The Phoenix Board – The Solar-powered Smart Longboard

Hanser Gonzalez, Alex Lapegna, and Simeon Richards

Dept. of Electrical and Computer Engineering
University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — In today's age, a growing number of people are doing whatever they can in order to reduce their carbon footprint and mitigate the effects of global warming. A part of this has been to make our methods of transportation more environmentally friendly. Outlined in this paper is the design process and methodology for the Phoenix Board, which is an electric longboard with numerous safety features and battery recharging through the combination of solar and wireless technologies. It is accompanied by a mobile app which is used to control the board while providing various statistics and metrics.**

***Index Terms* — Bluetooth, brushless motors, embedded software, global positioning system, inductive charging, mobile applications, solar energy.**

I. Introduction

The Phoenix Board is an electric longboard that is tailored towards short to medium range travelling, such as getting around a college campus and navigating within a metropolitan area with high traffic congestion. The motivation for this project came about due to an initial agreement between the engineers to integrate each individual's idea in some capacity. This helped in pushing us to learn about and begin to work with topics that were of interest to us while also serving as a way to keep us focused and motivated throughout the senior design process. The three main things we wanted our project to incorporate were solar energy, computer vision and wireless communication, either Wi-Fi or Bluetooth.

In developing the Phoenix Board, our team spent a significant amount of time making sure that our goals and expectations for this project were reasonable but also innovative enough that developing and fully realizing those ideas would be a worthwhile reward. The fundamental backbone for many of our functions and features focused on creating a product that would provide the user with an enjoyable experience, have a minimal environmental footprint, had the ability to self-sustain for an extended period of time, and provided high factors of reliability and efficiency. The core features we decided on for our project are as follows:

1. The motor functions properly and allows the user to speed up and slow down.
2. The battery receives charge.
3. Our PCB communicates with all necessary peripheral components.
4. Mobile communications are consistent and precise.
5. Weight-sensitive safety mechanism mitigates chances of injury should the rider fall off.

Along with that we had a number of stretch features we wished to include in order to give the user some customizability and further enhance their safety. For the sake of brevity, these will not be listed as these features changed to varying degrees or were completely eliminated throughout the design process.

II. System Design Overview

The design of the Phoenix Board can be broken down into five major subsystems, which will be working in unison all or most of the time. These subsystems can be seen below in Fig. 1, our Block Diagram, which also demonstrates some of the ways in which we expect them to interact with one another. Our PCB subsystem is composed of only the PCB but is central to our design because it will be in control of much of the functionality of our board. The power subsystem is made up of a wireless charging circuit, a solar charging station, a power source and a battery/charge sensor. The communications subsystem involves a Bluetooth module along with a mobile device that will be running our companion application.

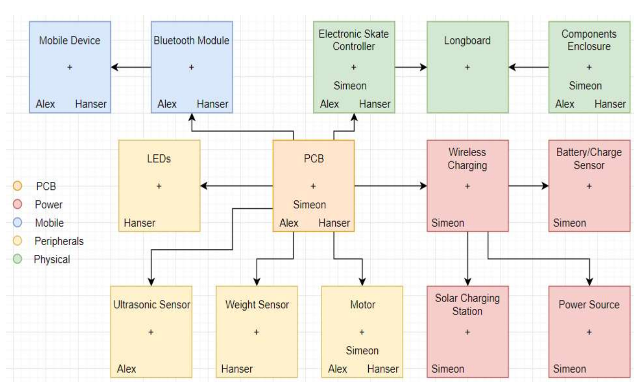


Fig. 1. Block diagram of the Phoenix Board's design and components.

The remaining two subsystems are the peripherals and the physical hardware. The former of these subsystems is composed of an ultrasonic sensor, LEDs, a weight sensor and the motor, while the latter consists of the electronic skate controller, the board itself and an enclosure which will house the majority of our components.

III. System Components

As previously mentioned, our project hosts a number of onboard systems, relying on a suite of sensors to provide continuous information to both the user and the onboard power system for safe and reliable usage. The following sections consist of data on each of these systems, further broken down into each electrical component they consist of and any relevant information.

A. Microcontroller

The microcontroller is necessary for communicating between the mobile application and the system via a Bluetooth connection. Our original choice for this component was the MSP430F5229, which has an integrated serial bus and a physical layer supporting USB 2.0. It is known for its high performance 12-bit analog to digital converter, shutdown mode, off mode, standby mode, real time clock module, a hardware multiplier and 63 input/output pins. It operates on a low voltage supply ranging from 1.8 to 3.6 volts and has a system clock that can respond at a rate of up to 15 MHz. The reason this was our original choice was due to the fact that it came with a high number of I/O pins, which we would need for the number of peripherals we intended to incorporate in our design. Along with that, our two computer engineers had previously worked with MSP products and felt fairly comfortable using Code Composer Studio (CSS), the embedded system IDE that is used to program this board in C. During the design phase however, the use of this board became increasingly difficult and detrimental to the project as there was very little documentation on utilizing many of the components we had chosen.

In order to keep our project progressing in a timely manner, we decided to pivot to the Atmega2560. This chip also supports USB 2.0, having its own integrated serial bus and physical layer. It is known for its high performance, high endurance capabilities, a low-power Microchip 8-bit AVR RISC-based microcontroller which combines 256 KB ISP flash memory, 8 KB SRAM, 4KB EEPROM, 86 general purpose I/O lines, 32 x 8 general purpose working registers, a real time counter and six flexible timer/counters with compare modes. Along with all of that it also has a 12-bit analog to digital converter. In addition to the 86 general purpose IO lines and 100 total pins, the Atmega2560 operates on a frequency scale of 0 to 16 MHz, a system clock that can response at a rate of up to 15 MHz and a voltage range of 1.8 to 5.5 V. This chip consumes less than or equal to 1.8 V making it very suitable for ultra-low power consumption applications. The reason we chose the ATmega is largely based on the number of I/O pins and UART lines it offered.

B. Battery

Our choice of battery is a lithium-ion battery. The reason for this choice is that they provide the greatest electrochemical potential and energy per weight. This battery is charged on an algorithm with a high capacity, low internal resistance and low self-discharge. This makes this battery ideal for our design and we ultimately settled on a 36V battery with a 10S2P7 configuration capable of providing power with a range of 800 to 1200 watts. This battery also has a charge current of less than 10 amperes. This is ideal for all of the other components we have which a much lower max amperage as well as a nominal capacity of 7 to 10 amp-hours.

In order to view the status of the battery we've decided to use a simple battery meter that will read the status and display the result on a screen.

C. Motor

A DC motor is an electrical device that serves the purpose of converting electrical energy into mechanical energy. It does so by taking the voltage that is applied to its armature winding through carbon brushes that are on the commutator, which is an electrical rotary switch. A specific kind of motor, known as a brushless DC or BLDC motor, does not use brushes for commutation but instead does the commutation electronically. These types of motors are widely used due to the speed to torque ratio they provide, their high dynamic response and efficiency. Due to our environment and the lack of hills in the area, a high torque motor was not required so our group decided to go with a brushless DC motor in order to be able to push our top speed towards the higher end of the range.

The next thing to look at was whether we would opt into using a belt or hub motor. Some of the perks of a belt motor are that it provides more torque, a full-sized wheel can be used which allows for a smoother ride and they are easier to perform routine maintenance on due to the ease of accessibility of the components. However, hub motors tend to be quieter as there is no belt/pulley system to create friction and noise. As such, they produce less drag, reducing the amount of drain on the battery, and they tend to generally weigh less. Hub motors also allow the user to free-roll, meaning once the battery has dies, they can continue to ride the board in a "kick-push" manner whereas a belt motor generates too much friction for that to be possible.

With these things in mind, our team began searching for a motor and ultimately, we decided on a 90 mm Hub Motor Kit from MBoards. This kit uses two 6364 hub motors which can provide speeds of up to 25 mph and comes with a matching pair of front wheels (non-motorized) along with an extra polyurethane sleeve. The polyurethane sleeve is used for replacing worn down wheels which happen more often with hub motors. This due to the motor being integrated into the wheel and taking up a significant amount of space leaving the actual wheel to be quite thin and more prone to wear and tear.

D. Longboard

Although selecting a longboard seemed like a simple task at first, since it is the most basic of our hardware components, we quickly learned that our board could have great implications on the way riders can effectively maneuver the board. When selecting a board, a few key features that should be looked at are the length and width of the board, its flex, style, and shape. These various aspects factor into both the maneuverability and the stability of the board, with each one sacrificing either the maneuverability or the stability for the other depending on the design choice made. The flex factors into this in a less direct manner as it deals with the board's ability to absorb shocks but isn't as easily measured as a factor such as length or width.

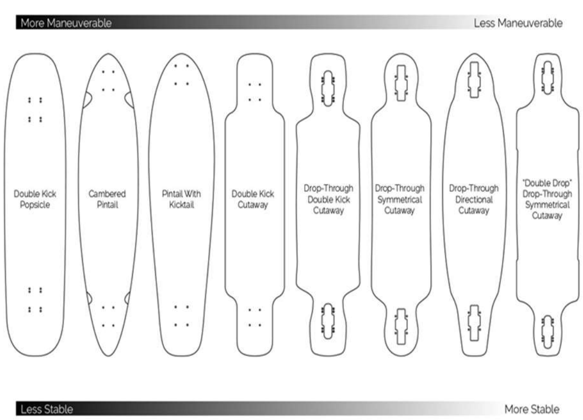


Fig. 2. Longboard Shapes - Maneuverability v. Stability.

As can be seen in Fig. 2, long boards come in various shapes and a couple of big points in our choice for one also came down to the amount of space we would have below the deck to mount our components and an unwillingness to give up maneuverability. Ultimately, we decided on the Retrospec Zed Bamboo Longboard, which is a 41 in. board and provides us with 26 in. of space from axel to axel. The board also tends to ride high which gives us plenty of space underneath to mount our components, along with an enclosure, without them hitting the ground or being dragged. The board sports a cambered pintail design which allows us to still maneuver fairly easily, especially given the context of where we believe this is best applied but the length and width provide the user with a sufficient amount of stability.

E. Printed Circuit Board

The Phoenix Board's printed circuit board (PCB) was designed using the Eagle software which allows for the creation of schematic designs and a smooth transition into the board layout. One obstacle that was faced when designing the PCB was that Eagle did not have all the necessary libraries for all of our components and various imports were needed. Upon completion of the schematic, we transitioned into the board layout where all of the wires were manually rerouted to ensure everything was connected correctly.

Initially, we started out using the eagleCAD software for our design but later faced several challenges, eventually reaching the point where it was taking a great deal of time and effort with very little return on investment. This was mainly due to the lack of support that was available for the MSP430F5529, our original microcontroller. Once we made the switch to the Atmega2560, we also made a switch to easyEDA, which made the design process much smoother and we were able to complete the schematic and board file.

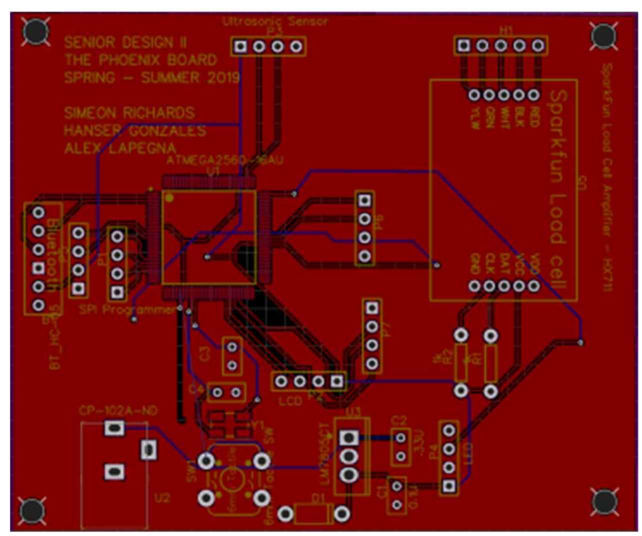


Fig. 3. Phoenix Board PCB board layout.

Seen in Fig. 3 is the PCB schematic, which was converted into a board file that used to fabricate the board. Once the fabrication was complete, we moved forward to soldering the components and testing to ensure the proper working conditions were met. This process has been somewhat daunting with a factor of uncertainty being involved upon receiving the PCB and whether it will actually perform or not. In order to mitigate some of these worries we used precautionary measures such as starting out with a lower voltage and gradually increasing that voltage until we were at 12 V to the input as the PCB was designed.

The first PCB vendor we used for the project was ALLPCB, a fabrication company based in China, which provided a low cost but resulted in us having to wait two weeks until our PCB finally arrived. These two weeks included multiple attempts to get in contact with them via phone call and email to no avail. For our second and third orders we went with JLC PCB, another company based in China, but we had a much better experience with this vendor and received our PCBs in a timely manner.

F. Electronic Speed Controller

An electronic speed controller (ESC) is needed to control our hub motors and interface with the PCB. An ESC is used to provide a pulse-width modulation (PWM) signal to the motors to increase/decrease the throttle. We are using one based on VESCs, which is an ESC designed by Benjamin Vedder. His original design is open source and includes a robust software suite known as the VESC Tool. The model we are using is the Flipsky Dual FSESC4.20 100 A VESC. It can support batteries ranging from 3S to 13S and provides up to 100 A of continuous current in dual motor mode. Each motor is connected to the VESC via three motor signal lines and a Hall sensor connector, totaling 6 signal lines and 2 sensor connectors. It has a USB port for easy configuration and testing using the VESC Tool software. Lastly, the VESC has a UART connection which we use to get the speed instructions from the PCB.

IV. Safety Suite

With our project involving a motor and being suited for use in populated areas, safety was a big concern for us from the beginning. With this in mind, we decided on a suite of features and components that would enhance the safety of both our rider and anyone in their vicinity. These features are encompassed by both hardware and software functionality but the following section will specifically cover the hardware.

A. *Ultrasonic Sensor*

A key safety feature on the Phoenix Board is the ability of the board to detect incoming obstacles and in the event that there is no additional user input to attempt to avoid it, the board itself will safely slow down to an emergency stop. For this we will be using an ultrasonic sensor, which utilizes a transducer to emit ultrasonic waves to measure the distance to an object at any given moment. The transducer sends the pulse out and receives the echo using (1), with L being the distance, T being the time and C being sonic speed.

$$L = \tfrac{1}{2}\, T\, C \tag{1}$$

For our project we decided to use the SMAKN Ultrasonic Module HY-SRF05 Distance Sensor. This sensor has detection range of 0.75 to 14.75 feet (or 2 to 450 cm) which was an important metric for us. This is because since we are going to have the Phoenix Board travelling at high speeds, we want as much time and space as possible in order to slow down the board without jettisoning the rider off. This module operates at +5 V DC with a quiescent current of < 2 mA. The effective angle is 15° with a resolution of 0.3 cm and a full viewing angle of 30°. The module also has a 5^{th} OUT pin which allows it to operate in such a manner that ECHO and TRIG run off of the same and effectively save on an additional wire, however this does put more strain on the processor.

B. Weight/Load Sensor and Load Cell Amplifier

For this project we wanted to incorporate the rider's weight in two key ways. We wanted to be able to factor in the user's weight and with it, along with the amount of charge left in the battery, calculate the distance that could still be traveled. Unfortunately, due to the time constraint of this project we will no longer be implementing this as there are many more factors that would go into this calculation in order to make it accurate. The second use of the weight sensor, and the one we are still utilizing, is as a safety switch which allows the board to know whether the user is still on or not. In the case that the user is not on to begin with, the board will simply lock out any incoming transmissions and will not accelerate. Should the user be on the board and then fall or jump off for whatever reason, the board would register this and would bring itself to a stop in order to mitigate any damage to nearby pedestrians and to itself.

For our load cell we chose the TAS606, which can translate up to 440 lbs (200 kg) of pressure into an electrical signal. With this, we set up a simple threshold within our software in order to make sure a person would register as being on the board but a random object with some weight would not which could potentially put others in harm's way. This component is accompanied by the HX711 Load Cell Amplifier, which is a small breakout board that would allow us to properly read the data coming from our load cell with relative ease. The amplifier uses a two-wire interface for communication and can work with virtually any microcontroller's GPIO pins. Along with that there are several libraries already written for it that would allow us to integrate this component effortlessly.

C. *Light Emitting Diodes (LEDs)*

In order to provide riders with a safer night-time experience we wanted the Phoenix Board to incorporate LEDs. These are set up underneath the board and provide a steady source of light for the rider so they can be easily spotted in times or areas of low visibility. We considered three types of LEDs for this project: single-color/non-addressable, RGB/non-addressable and RGB/addressable. Addressable LEDs provide a greater range of control for the user as each diode has its own chip which allows for each one to be controlled individually in terms of color and intensity (brightness), whereas non-addressable LEDs act as a single unit, sharing the same values for color and intensity throughout the strip.

For our design we went with the CHINLY 16.4ft WS2812B individually addressable LED Strip. This LED strips is convenient in that it runs off of +5 V and only requires one Data In pin to run the entire strip. The color and intensity are set using PWM, which we were able to easily implement using the FastLED Arduino Library. In total we are using 40 LEDs to provide the night-time feature for the board.

V. Wireless Technologies

From the onset of our project we knew we would be working with wireless technologies due to our desire to get some experience with this technology but initially we believed this would only encompass our mobile communications. While we did still implement a wireless technology for this, we also chose to utilize it in our power system in the form of inductive charging. The application for both of these uses is detailed below.

A. *Mobile Communications*

The Phoenix Board demands a strong and consistent communication link between the board and the mobile application in order to ensure there is full control of the ride at all times. Our main options for this were Wi-Fi and Bluetooth as they were the most practical in terms of the usage that we needed but they were also designed for very specific and different purposes. The abbreviated version of these differences boils down to Bluetooth being effective at short-range and being ideal for pairing one device with another, while Wi-Fi is more suited for longer ranges and works with multiple devices on a network.

We originally went with the CC2650 Bluetooth Module As its name implies, it is a simple way to add Bluetooth low energy capabilities to a LaunchPad development kit. The kit, which is programmed with wireless network software, allows you to add BLE to any application with a UART interface. The module is Bluetooth 4.3 specification certified ad is also pre-certified for FCC/IC, CE, and ARIB radio standards. However, we were running into major implementation issues where we were unable to get the module working. As a result, we shifted from BLE to a Bluetooth Classic enabled module.

Our final implementation uses a DSD Tech HC-05 Bluetooth Serial Pass-Through module. It contains 6 pins, +5 V, GND, TX, RX, EN and STATE. The module is connected to one of the microcontroller's UARTs and runs at a default baud rate of 9600, which allows for communication between our app and the PCB. Using the HC-05, the PCB receives the current desired speed and LED configuration data, while transmitting a speed reset flag to the app in the case of an emergency stop, triggered by either a rider stepping off the board or an obstacle being detected.

B. *Inductive Charging*

For the charging of the Phoenix Board we decided to implement inductive charging which allows for higher efficiency due to the closely coupled system. In some instances, this would mean that the construction of the charger must be done using concept of a transformer, which is split into two parts. The primary winding is housed in a unit connected to the main power supply, while the secondary winding is housed in the same sealed unit which contains the battery along with the other electronic components.

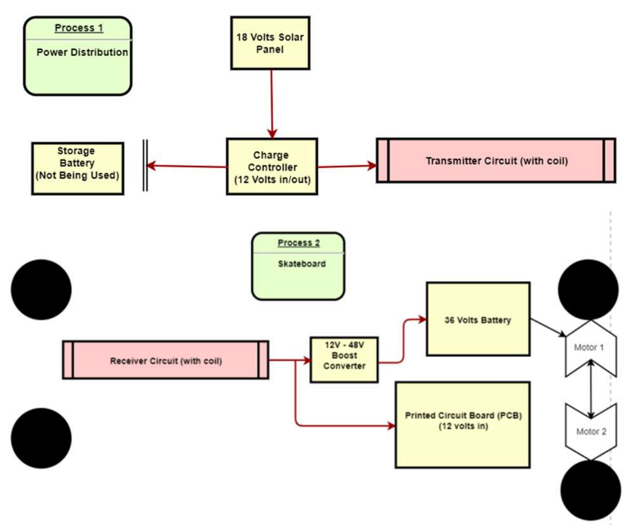


Fig. 4. Wireless charging (inductive circuit) diagram.

For our design, we are using this method of charging to add a more unique and efficient feature. As seen in Fig. 4, inductive charging makes use of a transmitter that will be a part of the circuit connected to the solar panel. Along with that, there is a receiver that will be installed on the board, and through the use of these inductive coils we will send and receive electromagnetic power. In order for this to be an effective method, a constant voltage and current is required in order to keep our lithium-ion battery from being damaged if the upper voltage limit is exceeded. For this reason, our charger will be capable of controlling both the current and battery voltage.

VI. Solar Charging

According to the Solar Energy Industries Association (SEIA), solar energy is the cleanest and most abundant renewable energy source available. The solar market has grown significantly over the last 5 to 10 years, with an average growth rate of 50% each year. However, there are two major disadvantages with solar charging, the cost of the equipment and the variability of sunlight that can be received due to both adverse weather and also the angle at which the sun is hitting the solar panel.

A. *Monocrystalline Solar Panels*

Monocrystalline panels are the oldest and most thoroughly developed of all the solar technologies. It is a type of PV cell material manufactured from a single crystal silicon structure that is dipped into molten silicon and is slowly pulled out of the liquid, producing a single continuous crystal ingot.

Monocrystalline panels are made using the highest-grade silicon and therefore have high efficiency rates. These panels offer a longer lifespan with some manufacturers offering a 25-year warranty on their products. Although these monocrystalline panels tend to be more expensive, they have the ability to outperform their competitors and for that reason we chose these for our project. For our project, we chose to use an 18 V solar panel for the main reason of ensuring that we have enough voltage going through the circuit.

B. *Linear Voltage Regulator*

Linear voltage regulators use an active pass device that is controlled by a high gain differential amplifier. It compares the output voltage with a precise reference voltage and adjusts the pass device in order to maintain a constant output voltage. It is very effective and is commonly used when designing low-power circuits by using power transistors, BJTs or MOSFETs that play the role of a variable resistor, raising or lowering the output voltage of the circuit as the input changes.

The linear regulator power dissipation is directly proportional to its output current for a given input and output voltage, so typical efficiencies can be 50% or even lower. Using the optimal components, a switching regulator can achieve efficiencies in the 90% range. However, the noise output from a linear regulator is much lower than a switching regulator with the same output voltage and current requirements. Typically, the switching regulator can drive higher current loads than a liner regulator. Ultimately, linear regulators are preferred for low-power applications and that is why we chose to use one in our implementation. The use of the voltage regulator comes into play mainly with the inductive charging circuit and on the PCB. A 12 V regulator is used on the transmitter coil circuit to stabilize the voltage coming in from the solar panel and keep it steady, thus protecting the circuit from voltage spikes. For the PCB, the regulator is used for a similar function, only now it is reduced to 5 V and will serve to protect the PCB circuit from voltage spikes.

C. *DC-DC Converter*

A DC-DC converter is an electrical circuit that takes any direct current input voltage and converts it to another direct current voltage. This is widely used in electronic systems with different applications to convert higher voltages to lower ones and vice versa, and is a crucial part of our project. The goal for designing this converter is to successfully have this device reducing the voltage from 12 V to a steady operating voltage of 5 V that will be required to provide power to the sensors and other components.

VII. Software

The software that went into the Phoenix Board is multifaceted as it encompasses both programming for a mobile application and embedded programming for all of our hardware components.

A. *Mobile Development*

In our original search for an IDE to develop our application on, we split our research into two tracks. The first was native application development for both Android and iOS and the second was explicit, Android-only development. We decided to go the route of native development in the hopes of producing both an Android and iOS version, although we ultimately only developed our app for Android because of the cost for an Apple programming license.

For our framework we decided to use React Native, a JavaScript framework used to write applications natively on Android and iOS. Originally developed by Facebook, it allows the use of both JavaScript and React for mobile application development. It uses React's JSX syntax for building the app's user interface. JSX syntax allows for inline JavaScript code to mimic HTML syntax, allowing for the familiarity of HTML design and the ability inject JavaScript into the application without having to call an outside JavaScript file. Unlike React which is comprised of web components, React Native utilizes native components which will run natively on both platforms, although some platform-specific development is needed. With regards to the project, React Native also has access to Bluetooth, Location Tracking and Mapping API files that will provide the user with a variety of features and functions that will not only aide, but enhance their rides.

B. *Mobile Design*

For the most part, our original concepts for the Phoenix Board application stayed consistent throughout development but there were some features that were improved upon or completely removed.

One key feature that evolved throughout our development was how the app controlled the board's speed. Originally, we had thought of giving the user an option between a digital and an analog control that would be displayed on the screen but this would give the user a reason to take their eyes off of any incoming traffic and presented a hazard. Instead what we did was use the phone's volume buttons in order to accelerate/decelerate the throttle on the motors. Although this method only gives the user one way of controlling the speed, it is much safer because the user can simply feel for the buttons instead of looking away from what is in front of them.

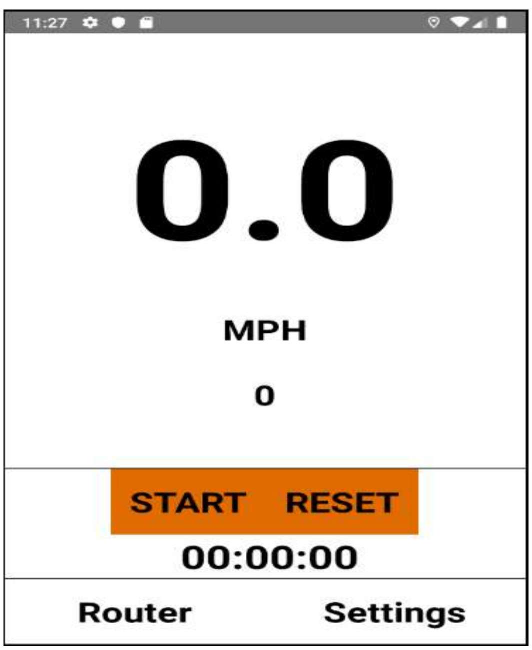


Fig. 5. Phoenix Board Application Home/Main screen.

The current speed will be displayed quite largely on the main screen, as shown in Fig. 5, and the speed display can be toggled to either MPH or KPH in the settings.

Along with that, the Home screen will display that Start/Stop and Reset buttons for a stopwatch that tracks the duration of a user's ride. This also factors into the average speed calculation that is displayed and saved on the user's map/route tracking screen.

On the topic of the tracking screen, this will be used to map out a user's route when operating the board and will be saved locally for future reference. For this we will be using React Native's Location Tracking API which uses geolocation as well as several native functions to display the user's current location. On this screen we will also be displaying the distance that was travelled during the user's trip which could serve them in selecting shorter routes in the future.

The last major screen in our app is the Settings screen, which gives the user control over a variety of different customizable options. As previously mentioned, the user can toggle between displaying their speed in miles or in kilometers and this will propagate throughout the app, displaying the change in the Home screen, the Router screen and also behind the scenes in the speed caps set for different rider levels.

The speed caps mentioned above are yet another safety feature of the Phoenix Board that we have dubbed "Ride Modes". Our application provides four different ride modes, each providing a different top speed that can be reached and also giving/denying access to other functions of the board. At this time, we only have the ultrasonic sensor functioning at the "Beginner" ride mode as it provides the lowest top speed and gives us enough time to safely slow and stop the board without jettisoning the user off. These ride modes are implemented in order to provide riders of any experience level a chance to utilize our board in a safe manner while still having fun.

Also found in the Settings screen is the LED screen that allows the user to select between eight different colors to give the user some customization on their night-time rides.

C. *Embedded Development*

When we were originally working with the MSP540F5229, we started developing software using Code Composer Studio v9. However, with the wide variety of sensors we planned on using, it became too time consuming to try and integrate existing libraries and/or create them from scratch for operation. The vast majority of our parts had extensive documentation geared towards Arduino development and therefore, it proved to be easier to switch to Energia, which allows for developing code in an Arduino style IDE but for TI products, utilizing existing Arduino libraries. However, even this was temporary. With the change in architectures to the Atmega2560, we were able to move onto the pure Arduino IDE. Due to the fact that we had already written extensive code in Energia using existing or slightly modified Arduino libraries, porting and getting our code functional was relatively simple. As such, development progressed much more smoothly due to the wide availability of libraries and documentation.

D. *Embedded Design*

The PCB, located on the main board itself, has an internal start-up procedure, which runs a brief diagnostic to ensure that all sensors are working properly and are ready for use. Once the mobile app is opened, it will attempt to connect via Bluetooth to the PCB. After this is successful, the app is ready to begin transmitting either LED configuration data and/or speed settings to the board. If an LED configuration is sent, it can enable/disable the LEDs and display one of eight different color schemes.

The board, before beginning any movement, checks the load cell to ensure that a rider is currently on the board before allowing any changes to be made. From here, any speed change is sent via Bluetooth, read by the PCB and sent via UART to the VESC where the motors will begin to throttle up/down as necessary. Board operation will continue as desired by the user, with the exception of an emergency stop flag being triggered from one of two sources. These include the load cell detecting a sudden loss of weight, indicating the rider has stepped or fallen off the board before coming to a full stop, or from the ultrasonic sensor detecting an obstacle in front of the rider.

VIII. Testing

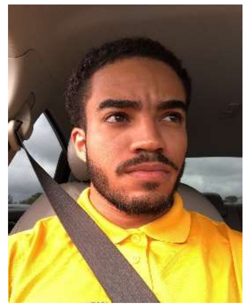
Each component in this project was thoroughly tested in order to ensure proper functionality. Our first string of tests on these components was done using the Atmega2560 development board and a breadboard, a system which was then redesigned as our PCB. Following the breadboard testing, we transitioned to having the PCB tested and implemented so that we had accurate communications with all the devices that needed to be connected to it. These devices included the Bluetooth module, the load cell, the load cell amplifier, the ultrasonic sensor, the VESC and the LEDs. Along with these sensors the power systems were also tested in order to ensure the battery was properly distributing its charge and no components were being overloaded.

To properly test out the mobile application we went back and forth between an emulator provided by Android Studio and a physical device owned by one of the project's engineers. Testing for some of these features, such as the location tracking, required the physical phone in order to perform live tests and get back real-time results. Along with that, all Bluetooth testing was done on the physical device as Android Studio's emulator does not provide Bluetooth functionality.

IX. Conclusion

The Phoenix Board is the culmination of various ideas brought forth by the engineers working on it and ultimately the project falls into two systems: power management and a mobile app. Within these two systems are various other subsystems, many interacting with one another in order to produce a smooth and fun riding experience for any user. With these systems working in tandem, we hope to produce a new and innovative produce, the likes of which have never been seen before.

Engineers



Hanser Gonzalez will be graduating from UCF in the summer of 2019 with Bachelor of Science in Computer Engineering. After concluding a 4-year long career in the UCF/Lockheed Martin College Work Experience Program he is looking to move on to full-time work within his field with a focus on software.

Alex Lapegna will be graduating from the University of Central Florida in August 2019 with a Bachelor of Science in Computer Engineering. Alex's interests lie in Machine Learning and Computer Vision and with the ultimate goal of having a career within those fields.

Simeon Richards will be graduating from the University of Central Florida in August 2019 with a Bachelor of Science Degree in Electrical Engineering. His main area of study is Power Systems and Renewable Energy, and currently works as an intern with Interplan LLC, designing electrical and lighting systems.