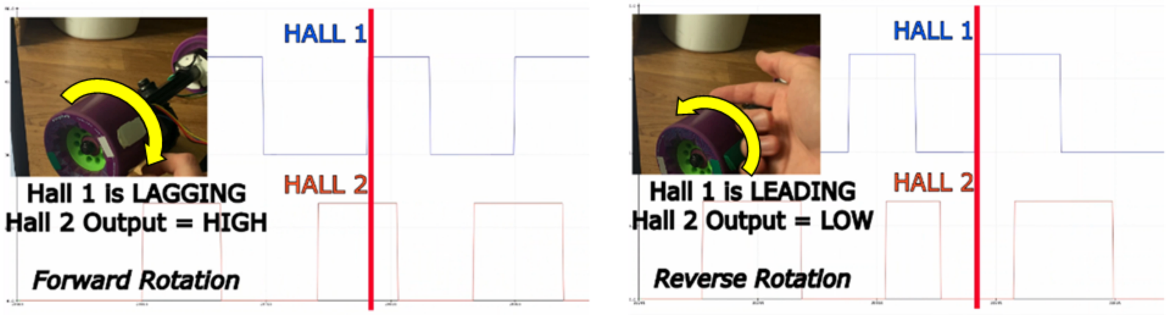
Figure 5: Diagram and assembly of magnetic ring on a free-spinning wheel with Hall sensors attached to the trucks





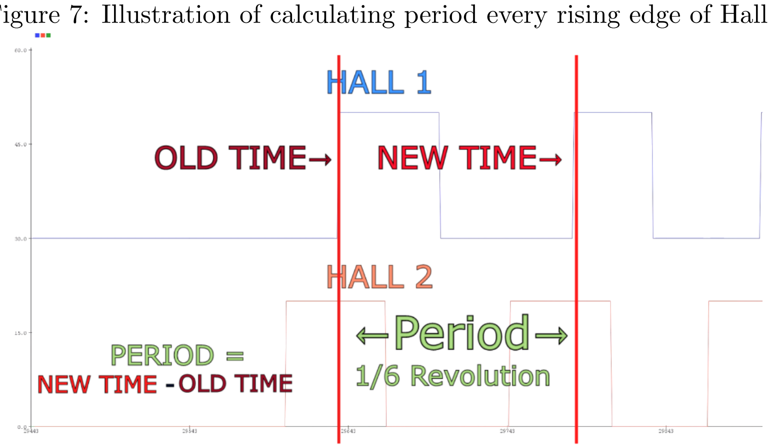
For the placement of the hall sensors, rather than following the configuration of 3 hall sensors spaced 120 degrees apart we drew inspiration from quadrature encoders. Quadrature encoders only have two hall sensors and this minimizes the amount of components needed. Placing the hall sensors at a distance that will result in output waveforms one quarter of a period away will result in leading/lagging waveforms that will allow us to determine rotation direction. As seen below just by reading the state of Hall 2 on the rising edge of Hall 1 gives us enough information to determine direction of rotation. On the rising edge of Hall 1 we know that if Hall 2 is HIGH then we are rotating in the forward direction, otherwise if Hall 2 is LOW then we are rotating in the reverse direction.

Figure 6: Determining rotation direction from Hall 1 and Hall 2 waveforms



The data we need to calculate RPM from these waveforms is the time between two rising edges of the Hall 1 signal, which is the time for the wheel to rotate ⅙ of a revolution. The image illustrates what this calculation looks like using Hall sensor waveform data. In order to do this, we will be using an interrupt service routine (ISR) that executes on the rising edge of Hall 1 to calculate this time, and this time captured will correspond to the time it takes for ⅙ of a revolution of the wheel.

Figure 7: Illustration of calculating period every rising edge of Hall 1



If we have this data we can convert to RPM with the equation below, keeping in mind that when we get the current time from the processor it is given in microseconds:

$$RPM = \frac{60}{6 \times 10^{-6}(\text{NEW TIME - OLD TIME})} \tag{1}$$

To minimize the impacts of executing the ISR interfering with normal operation, we need to make sure we are as concise as possible writing it. To ensure fast operation, we will be using direct port manipulation as opposed to something like the digitalWrite() function. In our interrupt routine, first we store the current time in "New Time". Next, we set the Rotation Direction flag = Hall 2 State. We set Δ Time = New Time - Old Time, and this Δ Time is stored in an averaging array. Lastly, we increment the index and set Old Time = New time. The averaging array contains the last 5 Δ times calculated, and we are using this array as a circular buffer that effectively replaces the oldest piece of data with the current data. To save processor time, we will be averaging this array at the same time we calculate RPM, rather than inside the ISR.

2.4 Throttle and Traction Control

Our throttle control application can be broken down into two parts. A flowchart depicting the first section can be seen in Appendix D. This first section depicts how the board microcontroller determines whether the user is actively riding the electric longboard. We assume the user is actively riding the longboard when the following conditions are met: 1) the board has received a transmission from the remote within the last second, 2) the weight sensing subsystem detects a rider is on the board, 3) the user has depressed the throttle/deadman switch. If at that point traction control has been disabled, our microcontroller will simply pass the throttle value given from the user on the remote's thumbwheel. Else, we enter our traction control algorithm elaborated on later in this section.

The cases where the user is not actively riding the longboard must be accounted for as well. Our first check is whether the remote has communicated with the board microcontroller within the last second; if this is not the case, the remote is considered disconnected. If the rider is still detected atop the board, the board will coast so as not to buck the rider. If there is no rider, the board microcontroller sends the stop command to the motor controllers. Next, if the remote is connected, the board microcontroller will check if the rider is detected; if this is not the case, the board microcontroller will check if the user is applying throttle. If so, the board microcontroller allows the board to accelerate no faster than to a walking pace, ensuring that the longboard may not accelerate out of control towards pedestrians but allows the user to conveniently reposition the longboard if need be. If instead there is no user throttle, the board microcontroller sends the stop command to the motor controllers.

If the board microcontroller determines that there is an active rider as described above, and traction control is enabled, the traction control algorithm is applied to each rear, motorized wheel individually. A flowchart depicting this process is shown in Appendix D. The initial steps of our traction control algorithm are to calculate the front wheel RPMs as outlined in Section 2.3 as well as determine if the user is braking. We assume the user is braking when the user's throttle input is in the opposite direction of the angular velocity of the front wheel.

If the user is braking, the board microcontroller will enter our braking traction control algorithm. For the purposes of explanation, we will initially ignore the feedback section of the algorithm (in other words, the wheel in consideration has not recently lost traction under braking) and return to it later. In our braking traction control algorithm, the first check is to ensure that the rear wheel RPMs are above a minimum threshold of 350 RPM (roughly 3 miles per hour for our longboard). At very slow motor speeds, traction control is ignored to prevent unnecessary throttle manipulation. Else, the board microcontroller will check if the rear wheels are spinning in the opposite direction of the front wheels. If so, wheel-slip has been detected. The algorithm then acts as a 2-Step controller with previous state feedback, releasing the offending motor until either the user is no longer braking or the rear wheel is now spinning in the same direction as the front.

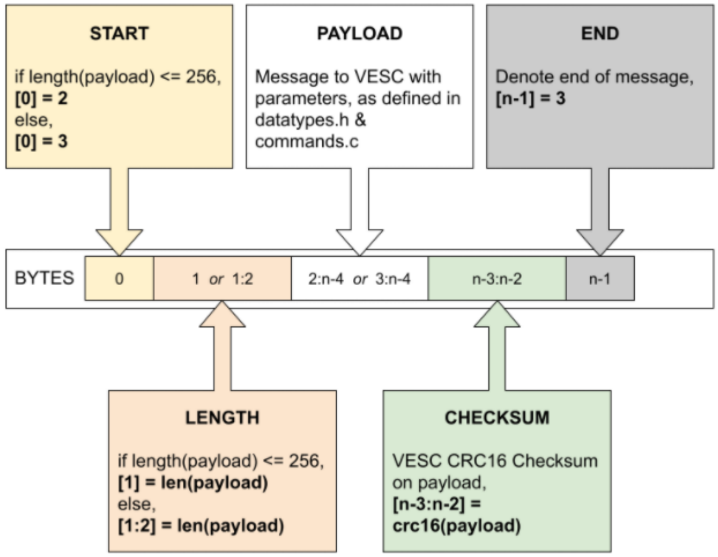
Otherwise, if the wheel is not slipping, the board microcontroller passes the user input throttle without manipulation.

If the user is not braking, the board microcontroller will enter our acceleration traction control algorithm. There are three output states - passing full user throttle, neutralizing throttle, or dampening the throttle. The microcontroller choses the output state based on the difference between the motorized wheel RPM and the slowest front wheel RPM. If the motorized wheel is slower than or equal in angular velocity to the front wheel, throttle is passed without manipulation. Else if the motorized wheel is faster than the front wheel by a margin of 350 RPM (roughly 3 miles per hour), the throttle is neutralized to the offending motor. Otherwise, the throttle will be proportionally dampened based on the RPM difference - the greater the difference, the more dampened the throttle to the corresponding motor will be.

In developing the traction control algorithm, we considered various possible designs that could meet the requirements of detecting and attempting to mitigate wheel slip. We ultimately decided on the 2-Step controller for braking for its relative simplicity and demonstrative qualities. From internal testing, the microprocessor we chose for the main controller would not be fast enough for our real-time system with an overly complex algorithm. With a faster microprocessor, we would heavily consider implementing a Proportional Integral Differential Controller that would more robustly mitigate wheel-slip, with the error being the difference in front to rear wheel RPMs. Further, the 2-Step controller makes it visually clear when the algorithm is attempting to regain traction or not, which highlights the board microcontroller's ability to sense that wheel-slip is occurring for our third high-level requirement. Further, we determined the threshold values for our traction control algorithm from informal testing - 350 RPM was observed to be the slowest threshold speed that would not kick traction control unnecessarily.

Finally, to send the throttle commands, our board microcontroller follows the UART communications protocol of our motor controllers. For our custom electric longboard, we used a pair of Vedder Electric Speed Controllers, designed by Benjamin Vedder [7] to power our motors. VESCs communicate over UART using packets with the following format [8]: One Start byte (0x02 for 256 byte or less payloads, 0x03 else), One or Two bytes specifying packet length, the Payload of the packet (VESC communication message, as detailed in VESC firmware datatypes.h file), Two bytes of Cyclic Redundancy Checksum on the Payload alone, and One Stop Byte (0x03). The board microcontroller will send the throttle values in a 'set y-axis throttle' message to the VESCs and will receive motor RPM data in a 'query telemetry data' message coming back from the VESCs.

Figure 8: Vesc UART Protocol



3 Verification

In this section we cover the subsystem unit tests and system integration tests that were performed on our custom electric longboard. This was done to confirm that our modules work not only individually but also as a part of the whole board and remote system. For the complete table view of our subsystem requirements and verifications, please refer to Section 5.3.

3.1 Board Control Subsystem

The Board Control Subsystem is the most important subsystem across our entire project; it is responsible for interpreting the user input from the remote, receiving the user ejection detection and wheel RPM sensing data from the sensing subsystem, and computing the output throttle to the motor controllers based on these inputs. Should this subsystem fail, our project would certainly fail to make electric longboards safer. In Section 2.4, we depicted the throttle algorithm flowchart for the board under all possible conditions. For example, if the remote control is disconnected, the Board Control Subsystem shall neutralize the throttle. The flowchart describes the behavior we should expect from the board under said conditions.

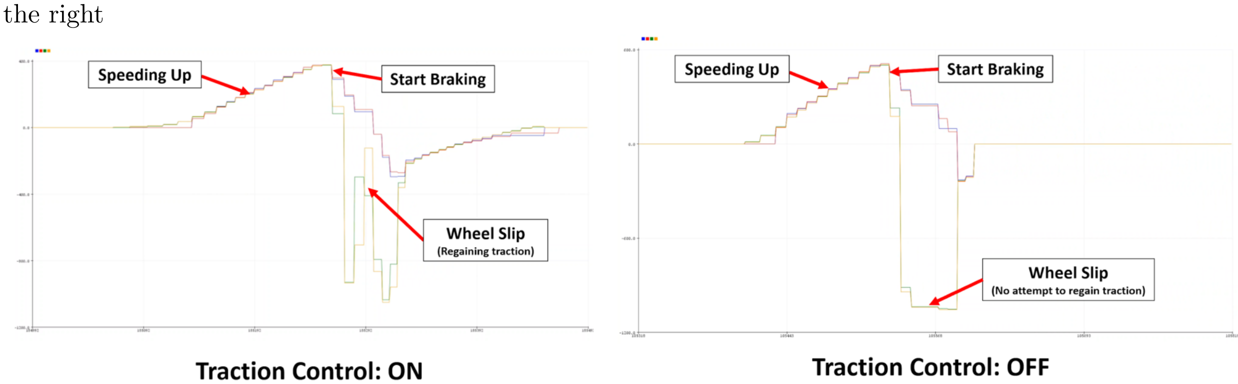
During the earlier stages of our project, the Board Sensing Subsystem and Board Drivetrain & Power Subsystem were still under development and were not ready to be inputs into our throttle control algorithm. As a result, we developed a test program that allows the tester to mock the various parameters of our system - from weight sensing, front & rear wheel RPMs, to the throttle input from the remote. This test program would output the throttle commands that our algorithm would send to the motor controllers. This program in particular was crucial in squashing bugs in our throttle and traction control algorithm. Furthermore, it is publicly available, see [9].

To verify that our custom electric longboard actually matches the behavior we expect, we evaluate whether the specific test may be run in 'bench' mode or not. The longboard is considered in bench mode if it is placed upside down with the wheels facing upwards off of the ground and weight sensing is deactivated. For the testing that does not require weight sensing, we default to running in bench mode as it is often the safest option that ensures the board cannot move from its physical position. Returning to the example above, to verify that if the remote control is disconnected, the Board Control Subsystem neutralizes the throttle, we have performed the following: put the board in bench mode, apply continuous throttle via the remote to get the motors spinning, then turn off the remote. With the remote turned off, there is no radio communication from the remote to the board. The Board Control Subsystem does not erroneously maintain the last received throttle value, but instead neutralizes throttle after a second of radio silence as we required it to do.

Some testing for the Board Control Subsystem cannot be performed in bench mode. Instead, we will place the board with the wheels facing the ground. For example, the Board Control Subsystem must prevent the longboard from accelerating faster than a walking pace (5 mph). To test this, we ensure that there is no load on the board, and then apply maximum throttle. We then physically observe that acceleration of the longboard is a fraction of its potential, and is as slow enough to be a typical walking pace.

Furthermore, our third high-level requirement states that our electric longboard will sense whether the powered wheels are slipping in the opposite direction of the unpowered wheels (as would be the case under extreme braking); when sensed, our electric longboard will automatically dampen the reverse-throttle value until traction is recovered. To test this, we ensure that we have plenty of space for the longboard to accelerate, and then simulate a rider by deactivating weight sensing. There, we accelerate the board to a non-trivial velocity, then immediately apply maximum braking throttle. This will induce rear wheel slip under normal conditions, even on concrete or pavement. When our traction control system is activated, we observe that the Board Control Subsystem senses the wheel slip attempts to mitigate it by 'pumping' the brakes (neutralizing throttle momentarily), despite the user holding down maximum throttle. We then repeat this test but with traction control deactivated, and observe that the rear wheels continually slip with no mitigation whatsoever.

Figure 9: Real-time waveforms of RPM vs time with traction control enabled on the left and disabled on the right

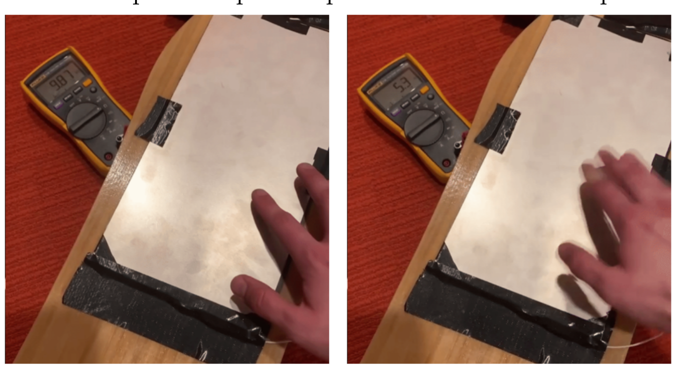


3.2 Board Sensing Subsystem

In order to verify our Board Sensing Subsystem works as expected, we must test the two parts of this system separately. The first part is the weight sensing - we want to ensure the pad is sensitive enough to detect the weight of a user, yet not too sensitive that it will trigger while riding over large bumps or obstacles. On the other hand, we need this to be just sensitive enough that when the user shifts their weight when turning or lifts up one foot to readjust their feet it still detects that a user is on the board. The first primitive test we did was to just measure the resistance of the pads under compression and under no compression to verify that the pads drop from high resistance to low resistance when force is applied. The figure below demonstrates that when the pad is uncompressed, the multimeter reads 10 kiloohms, and when someone applies pressure down on the pad it reads 5.3 ohms.

The second test involved the integration of the pressure pads with the board microcontroller and remote display. As seen in the figure below, when the user is off the board the display reads "User Ejected!". This remains true even when picking up the board and carrying it. When the user steps on the board, no matter how the weight is distributed or even if one foot is lifted up to reposition, the "User Ejected!" message disappears and the remote instead shows the throttle value. In the reverse manner, it's not until the user steps off the board that the screen goes back to "User Ejected!". Additionally a distinct chime can be heard for the event of stepping on the board and stepping off the board.

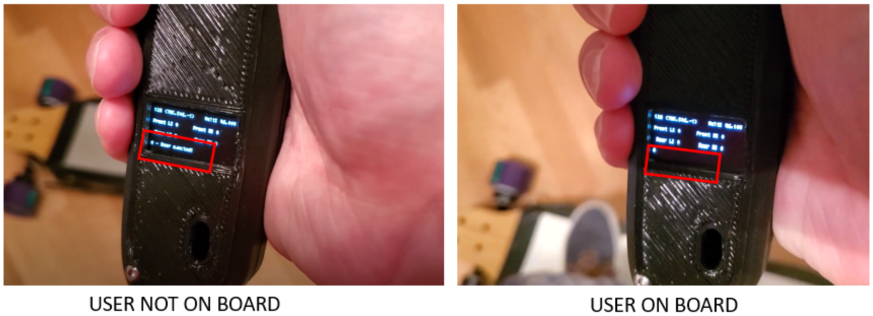
Figure 10: Resistance of pressure pad un-pressed on the left and pressed on the right



PRESSURE PAD UN-PRESSED PRESSURE PAD PRESSED

The second part of the board sensing is RPM detection, ensuring that RPM sensed in the front wheels is within 5% of the true value, in addition to detecting direction of rotation. The first test for basic functionality was putting the board on its back and manually spinning one wheel at a time, verifying that the wheel we spun was the only wheel with an RPM reading on the remote display with a magnitude greater than zero. Additionally, we saw that in one direction the RPM was positive and in the other direction. The second test we performed confirmed the accuracy and real-time tracking of the front wheel RPM sensing. We attached a

Figure 11: Detection and display alert of user off the board on the left and user on the board on the right



belt connecting the right front wheel and the right rear wheel. This forces the front and rear wheel to spin at the same speed, and we will be using the electronic speed controller's RPM readings for the rear wheels and comparing them to the front wheels. We observed the RPM waveforms plotted by our real-time monitoring system. As seen below, when the user applies throttle, the speed controller commands the rear wheels to spin at the same speed proportional to the throttle value. The front right wheel very closely follows the rear wheel RPMs since it is forced to spin at the same speed. On the other hand, the front left wheel is not connected to a motorized wheel and so is not spinning, reflected by the flat reading of RPM on the plotter below.

Figure 12: Testing tracking of RPM when coupling one side with a belt

After performing this belt test, a member of the team rode on the board in normal conditions with no wheel slip. Our real-time RPM graphs showed all 4 wheels were tracking the same speed, within a margin of 5% (to account for the turning radius and other causes of variance). When accelerating forward these values were positive, and when accelerating backwards these values were negative.

3.3 Board Drivetrain and Power Subsystem

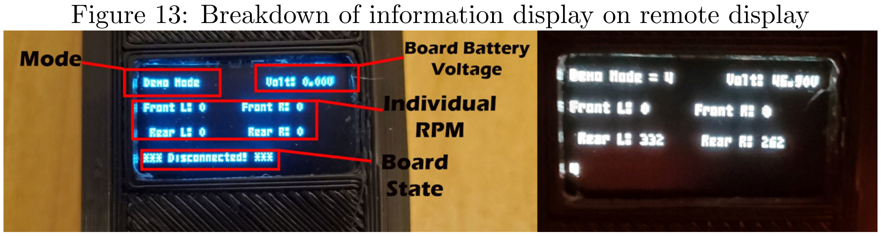
The Board Drivetrain & Power Subsystem is responsible for supplying battery power to the entire board system. It contains the motor controllers which ultimately allow our custom electric longboard to be a personal electric vehicle in a fundamental sense. For our project in particular, we required that our motor controllers keep track of their respective motor's rotational velocity, and can provide that information to our Board Control Subsystem. The pair of VESCs that we chose are capable of this function. Early in this project, to test for this, we connected a microcontroller to the UART port of our motor controllers and constantly queried for telemetry data from the VESC. On the serial plotter, we could record the current rotational speed as we physically pushed the motor at various speeds. Later, during full system integration, our board microcontroller fulfills this same role and sends the rear wheel RPMs to the remote. There, the remote's OLED display clearly shows the current rotational velocities of each rear wheel.

3.4 Remote User Interface Subsystem

The Remote User Interface Subsystem is important as it is the method by which the user inputs commands to move the longboard. To verify that this subsystem is working as expected, we rely on the OLED display located on the remote. Information on the display includes but is not limited to the following: current throttle value, throttle/deadman switch status, user ejection status, front & rear wheel RPM values.

To test whether the remote interprets user input, we first check the display when the throttle/deadman switch is pressed and then not pressed. The OLED display will show the phrase "Press button." when the button is not pressed, else show the current throttle value. Next, we test all throttle ranges, from maximum reverse to maximum forward. The OLED display will show values from -128 to 127 that correspond with all possible throttle values. These two tests confirm the fundamental functionality of the Remote User Interface Subsystem. From there, we have also tested and verified the remote's ability to display the current readings of the front & rear wheel RPMs as well as the current user ejection status atop the board.

Figure 13: Breakdown of information display on remote display



3.5 Remote Control Subsystem

The Remote Control Subsystem is responsible for sending the state of the user's throttle input to the board. Should the remote fail to receive acknowledgements from the longboard over a period of 3 seconds or more, the remote shall indicate this to the user. In the early stages of our project, we would use arduino-based microcontrollers and the NRF24L01+ modules to confirm that wireless communication of 1 to 32 byte packets was achievable. At the system integration phase, we can confirm that the remote is communicating with the longboard by observing if the remote throttle has an effect on the motors. By placing the longboard in bench mode, we are free to apply as little or as much throttle from the remote to command the motors to accelerate in the forward or reverse direction. Furthermore, we can confirm that the remote can detect if it is no longer communicating with the board by turning off the longboard system. Within 3 seconds, the remote will display the phrase “Disconnected” on the OLED display.

Figure 14: Recognition of deadman switch not pressed on the left setting motors to coast, while deadman switch pressed on the right allows throttle input

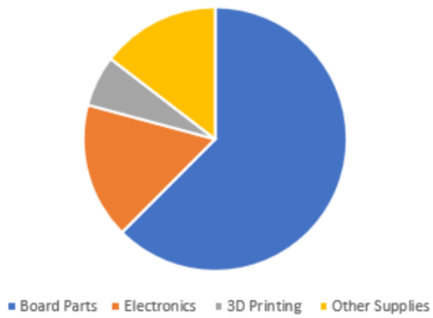
3.6 Remote Power Subsystem

The primary importance of the Remote Power Subsystem is that the remote can be powered by a single li-ion battery and to properly regulate the voltage to 3.3V. Through bench testing we found that the regulator is able to regulate with a low dropout voltage of less than 100mV, so it will work with a minimum input voltage of 3.4V. We don’t want to use a li-ion battery below 3.4V which is why we don’t cross that threshold.The regulator is also able to regulate the maximum battery voltage of 4.2V to 3.3V. With the remote PCB assembled under full load with all components attached, we are still able to get an output voltage of 3.3V powered off of the input supply of our li-ion battery. The low voltage protection circuit has also been tested with an input supply and we found the cutoff voltage to be around 3.4V, so we met our requirement in terms of shutting off the circuit when input voltage drops below 3.4V. Not explicitly part of our requirements, but our load sharing circuit has also been verified to seamlessly switch between USB 5V power and battery power which is a nice to have feature. We were able to verify that our battery charging circuit worked and automatically stopped charging at 4.184V, just below the max charge voltage of 4.2V.

4 Costs

This project was very expensive, especially considering that we only had a $100 budget. Overall, our expenses reached upwards of $2400. Note that while the parts for the final build only cost us $1384.81, this neglects all the 3D printed parts, parts for prototyping, and parts scrapped due to design changes. The cost of breakdown could be seen easier in the pie chart and table. Again, while the cost was very high, the quality of parts can be seen aesthetically and helps extend the life of our product. For an iutemized list of expenses, please see Appendix B.

Figure 15: Cost Breakdown per category



Board Parts	$1,500
Electronics	$400
3D Printing	$150
Other Supplies	$350
Total	**$2,400**

5 Conclusion

5.1 Accomplishments

This project was a complete success. In the beginning of the semester we set out the goal to increase the safety of electric longboards, and by the end that is precisely what we accomplished. On our custom electric longboard, if the user falls off, the longboard cannot accelerate out of control. If the rear wheels begin slipping from excessive user throttle input, our controllers will identify this and command the motors to actively attempt to regain traction. We achieved all of these goals and can clearly verify all of our requirements. Furthermore, we strived to continually improve our designs physically and aesthetics as well; through many different 3D-printed prototypes and configurations, we were able to create an Electric Longboard that was not only functional, but had added safety features and looked close to a real, complete product.

5.2 Future Work

While we are very proud of the product we have made, we will continue to make both physical and functional design improvements. For physical improvements, we seek to minimize enclosure sizes, add protective skid guards to protect the motors and belts, route the cables inside the deck, and mount boards with bolts rather than adhesive materials. For functional improvements, we seek to upgrade the ATmega328 board microcontroller chip to something faster allowing us the headroom for more advanced traction control algorithms, as well as to allowing us to communicate with both VESCs simultaneously (instead of communicating only with the master VESC) for increased real-time performance.

5.3 Ethics

Our project aims to improve safety features on consumer electric longboards by targeting two dangerous situations that are possible with general electric longboard designs. With this project we are increasing the effectiveness of safety standards, which correlates to IEEE's Code of Ethics Section I.1, which is to "to hold paramount the safety, health, and welfare of the public" [10]. Skateboards and longboards are known to be dangerous, as users are not secured to the vehicle. Should a user lose his or her balance, they automatically risk injury to themselves or others. Our project aims to mitigate some of the hazards present in current electric longboard designs; however, we will not completely eliminate the risks involved. To do so would be to eliminate the longboarding part of the experience entirely. With regards to the effectiveness of the mitigation we will not falsely claim that our system cannot prevent all accidents, which falls under IEEE's Code of Ethics Section I.5 by being "honest and realistic in stating claims or estimates based on available data" [10].

References

[1] T. Conneally, "The motorized skateboard renaissance," 2014.

[2] P. Hertach, A. Uhr, S. Niemann, and M. Cavegn, "Characteristics of single-vehicle crashes with e-bikes in switzerland," *Accident Analysis & Prevention*, vol. 117, pp. 232–238, 2018.

[3] "Nordic Semiconductor Infocenter."

[4] E. Jacobsen, "Firefly Remote," Feb. 2018.

[5] SolidGeek, "Electric skateboard motor mount - Caliber trucks by SolidGeek."

[6] RaouI, "Orangatang Kegel 48 Tooth HTD-3M Drive Pulley for 10 mm belt by RaouI."

[7] Benjamin Vedder, "The vesc project."

[8] "Communicating with the VESC using UART | Benjamin's robotics."

[9] Alex Krysl, "handle_throttle_control_tester."

[10] IEEE, *IEEE Code of Ethics*, 2020.

Appendix A - Requirements and Verification Tables

Table 1: Board Control Subsystem Verification and Requirements

Requirements	Verification
1. When the Board Control Subsystem detects a fatal communication failure and detects that the user is on the board, the Board Control Subsystem must command the electronic speed controllers to coast.	1. Place the board upside-down and turn off Weight Sensing with the toggle switches. 2. Press and hold the throttle/deadman switch and continuously apply throttle via the remote to spin the motors. 3. Turn off the remote. 4. Confirm that within two seconds, the motors stop spinning.
2. When the Board Control Subsystem detects that the user is no longer on the board and that the remote deadman switch is pressed, the Board Control Subsystem allows the throttle input to take effect up to a maximum speed of 5 miles per hour or less.	1. Place the board on the ground and turn on Weight Sensing with the toggle switches. 2. Press and hold the throttle/deadman switch and continuously apply maximum throttle via the remote. 3. Confirm that the longboard does not accelerate faster than 5/mph squared.
3. When the Board Control Subsystem detects that the user is on the board and that the remote dead-man switch is not pressed, the Board Control Subsystem must command the electronic speed controllers to coast.	1. Place the board upside-down and turn off Weight Sensing with the toggle switches. 2. Press and hold the throttle/deadman switch and continuously apply throttle via the remote to spin the motors. 3. Release the throttle/deadman switch but do not release the thumbwheel. 4. Confirm that within two seconds, the motors stop spinning.
4. When the Board Control Subsystem detects that the user is on the board, the remote dead-man switch is pressed, and the rotational direction of the front wheels are opposite the wheel rotational direction of the motorized rear wheels, the Board Control Subsystem will adjust the throttle value commanded to the electronic speed controllers such that the motorized wheels regain static frictional traction.	1. Ensure that there is enough space for the longboard to accelerate safely. 2. Place the board on the ground, turn off Weight Sensing, and turn off Traction Control with the toggle switches. 3. Using the remote, accelerate the board forward to a nontrivial velocity. 4. Then, apply maximum braking throttle. 5. Confirm that the rear wheels spin continuously in the direction of the throttle, regardless of the rotational velocity of the front wheels. 6. Reposition the board if needed and turn on Traction Control with the toggle switches. 7. Using the remote, accelerate the board forward to a nontrivial velocity. 8. Then, apply maximum braking throttle. 9. Confirm that the rear wheels behave differently from before and do not simply spin continuously in the direction of the throttle. The rear wheels make at least one or more attempts to regain traction before braking once more.

Table 2: Board Sensing Subsystem Verification and Requirements

Requirements	**Verification**
1. Board must detect whether the weight atop of the longboard deck is greater or lesser than 5-40 pounds.	1. Place the board on the ground and turn on Weight Sensing with the toggle switches. 2. Confirm that "User Ejected!" is present on the remote's display. 3. Mount the longboard. 4. Confirm that "User Ejected!" is no longer present on the remote's display. Confirm the board produces the corresponding mount chime. 5. Dismount the longboard. 6. Confirm that "User Ejected!" is now present on the remote's display. Confirm the board produces the corresponding dismount chime.
2. Board should be able to detect the RPM and direction of all four wheels within a tolerance of +/-5% of their true value.	1. Place the board upside-down and turn off Weight Sensing with the toggle switches. 2. Attach a belt between the left rear wheel and left front wheel. 3. Press and hold the throttle/deadman switch and continuously apply various throttle inputs via the remote to spin the motors. 4. Confirm that the left front wheel spins with the left rear wheel. 5. Confirm that the RPM value for the left front wheel and the left rear wheel are within 5% of each other, as displayed on the remote. 6. Attach the belt between the right rear wheel and right front wheel. 7. Press and hold the throttle/deadman switch and continuously apply various throttle inputs via the remote to spin the motors. 8. Confirm that the right front wheel spins with the right rear wheel. 9. Confirm that the RPM value for the right front wheel and the right rear wheel are within 5% of each other, as displayed on the remote. 10. Release throttle. 11. Confirm that the RPM values for all four wheels are 0, as displayed on the remote.

Table 3: Board Drivetrain and Power Subsystem Verification and Requirements

Requirements	Verification
1. Board must detect whether the weight atop of the longboard deck is greater or lesser than 5-40 pounds.	1. Place the board on the ground and turn on Weight Sensing with the toggle switches. 2. Confirm that "User Ejected!" is present on the remote's display. 3. Mount the longboard. 4. Confirm that "User Ejected!" is no longer present on the remote's display. Confirm the board produces the corresponding mount chime. 5. Dismount the longboard. 6. Confirm that "User Ejected!" is now present on the remote's display. Confirm the board produces the corresponding dismount chime.
2. Board should be able to detect the RPM and direction of all four wheels within a tolerance of +/-5% of their true value.	1. Place the board upside-down and turn off Weight Sensing with the toggle switches. 2. Attach a belt between the left rear wheel and left front wheel. 3. Press and hold the throttle/deadman switch and continuously apply various throttle inputs via the remote to spin the motors. 4. Confirm that the left front wheel spins with the left rear wheel. 5. Confirm that the RPM value for the left front wheel and the left rear wheel are within 5% of each other, as displayed on the remote. 6. Attach the belt between the right rear wheel and right front wheel. 7. Press and hold the throttle/deadman switch and continuously apply various throttle inputs via the remote to spin the motors. 8. Confirm that the right front wheel spins with the right rear wheel. 9. Confirm that the RPM value for the right front wheel and the right rear wheel are within 5% of each other, as displayed on the remote. 10. Release throttle. 11. Confirm that the RPM values for all four wheels are 0, as displayed on the remote.

Table 4: Remote User Interface Subsystem Verification and Requirements

Requirements	Verification
1. The remote microcontroller must be able to interpret the position of the throttle input.	1. Continually Press and hold the throttle/deadman switch. Apply no throttle on the remote. 2. Confirm that the throttle value displayed on the remote is 0 +- 1. 3. Apply maximum forward throttle on the remote. 4. Confirm that the throttle value displayed on the remote is between 125 to 127. 5. Apply maximum reverse throttle on the remote. 6. Confirm that the throttle value displayed on the remote is between -128 to -126.
2. The remote microcontroller must be able to interpret the status of the dead man switch.	1. Release the throttle/deadman switch. 2. Confirm that there is no throttle value displayed on the remote, and that "Press button." is displayed on the remote. 3. Press and hold the throttle/deadman switch. 4. Confirm that there is a throttle value displayed on the remote, and that "Press button." is not displayed on the remote.

Table 5: Remote Control Subsystem Verification and Requirements

Requirements	Verification
1. The remote control subsystem must send the throttle and dead man switch values, as read from the microcontroller, to the board control subsystem	1. Place the board upside-down and turn off Weight Sensing with the toggle switches. Press and hold the throttle/deadman switch but do not apply any throttle on the remote. 2. Confirm that motors do not spin then apply maximum forward throttle on the remote, and confirm that the motors spin in the forward direction. 3. Apply maximum reverse throttle on the remote, and confirm that the motors spin in the reverse direction. 4. Release the throttle/deadman switch while applying various throttle inputs and confirm that the motors do not spin.
2. The remote control subsystem must indicate when a fatal communication failure has been detected.	1. Place the board upside-down. 2. Confirm that "*** Disconnected ***" is not displayed on the remote. 3. Turn off the board. 4. Confirm that within 3 seconds, "*** Disconnected ***" is displayed on the remote.

Table 6: Remote Power Subsystem Verification and Requirements

Requirements	Verification
1. Must be able to regulate battery voltage to power components throughout the discharge cycle of the battery and automatically cut out power when battery voltage drops too low.	1. Connect input of voltage regulator to voltage supply. Connect output of voltage regulator to programmable load. Set voltage to maximum battery voltage, 4.2V. 2. Check voltage reading with the multimeter to make sure output voltage does not fall outside of 3.3V $\pm 5\%$ under no load and full load (50mA). Repeat this with the lowest battery voltage for proper operation, 3.45V. Now lower the input voltage and verify that the output goes to 0V somewhere between 3.45V and 3.40V.
2. The remote battery must be able to be recharged via 5V USB input within 2 hours, automatically stopping charging when the battery is at full capacity. The remote battery must be able to last at least 5 hours.	1. Start with the battery discharged to the point where it reads 3.4V. Apply 5V input via power supply to input of BMS. Monitor current delivered to battery w/ multimeter and monitor battery voltage. Verify that at the end of the charge cycle, battery voltage is 4.2V and no current is being delivered to the battery. Record time of charge cycle and verify it is less than 2 hours. 2. Apply 50 mA load and record the time it takes for the output voltage to fall to 3.4V. Ensure this is greater than 6 hours

Appendix B - Expenses Tables

Table 7: Expenses Table

Part	Manufacturer	Price	Quantity	Total
8 Mhz Crystal	Ralton Electronics	0.5	10	5
16 Mhz Crystal	Abracon LLC	0.48	10	4.8
0.01uF Ceramic Capacitor	KEMET	0.1	10	1
0.1uF Capacitor	KEMET	0.1	25	2.5
4.7uF Ceramic Capacitor	Samsung Electro-Mechanics	0.24	10	2.4
10uF Ceramic Capacitor	Samsung Electro-Mechanics	0.12	10	1.2
22pF Capacitor	Stackpole Electronics Inc	0.1	25	2.5
1uF Ceramic Capacitor	Samsung Electro-Mechanics	0.1	10	1
10kΩ Resistor	Bourns Inc	0.1	25	2.5
470kΩ Resistor	Stackpole Electronics Inc	0.1	10	1
470Ω Resistor	Stackpole Electronics Inc	0.1	25	2.5
2.26MΩ Resistor	Vishay Dale	0.1	10	1
1.18MΩ Resistor	Panasonic Electronic Components	0.1	10	1
2kΩ Resistor	Stackpole Electronics Inc	0.1	10	1
LTC4412	Linear Technologies	3.62	3	10.86
MIC5301-3.3	Microchip Technology	0.62	10	6.2
LTC1560	Linear Technologies	4.27	5	21.35
MCP73831T	Microchip Technology	0.59	10	5.9
FDC638P	ON Semiconductor	0.5	10	5
1N5819	Diodes Incorporated	0.5	10	5
Atmega328p DIP	Atmel	2.5	5	12.5
Atmega328p TQFP	Atmel	2.5	5	12.5
JST-XH Connector Kit	elechawk	20	1	20
JST-PH Connector Kit	elechawk	20	1	20

Part	Manufacturer	Price	Quantity	Total
0.1" Male/Female/ Pin Headers Kit	Honbay	10	1	10
Linear Hall Sensor	Texas Instruments	2.5	5	12.5
Latching Hall Sensor	Honeywell	1	10	10
NRF24L01+	Makerfire	10	10	100
Piezo Buzzer	uxcell	10	5	50
3.7V 480mAh Lipo Battery	YDL	7	1	7
USB C Breakout	Adafruit	3	5	15
LED Kit	OFNMY	20	1	20
64x128 OLED 5-Pack	Frienda	10	1	10
12mm Slide Switch- 10 Pack	Cylewet	6	1	6
SPDT Micro Switch- 26 Pack	Cylewet	7	1	7
Extension Spring Kit	Accessbuy	1	11	11
5mmx5mm Magnet 40 Pack	WOBATOY	10	1	10
5/16" Neodynium Magnet 10 Pack	Harbor Freight	2.8	2	5.6
Metal Plates	K&S	6	4	24
BLDC Belt Motor 6354	Flipsky	60	2	120
Kegel Pulley System	MBoards	80	2	160
Extended Trucks	MBoards	50	1	50
Custom 18650 Battery Packs	Zach Tetra Batteries	245	1	245
Jehunion V2 Bearings	Loaded Boards, Inc.	28	1	28
Vanguard Deck	Loaded Boards, Inc	100	1	100
Caguama Orangatang Wheels	Loaded Boards, Inc	100	1	100
TORQUE6 ESC	Torque Boards	135	2	135
TOTAL				$1,384.81

Appendix C - Schematics

Table 8: Remote PCB Schematic

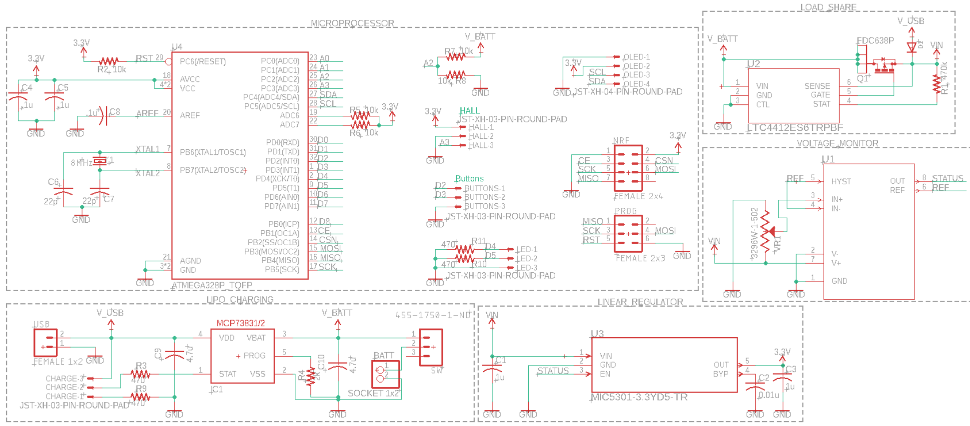
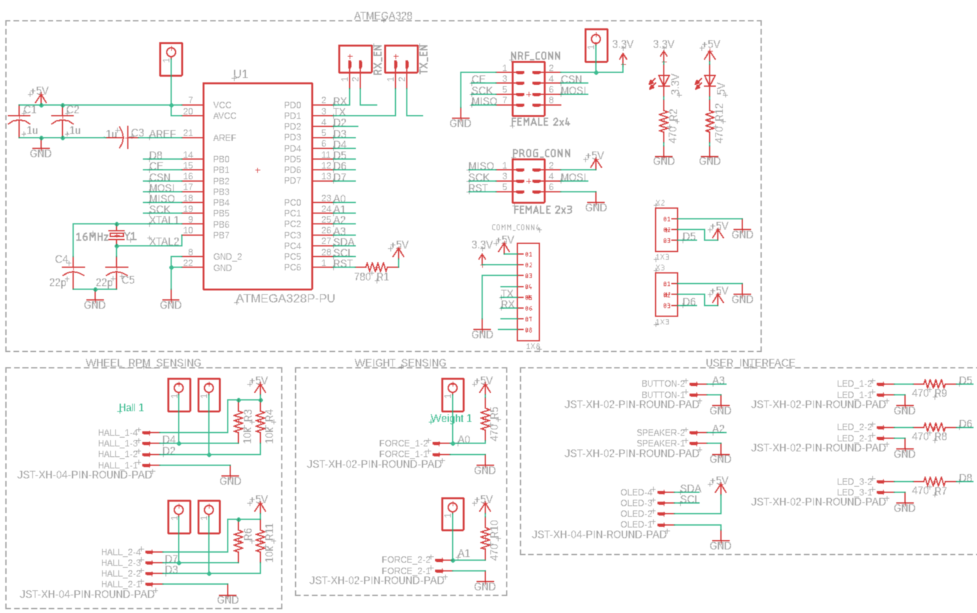


Table 9: Board PCB Schematic



Appendix D - Flow Charts

Figure 16: Traction Control Flowchart

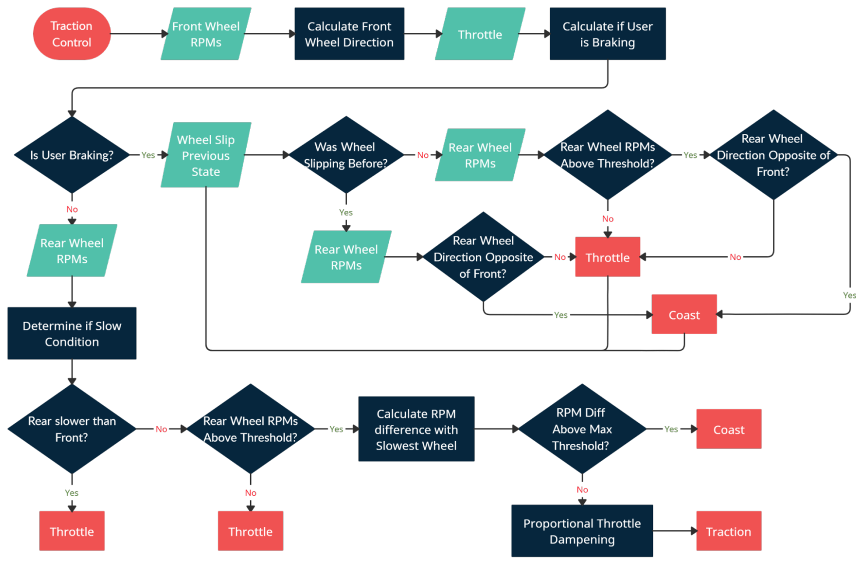


Figure 17: Throttle Control Flowchart