Motorized Longboard

Design Review

Daniel Moon – Kevin Lee – Leon Ko
TA: Mustafa Mukadam

ECE 445: Senior Design

February 25, 2013

Table of Contents

I. Introduction

1. Project Vision

Longboarding has been around for over 60 years, but has gained traction amongst younger communities in the last decade. In the last couple of years, there has been an assortment of upstart projects attempting to create a commercially viable motorized longboard. Basic propulsion and even regenerative braking have been achieved, and we look to build on those successes. We look to add assisted turning, the ability to reverse, and a kill switch as new functions.

Some of our group members currently use longboards as their mode of transportation around campus. Like bicycles, it is a cheap, efficient, and environmentally friendly way to travel. We were excited to consider different ways to make longboarding even better. All the functions that we are looking to add are functions that most longboarders have at some point desired. Our ultimate goal is to provide a product that is not only fun to use, but also commercially viable.

2. Objectives

2.1 Goals

- Develop a kill switch that will deploy in case of emergency
- Enable ability to reverse
- Add regenerative braking to conserve energy
- Include assisted turning for safer turns

2.2 Functions

- Flex sensor to determine whether to enact kill switch and regenerative braking
- Reversible motor to enable reverse propulsion
- Pulse width modulator to enable assisted turning

2.3 Benefits

- Create an environmentally friendly mode of transportation
- Provide a longboard with beginner-friendly functions
- Offer quick transportation in certain locales (few hills)
- Grant speed without physical exertion

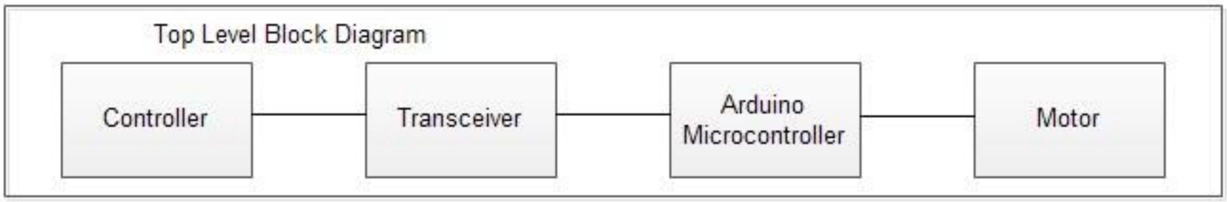
2.4 Features

- Longboard able to attain speeds up to 13-20 mph
- Longboard able to reverse at up to 4-8 mph
- Pulse width modulator that grants assisted turning

II. Design

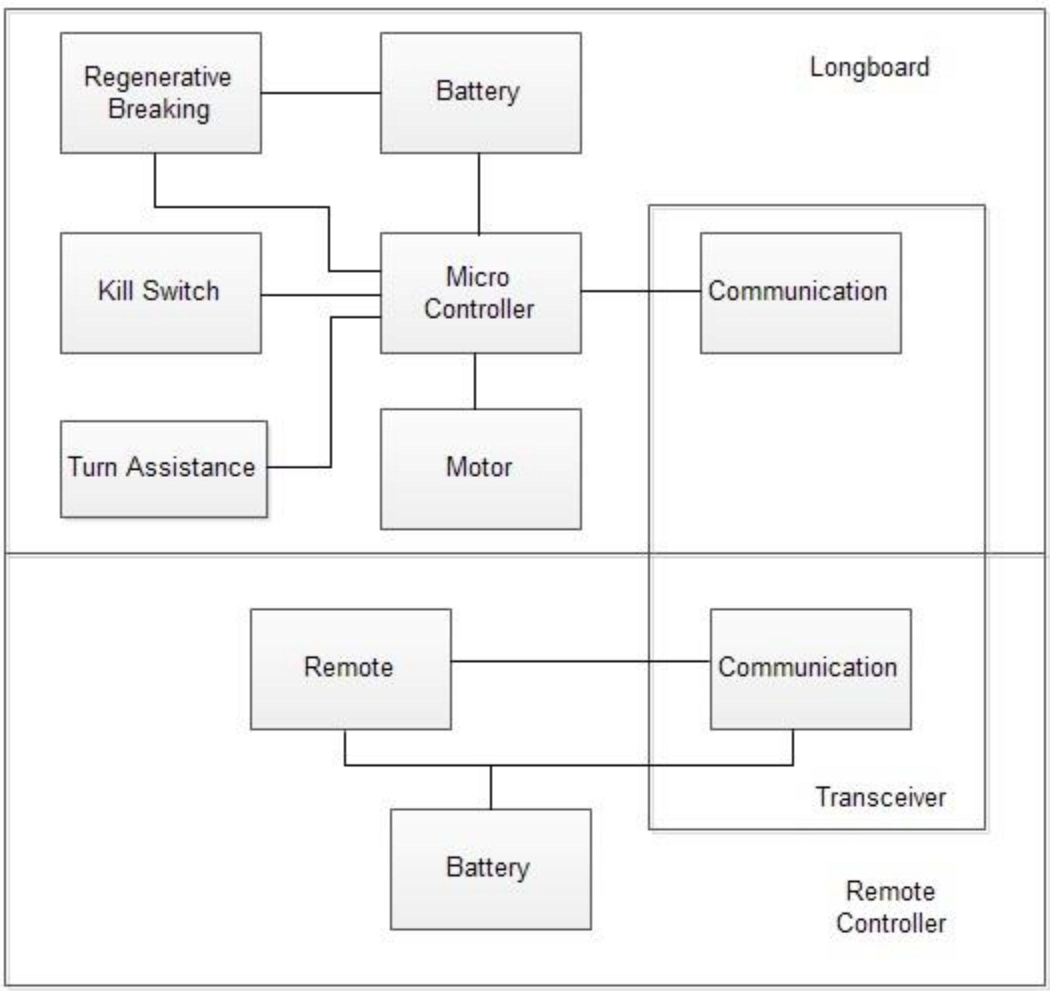
1. Top Level Block Diagram

Figure 1: Top Level Block Diagram



2. Block Diagram (Specific)

Figure 2: Specific Component Block Diagram



3. Block Descriptions

3.1 Remote

The remote is the only user input for the motorized longboard. Three potential inputs exist on the remote: access to acceleration, deceleration, and regenerative breaking. Acceleration and deceleration will be achieved through manipulation of a joystick, with the magnitude of acceleration determined by how far the joystick is pushed. Regenerative breaking will be activated by pressing and holding a button. All input received by the remote will be transmitted wirelessly to the transceiver block.

3.2 Transceiver

The transceiver is the waypoint between the remote and Arduino micro-controller. While it does not perform any data manipulation or analysis, it is a vitally distinct section.

3.3 Arduino Micro-Controller

The micro-controller is where all the data analysis is performed. For acceleration and deceleration, it decodes the signal and passes the information via pulse width modulator to the motor. When the regenerative breaking button is held down, the micro-controller will decode the signal and activate the regenerative breaking. The last function the micro-controller manages is the kill switch. When the implanted flex sensor senses that the longboard is no longer bent (indicative of a person's weight on the board), the micro-controller will activate the regenerative breaking.

3.4 Motor

The motor receives the acceleration and deceleration signals from the Arduino micro-controller. Based on the pulse width modulator signal, the motor outputs the appropriate revolutions per minute. Furthermore, when regenerative breaking is enacted, the motor turns into a generator that will recharge the battery.

3.5 Remote

The physical remote will consist of a joystick to control forward and reverse acceleration. There will also be a button that can be held to enact regenerative breaking. The remote is battery powered (separate from longboard battery) and interacts with the wireless transceiver.

3.6 Communications (Transceiver)

The transceiver looks to provide instantaneous communication between remote and micro-controller.

3.7 Micro-Controller

The micro-controller performs all signal processing from the remote and data analysis from the flex sensor. The micro-controller is directly connected to the kill switch, which is a flex sensor. The micro-controller is also directly connected to turn assistance, which consists of a pair of infrared sensors. The micro-controller passes the information of whether the regenerative breaking button is pushed to the regenerative breaking block. Code is then run which sends information on how to transform the motor into a generator. Additionally, when the infrared sensors detect that the board is tilted, the micro-controller will use the pulse width modulator to decrease the rate of the motor. The micro-controller is powered by a lithium-ion battery.

3.8 Turn Assistance

Turn assistance consists of a pair of infrared sensors hard wired into the micro-controller. One sits on the left side and the other sits on the right side. Longboards tend to ride extremely close to the ground, and as a result it is possible to use an IR sensor to detect turns. When the user enters a sustained turn, one side of the longboard is closer to the ground, enough to increase the voltage output of an IR sensor. When the IR sensor tells the micro-controller that the longboard is turning, the micro-controller will decrease the PWM accordingly and send the new PWM to the motor. There will be a time delay and accepted voltage ranges instituted to allow for extenuating circumstances (such as turbulence, brief turns, etc).

3.9 Kill Switch

The kill switch consists of a flex sensor that is hard wired into the micro-controller. Because of the flexibility of longboards, the board tends to sink slightly at the center when there is a rider. The flex sensor takes advantage of this information, detecting when the board is no longer flexed (suggesting no rider). When there is no rider, the kill switch sends a signal to the micro-controller, enacting the regenerative breaking. We will institute a reasonable time delay to allow for extenuating circumstances (i.e. the rider jumps).

3.10 Regenerative Breaking

This block is a length of code that detects whether the button has been pushed or if the kill switch has been enabled. If so, it will send a signal to the micro-controller changing the motor into a generator.

3.11 Battery

The motorized longboard will utilize a lithium battery, with a projected life of 4-8 miles.

3.12 Motor

The motor inputs a signal from the micro-controller. For acceleration/deceleration, the motor will simply run at the required revolutions per minute. When regenerative breaking is activated, the motor will instead act as a generator.

4. Schematics

Figure 3: Detailed Schematic of Arduino Uno R3

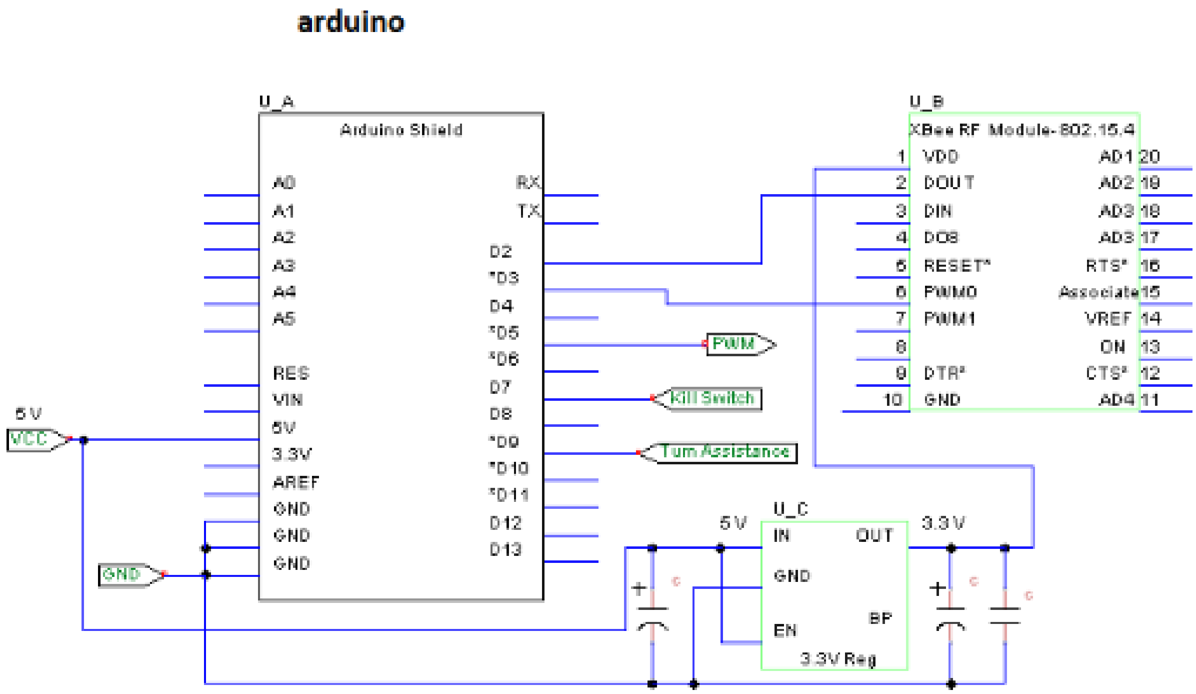
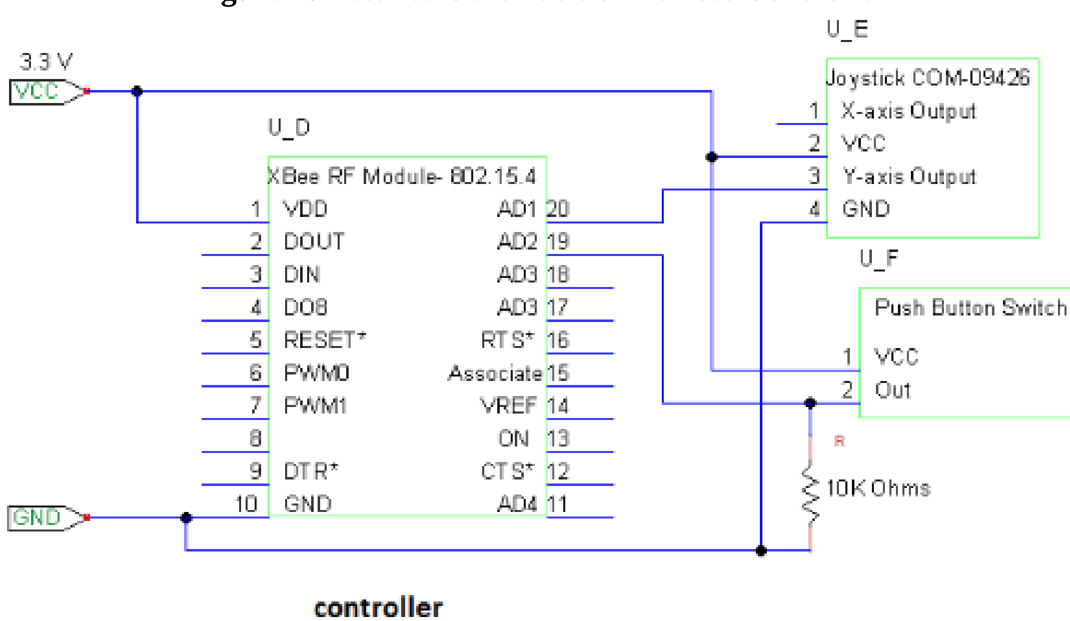


Figure 4: Detailed Schematic of Remote Controller



III. Requirements and Verification

1. Requirements and Verification

Requirements	Verifications
Longboard Battery 1. Output voltage of the battery is 25.6V +/- 10% 2. Operating range (without any regenerative braking) of 4-8 miles 3. Can be successfully recharged with regenerative breaking	**Longboard Battery** 1. Probe the battery terminals and confirm that the voltage falls within desired range. 2. Run board (without flex sensor) at specific speed until battery expires. 3. Test compatibility with motor/generator
Kill Switch 1. Ensure flex sensor can bend as much as the board 2. Check that change in flex sensor creates a large enough resistance change	**Kill Switch** 1. Derive number of degrees that board bends, then apply to sensor 2. Compare resistance from flat to resistance with rider for meaningful difference
Regenerative Braking 1. ~4-6% of energy is recovered when braking	**Regenerative Braking** 1. Test amount of energy to achieve certain velocity. Proceed to activate regenerative breaking, and confirm that recovered energy falls within desired range.
Micro Controller 1. Ensure that ~4.8-5.2V is supplied to the Arduino via voltage divider. 2. Bug-free software 3. The Arduino communicates with the XBee through the serial protocol with a baud rate of 9600	**Micro Controller** 1. Use a voltmeter and confirm that voltage falls within desired range. 2. Test each function a. Display adjustments and outputs to ensure proper function b. Simulate scenarios with hardcoded inputs c. Confirm no bugs 3. Ensure that XBee is properly mounted on shield, which is in turn mounted on the Arduino. Ensuring the baud rate is done in the setup section of code.
Motor 1. Ensure smooth transition to generator 2. Must provide enough thrust to propel longboard (with user) at 13-20mph 3. Must provide enough thrust to reverse longboard (with user) at 4-8mph 4. Make sure the current capacity does not	**Motor** 1. Run dummy code that pushes motor to become generator and vice versa 2. Test amount of propulsion generated when running forward with max PWM 3. Test amount of propulsion generated when running in reverse with max PWM

exceed boundaries (which would overheat the motors)	4. Use resistor and ammeter to test current
Turn Assistance 1. Making a turn results in change in IR output 2. Change in IR output results in change in PWM	**Turn Assistance** 1. Test IR sensor independently with Arduino code 2. Ensure that code results in reduced PWM sent to motor if IR is not within normal range.
Communication 1. Communication range of 100 ft +/= 10% (2.4 GHz) 2. Ensure that 3.3V+/- 10% is supplied to XBee 3. Check that correct data is transmitted	**Communication** 1. Test that the data transferred by one XBee is received by the other XBee 2. Use voltage divider and test leads going into XBee 3. Connec t XBee to Arduino and run code that outputs data
Remote Battery 1. Check that batteries supply a total of 2.8~3.4 V for the XBee 2. Check that batteries supply enough voltage that the potentiometer will still give meaningful differences in output 3. Battery lifetime of 20 Hours	**Remote Battery** 1. Use a voltmeter to test total voltage going into XBee 2. Arrange final configuration of batteries, and connect potentiometer. Test sensitivity and change in voltage with a supply of ~2.8-3.4V 3. Test the lifetime of 2 AA batteries in series
Remote Controls 1. Ensure that potentiometer outputs a voltage of range ~0-3.0V 2. Use pull-up resistors to regulate input 3. Ensure button is registering signal	**Remote Controls** 1. Test with dummy Arduino program 2. Check that inputs does not change with turbulence 3. Connect button directly to Arduino/LED and test signal

2. Tolerance Analysis

To perform the requirements, the pulse width modulator must be transmitted properly from the wireless controller to the micro-controller. The magnitude that the joystick is tilted is proportional to the speed at which the motor will run. Tilting the joystick forward will run the motor forward, and vice-versa. The PWM will also be affected by the angle at which the longboard itself is tilted. In order for the assisted turning to be effective, PWM will be increasingly lowered to a soft cap of about 25% speed reduction the more the board is tilted from the neutral position. The PWM will be observed through the given Arduino program that measures output to make sure that the appropriate sampling rate is seen in accordance to the tilt of the joystick and board.

IV. Diagrams

1. Flow Charts

Figure 5: Flow Chart for Potentiometer

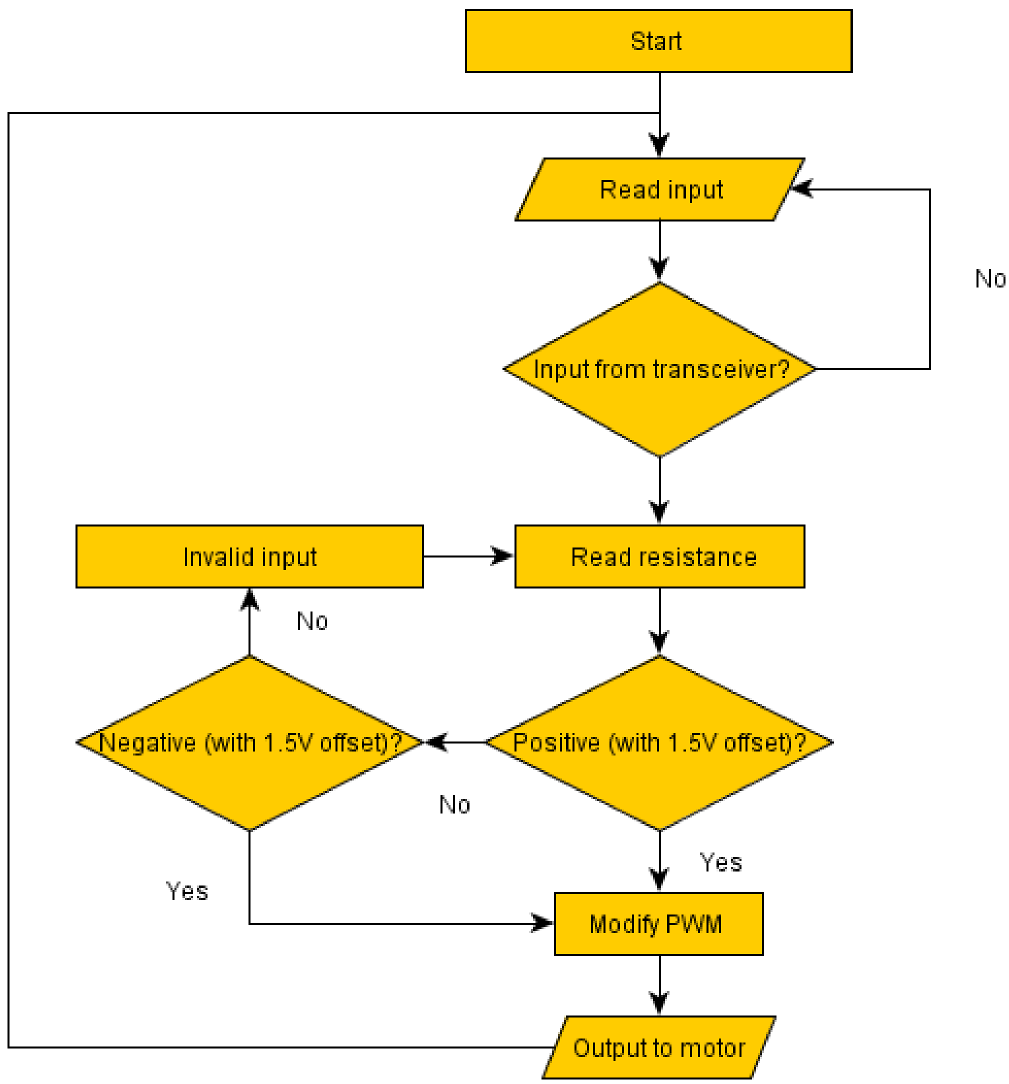
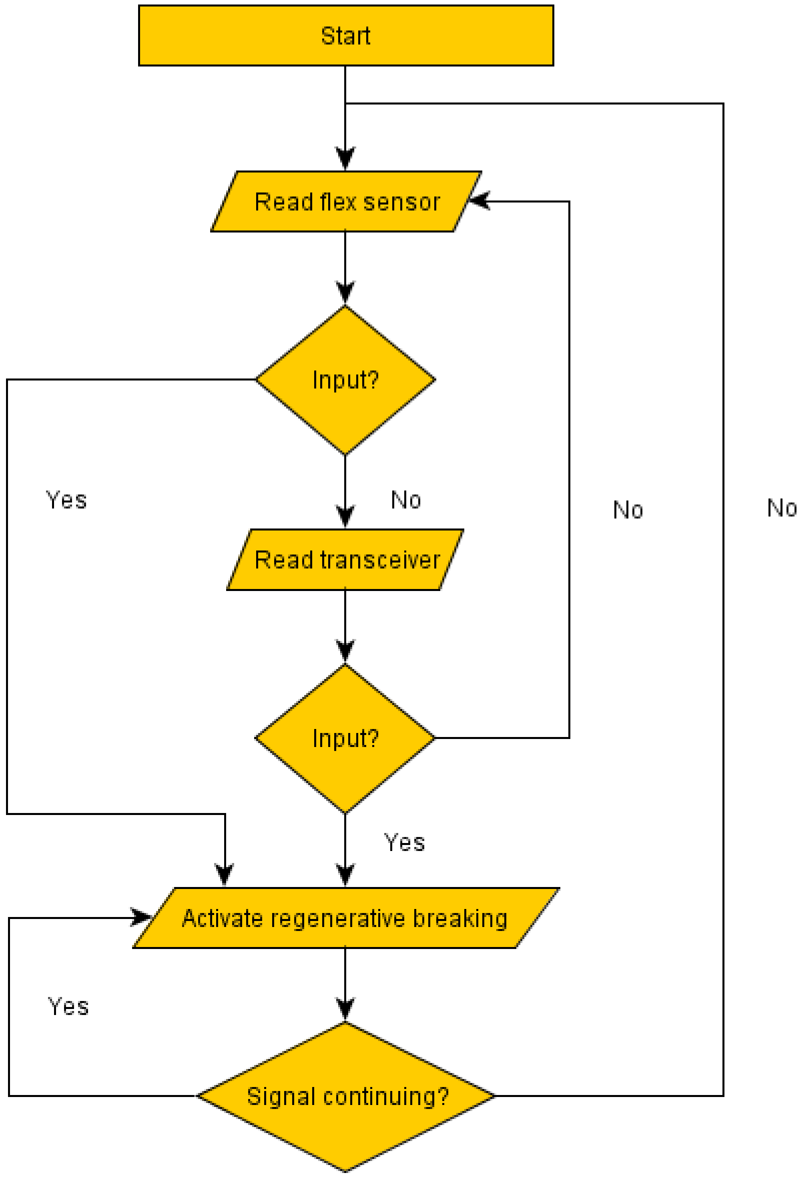
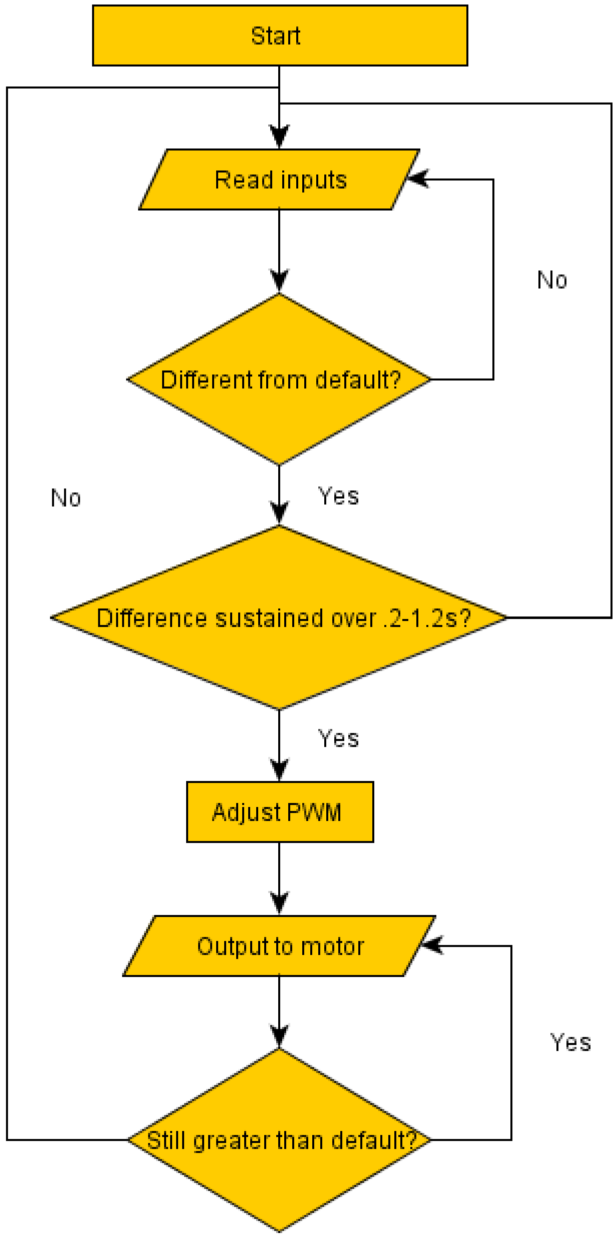


Figure 6: Flow Chart for Regenerative Braking



Note: After the first "Input?" conditional box, "Yes" means the rider is off of the longboard, and that regenerative braking should be activated. "No" means the rider is on the longboard, and regenerative braking should remain off unless the brake button is pressed.

Figure 7: Flow Chart for Assisted Turning with IR Sensor



2. Pin Charts and Part Specifications

Figure 8: Pin Schematic for Xbee through Xbee Shield [3]

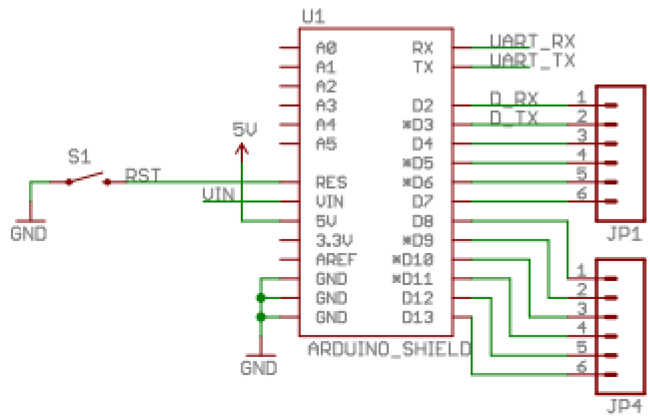


Figure 9: Physical Pin Diagram for Xbee Shield [3]

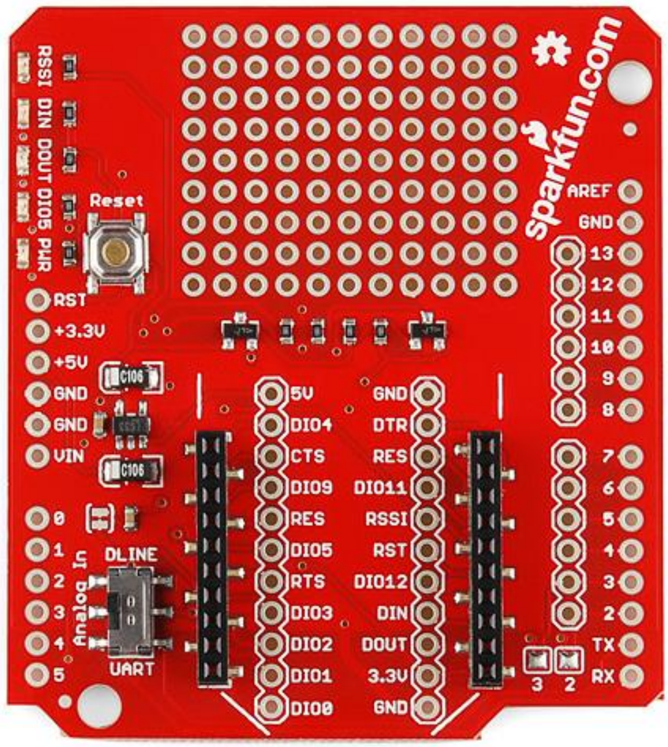


Figure 10: Pinout for Arduino Uno R3 Board [4]

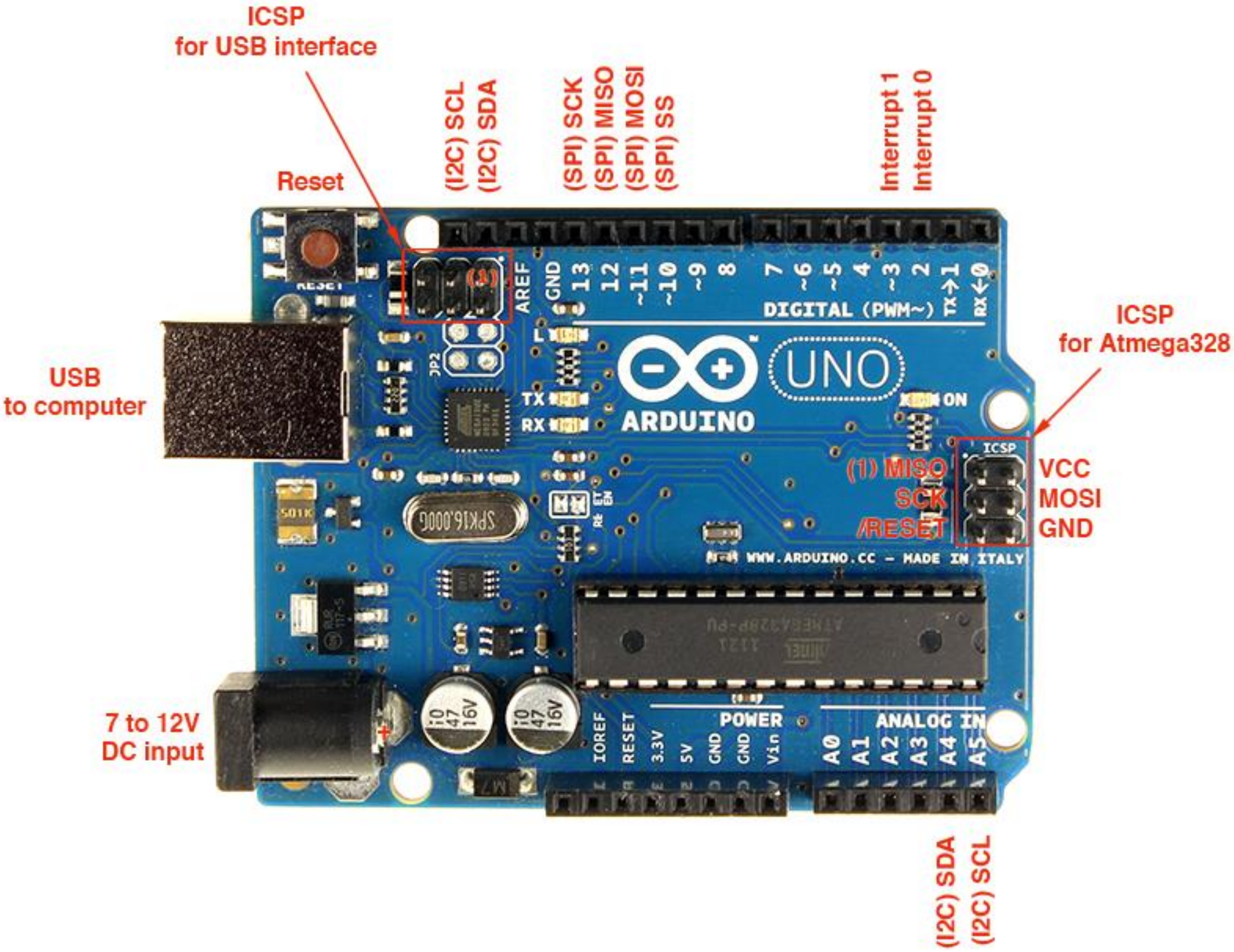
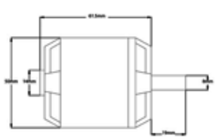


Figure 11: Specifications for RimFire 1.60 [5]

RimFire 1.60

- *Includes prop adapter and motor mount.*



GPMG4795

Description	Stock No.	Diameter	Length	kV	Constant Watts	Burst Watts	Weight	Shaft Diameter (mm)	Voltage Range	Sport	3D	Power System Recommendation		
												ESC	LiPo	Prop
RimFire 1.60	GPMG4795	63 mm (2.5 in)	62 mm (2.4 in)	250	2500	3200	634 g (22.4 oz)	8 mm (0.31 in)	33.3 - 44.4V / 9-12S LiPo	9070 g (20 lbs)	5445 g (12 lbs)	80 Amp	12S	16x8 to 18x8 Electric

V. Calculations and Simulations

1. Pseudo-code

```
//Pin numbers
int flexSensor = 1; //flex sensor connected to analog pin 1
int regenButton = 2; //digital input from transceiver (button) connected to digital pin 2
int potentiometer = 3; //PWM from transceiver connected to digital pin 3
int IRleft = 4; //IR sensor for turn assistance connected to digital pin 4
int IRright = 5; //IR sensor for turn assistance connected to digital pin 5
int motor = 9; //output pin to motor, digital pin 9

//Values
int flexValue = 0; //variable to store resistance from flex sensor
double leftValue = 0; //variable to store voltage from left IR sensor;
double rightValue = 0; //variable to store voltage from right IR sensor;
bool buttonPush = FALSE; //variable to store whether button is pushed

//functions

void setup()
{
    Serial.begin(9600); //opens serial port, set data rate to 9600bps
    pinMode(flexSensor, INPUT);
    pinMode(regenButton, INPUT);
    pinMode(potentiometer, INPUT);
    pinMode(motor, OUTPUT);
}//end setup

void loop()
{
    flexValue = analogRead(flexSensor);
    buttonPush = digitalRead(regenButton);
    //read potentiometer/IR sensor
    if(flexValue > threshold || buttonPush == TRUE)
            //call regenerativeBraking function
    if(//joystick y value is non-0)
            //call PWMmodifier function
    if(IRleft > threshold || IRright > threshold)
            //call assistedTurning function
}//end loop
```

2. Calculations

In order to match and order compatible parts, many values must be calculated. Due to our requirement of minimum top speed being 13 mph, we will use this minimum top speed to calculate the desired RPM of the wheel. The Abec 11 Big Zig wheels chosen have a diameter of 75 mm. So:

$$RPM_{wheel} = \frac{S}{C(60)} = \frac{20921.5\ m/hr}{(.075\pi)(60)} = 1479.89\ RPM \tag{1}$$

where S is the desired minimum top in meters per hour and C is the circumference of the wheel in meters. From the above calculation, we see that about 1500 RPM is desired from our wheel. Next, we must calculate the required RPM of the motor to go along with our wheels. Assuming we use a gear ratio of 4:1 from motor to wheel (the gear ratio should not be too high since this could cause clutter under the longboard), the motor RPM would then be:

$$RPM_{motor} = G.R. x\ RPM_{wheel} = 4\ x\ 1500 = 6000\ RPM \tag{2}$$

where G.R. is equal to gear ratio from motor to wheel. So now, a motor with a RPM of 6000 is desired. At this point, the calculation for motor and battery becomes a problem of availability of products. Motors have a kV rating which stand for RPM/volts, and a motor with a lower kV rating are generally more expensive, but give more torque. A rather inexpensive motor was found with a rather low kV rating, and so the RimFire 1.60 Outrunner was chosen for this project. The kV rating for this motor is 250 kV. Now we must find a compatible battery for this motor. Knowing that the desired RPM is 6000 and the kV rating is 250 kV:

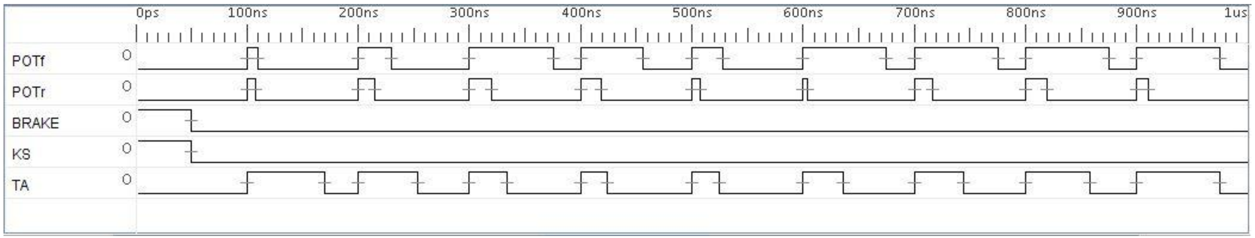
$$V_{battery\ desired} = \frac{RPM_{motor}}{kV\ rating} = \frac{6000\ RPM}{250\ RPM/volt} = 24\ volts \tag{3}$$

Now a voltage of at least 24 volts must be chosen for the battery. LiFePO4 type batteries were chosen for this project due to their high discharging current, non explosiveness, and long life cycles. LiFePO4 batteries are created by putting single cells in series, and each cell is usually rated at 3.7 V. So the desired cells of the battery S is:

$$S_{battery} = \frac{24\ v}{3.7\ v} = 6.48\ cells \tag{4}$$

A 7 cell battery must be used for this project, which is rated at 25.9 V. The LiFePO4 battery that we have selected is rated at 25.6 V, which will fit the minimum speed requirements, even with losses due to load and wire resistance. A higher voltage battery will be considered if funding exists, and if so, a 38.4 V battery will be used in order to achieve max speeds of 20 mph.

3. Simulation



POTf (potentiometer forward): The first clock cycle is with the potentiometer at the default setting. Starting at 100ns, the potentiometer is slowly incremented, increasing the PWM. At 300ns the largest PWM is achieved with ~80% duty cycle. Afterwards, the potentiometer is eased until it drops off. From 600ns to the end, the potentiometer is held at near constant PWM.

POTr (potentiometer reverse): Like in POTf, the first clock cycle is the potentiometer at default. At 100ns, the joystick is slowly pushed back, increasing the PWM. However, because we wish to limit the speed at which the longboard can reverse, the resulting PWM increase is much smaller than in POTf. In POTf, the PWM peaked at almost 80% duty cycle. In POTr, the max duty cycle is at most ~20%.

BRAKE (brake button): This timing diagram is reflective of the effect of the brake on the controller. The first clock cycle is reflective of a normal PWM. At 100ns, the brake button is pushed, and the PWM is dropped to 0% duty cycle. This continues throughout the rest of the timing diagram. Because this is a simulation, it did not include the time delay that would be implemented in the actual design.

KS (kill switch): This timing diagram is reflective of the effect of the flex sensor on the controller. Although it is a completely different sensor, it enacts the same function in Arduino as the brake button. As a result, the timing diagrams mirror each other.

TA (turn assistance): The final timing diagram demonstrates the effect of the turn assistance on PWM. The first clock cycle is a control, with no input. At 100ns, the PWM resembles that of a normal PWM. At 200ns, the turn assistance is engaged. As a result, the PWM slowly decreases, resulting in a slightly slower longboard. This is a gradual process as the turn assistance does not want to throw the user off. Starting at 600ns, the turn ends, and the PWM slowly increases back to its original value.

VI. Cost and Schedule

1. Cost Analysis

1.1 Labor

Name	Hourly Rate	Overhead (2.5)	Hrs/wk	# wks	Total Hrs	Total
Daniel Moon	$40	$100	10	11	110	$11000
Kevin Lee	$40	$100	10	11	110	$11000
Leon Ko	$40	$100	10	11	110	$11000
Labor Total						$33000

1.2 Parts

Item Name	Item No.	Unit Cost	Item Quantity	Item Cost	Ordered ?
Loaded Vanguard (Complete Longboard: Flex 2, Paris or Randall 180 mm, Abec 11 80A Big Zig, Bones Super Reds)	N/A	$320.00	1	$320.00	Yes
Great Planes Rimfire 1.60 Outrunner Brushless Motor	GPMG4795	$179.97	1	$179.97	No
LiFePo4 Battery (25.6V, 9.9Ah, 253 Wh)	LFP-25.6V10.2Ah	$302.32	1	$302.32	No
Arduino Uno Revision 3	DEV-11021	$35.00	1	$35.00	No
Hobbywing Platinum 60A Pro ESC	Hobbywing-Acc-HW-60A-ESC	$65.36	1	$65.36	No
Smart Charger (6.0A) for 25.6V LiFePo4 Battery Pack	CH-L2596N	$84.69	1	$84.69	No
Flex Sensor 4.5”	SEN-08606	$15.00	1	$15.00	No
XBee Chip Antenna + Shield	WRL-08664	$38.95	2	$77.90	No
Thumb Slide Joystick	CO-09426	$3.50	1	$3.50	No
Mini Push Button Switch	COM-09190	$0.50	1	$0.50	No
2.2 x 6.5 in. Breadboard	103-1100	$12.88	2	$25.76	Yes
AA Battery	EN91	$0.34	12	$4.08	Yes
Infrared Proximity Sensor Short Range	Sharp GP2D120XJ00F	$13.95	2	$27.90	No
PCB	N/A	$40.00	1	$40.00	Design
Total Cost				**$1179.98**	

1.3 Grand Total

Grand Total = $1179.83 + $33000 = $34179.83

2. Schedule

Week	Person	Task
2/4	Daniel	Complete proposal
	Kevin	Begin design process
	Leon	Learn and familiarize with Arduino
2/11	Daniel	Research power supply and review design
	Kevin	Finalize design
	Leon	Learn Eagle CAD, design flow charts
2/18	Daniel	Specs for motor, flex sensor, battery
	Kevin	Design and build wireless controller
	Leon	Code Arduino
2/25	Daniel	Mount components onto the board
	Kevin	Output of controller is detected by the longboard transceiver
	Leon	Assemble PCB for micro-controller
3/4	Daniel	Begin preliminary testing of board
	Kevin	Assist with preliminary testing of board
	Leon	Verify coding components of project are working as desired
3/11	Daniel	Continue preliminary testing of board
	Kevin	Continue assisting with preliminary testing of board
	Leon	Using feedback from riding experience optimize code to smooth motor and PWM controls
3/18	Daniel	Verify board components
	Kevin	Verify controller and transceiver components
	Leon	Verify Arduino components
3/25	Daniel	Debug board components
	Kevin	Debug controller and transceiver components
	Leon	Debug Arduino components
4/1	Daniel	Final testing of all components of board
	Kevin	Final hardwiring and calibration of components
	Leon	Final optimization of code
4/8	Daniel	Ensure all specifications met with board
	Kevin	Prepare presentation
	Leon	Prepare final paper
4/15	Daniel	Prepare demo
	Kevin	Finish presentation
	Leon	Finish final paper
4/22	Daniel	Demo
	Kevin	Check in supplies
	Leon	Complete final paper

VII. Safety

As this project deals with motorizing a longboard to high speeds, many safety precautions should be taken while conducting tests or general riding of the longboard. Protective gear should be worn by the rider at all times which, at the minimum, requires a skateboard/longboard helmet that at least covers the entire top of the head. The helmet should extend down the back of the head as well, to about two inches above the start of the neck, to ensure damage to the skull from falling or crashing does not happen. A full covering helmet can also be used, if desired. Protective gear can also include elbow and knee pads, ankle braces, wrist guards, and slide gloves. These will ensure that no damage to any joints will occur, but are not required and should be used according to the rider’s discretion.

Only an experience rider should be attempting to use this board at higher speeds. If an inexperienced user is attempting to learn on this board, a switch or some variation of PWM limiter should be used, so top speeds cannot be obtained. High speeds can cause falls, crashes, and the board being sent towards dangerous areas if the rider decides to get off the board for his or her own safety. The kill switch was designed to help with these issues, but rider precautions should be taken.

Apart from the riding portion of this project, some other safety issues should be considered, and precautions should be taken. Since a rather large lithium-iron-phosphate battery will be used, precautions with charging and wiring to the circuitry should be taken. Also, ensuring that a large voltage does not enter the transceivers or the microcontroller is another precaution that should be taken.

VIII. Ethical Considerations

With accordance to the IEEE Code of Ethics, throughout the entirety of this project, we vow the following:

1. "To accept responsibility in making decisions consistent with the safety, health, and welfare of the public, and to disclose promptly factors that might endanger the public or the environment"

   In the interest of public safety, all testing of this longboard will be done away from busy streets and high population density areas. The board will also be designed with the safety and welfare of the rider in mind, as well as pedestrians and other vehicle riders.

4. "To be honest and realistic in stating claims or estimates based on available data"

   All claims of battery life, top speeds, and other specifications of this longboard will be calculated, with the goal of obtaining the most accurate claims and simulation results possible.

7. "To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others"

   As our group goes through the different reviews and criticisms, we will accept all comments with open minds, be upfront about our errors, and credit those who have contributed to our project outside of the group.

IX. References

[1] "IEEE Code of Ethics." IEEE - Advancing Technology for Humanity. N.p., n.d. Web. 25 Feb. 2013.

[2] "Xbee / Xbee-PRO RF Modules." *Sparkfun*. N.p., 23 Sept. 2009. Web. 25 Feb. 2013.

[3] "Xbee Shield." *XBee Shield*. Sparkfun, n.d. Web. 26 Feb. 2013.

[4] Gammon, Nick. *Gammon*. Ed. Nick Gammon. N.p., 22 Dec. 2011. Web. 25 Feb. 2013. <http://www.gammon.com.au/forum/?id=11473>.

[5] "The Brushless Advantage for Outrunner Design Motors!" *ElectriFly*. N.p., n.d. Web. 26 Feb. 2013.

[6] “The Ultimate Guide to Lipo Batteries - Electric Motors, ESCs, Batteries, Etc. @ URC Forums." *Ultimate RC*. Ultimate RC, 14 Nov. 2010. Web. 26 Feb. 2013.