The Backpack E-Skate

Joshua Andrews and Danner de la Rosa

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

***Abstract* — This paper describes the methods used to develop an electronic skateboard. The implementation of certain technologies, such as brushless motors, passive infrared sensors, and the use of lithium-ion batteries made the project possible.**

Index Terms* — Bluetooth, Brushless Motors, Infrared Sensors, Light Emitting Diodes, Lithium Batteries, Microcontrollers, Voltage Control

I. INTRODUCTION

In recent years, many new short-to-medium-distance transportation devices have become available such as scooters, electric scooters, bicycles, electric bicycles, skateboards, electric skateboards, and hoverboards. These are all great ways to go to the neighborhood's park or to go from class to class while in school; they save time and energy, especially since some classes may be just 10 minutes apart and very far from each other. However, these devices are controlled by humans and rely on human instinct. In addition, they required an inconvenient amount of space. Imagine if all the students in a class were to ride skateboards to class; students leave their skateboards against the wall, which can make 30 to 200 skateboards a very messy process to pick up one by one.

The Backpack E-Skate project is a foldable electric skateboard. The goal was to design an electric skateboard that would provide convenience, fun, and safety for the rider. We achieved these conditions in various ways. For safety, the Backpack E-Skate makes use of light sensors to automatically control LEDs to increase visibility in the dark. It also uses motion sensors to provide warnings in case of pedestrians and unexpected obstacles. The E-Skate is controlled with a remote with simple controls to make operation intuitive. Easy operation increases safety by eliminating distractions. An Android application was deployed, which is used for the rider's enjoyment by customizing the RGB lighting sequences. This application communicates with the E-Skate via Bluetooth. As far as convenience, the motors make it convenient to ride and the folding design makes it convenient to transport when not being ridden. In addition, it is made to fit standards set by other electric skateboards on the market, such as battery life and capable travel distance.

II. SYSTEM DESIGN OVERVIEW

The e-skate as a whole is a collection of subsystems that work together to achieve one or more objectives. These subsystems are categorized as sensors, LEDs, Bluetooth module, microcontroller, electronic speed controller (ESC), hub motor kit, battery, PCB, mobile application, and remote. Figure 1 shows how these systems fit together to make the backpack e-skate.

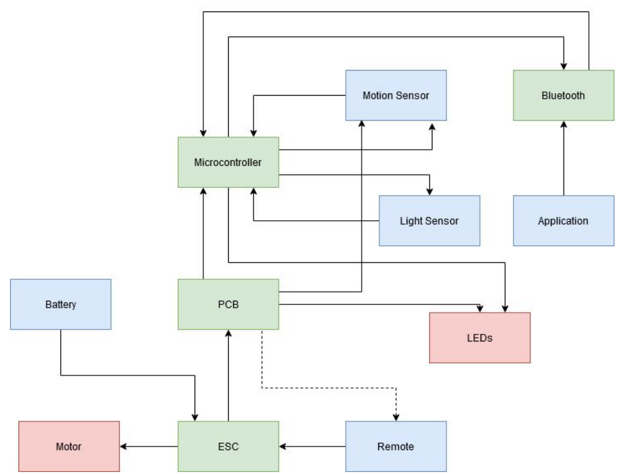

Figure 1: Integration of subsystems.

When the user presses a color sequence button, the mobile application sends the corresponding serial data to the microcontroller through the Bluetooth module; the microcontroller uses the program to execute the proper block of code and trigger the right LED sequence. The sensors work the same way, as inputs, but they are automated to work in specific situations.

On the other hand, the remote sends a frequency to the ESC, which executes the code to make the motor spin. A pin from the ESC supplies 12V, which constitutes the input voltage of the PCB to power the microcontroller, the motion sensor, and the LEDs. The battery itself powers the ESC, which supplies power to the motor and also has a charging port for the remote.

III. SAFETY

The difference between an electric skateboard and the backpack e-skate is primarily the ability of the e-skate to detect certain variables of the environment and provide the necessary adjustments in order to provide an excellent riding experience that prioritizes safety.

When one thinks about safety in the world of vehicles, the first thing that comes to mind is sensors. Sensors are a key safety feature, since they read environmental measurement values that allow a system to trigger change in order to adapt. In the category of popular sensors used on transportation are light sensors and motion sensors. The e-skate employs these two types of sensors, along with LED lights, to provide excellent safety as a standard.

The light sensor introduces a concept known to automobiles for a long time, automatically turning the lights on at dusk. For a skateboard rider, this is also important; there are many riders who ride on the sidewalks of busy streets and cross intersections as well. When it gets dark, drivers need to be aware of people, and the e-skate does that! In addition, automatic night lights provide more visibility for the driver, allowing them to be aware of any obstacles on their paths and avoid them, preventing accidents that could end up on a road with heavy traffic. The effective brightness at which the lights turn on is custom and controlled through the microcontroller; it can be any value desired.

The motion sensor is important for the opposite; instead of making others aware of the e-skate, the motion sensor makes the e-skate aware of others, allowing the rider to react in time and avoid crashing into someone walking by. After looking at the documentation, the effective distance of detection is very good at about 20 feet at the front, which is the most important direction. This distance allows the rider to see the warning signal in time and react appropriately.

While sensors are the input, there could be different outputs, such as LED lights. The e-skate is packed with LEDs all around and under the board and possesses a potent, warm-white headlight, specially designed for nighttime. When the night comes, the LED strip turns on just like the headlight; the pattern of the strip is a solid, bright white to maximize visibility. If movement is detected ahead, the LED strip flashes red and blue, much like the police; this warning signal is much more effective because it calls for attention in a more urgent manner.

IV. MICROCONTROLLERS AND BLUETOOTH

The Bluetooth module HC-05 used for the project was first tested using the MSP430G2553 development board; it is easily connected after analyzing its pinout. The STATE and KEY pins were not needed; the VCC and ground pins are directly connected to a VCC and ground pin on the microcontroller. Pins TXD and RXD connect to the UART pins 1.1 and 1.2 respectively.

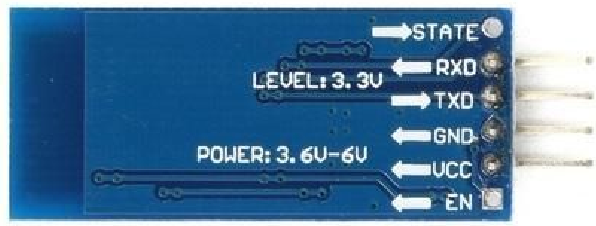

Figure 2: Bluetooth module HC-05 pinout.

After successfully connecting the module, we can simply use the polling technique to check when any incoming information is available. These are the serial functions that were necessary to implement this code.

Function	Action
Serial.begin();	Enables serial communication
Serial.end();	Disables serial communication
Serial.read();	Reads incoming serial data
Serial.available()	To get number of bytes available

Table 1: A table that shows the serial functions necessary to implement the Bluetooth section of the code.

The polling technique works with Serial.available() inside the infinite loop that constitutes the main function of the code. When the function returns anything greater than zero, the bluetooth module has sent serial data to the microcontroller. Then the function Serial.read() can be used to identify the data received and save it to a character variable. From here, the code can include a variety of blocks that check the character to execute each particular block when needed.

When we removed the MSP430 microcontroller from the development board, we discovered an issue with the voltage levels between the MSP430. Originally, we attached the MSP430 to an LM317 regulator adjusted to 3.3V. The Bluetooth module could not be connected to this regulator because it needs at least 3.6V. However, when we attached the Bluetooth module to a higher voltage separately, there was trouble with the MSP430 and the module communicating since they had different supply voltages. The solution was to adjust the LM317 to an output of 3.75V and attach both the MSP430 and the Bluetooth module to this voltage. The circuit design for this voltage regulator can be seen below.

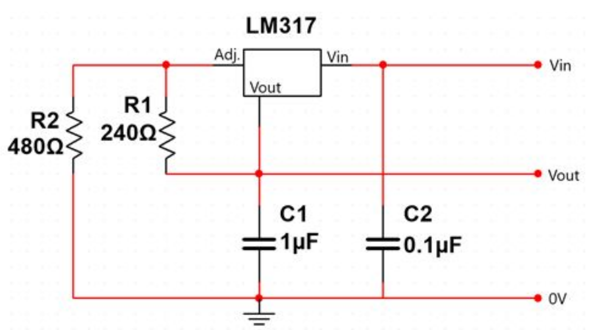

Figure 3: This is the design of the circuit to regulate the voltage for the MSP430 and the Bluetooth module. Vin can be any value, but in this case it connects to the 13V output of the ESC. With the given resistor values, the Vout is adjusted to 3.75V, which is a suitable Vcc value for the MSP430 and the Bluetooth module. In fact, the 3.75V output was also suitable for the motion sensor.

V. ANDROID APPLICATION

The android application needs to send characters to the microcontroller; the microcontroller will use the code to execute the right action. For applications as such, the MIT App Inventor 2 is a convenient tool that allows applications as such to be developed. This tool is divided into two sections, graphic user interface (GUI) development and backend and execution commands in the form of blocks.

Using the GUI development section, the team generated a list picker, a label, and 10 buttons, along with a background; all these things were organized through horizontal and vertical arrangements found in the layout tab. A descriptive image was chosen for all the buttons and the list picker in order to enhance the appearance of the application. The picture that follows shows the GUI.

Figure 4: The mobile application's GUI

Pressing a button triggers the Bluetooth module to send data according to the blocks. For example, the blocks in Figure 5 indicate that when button 0 is clicked, the Bluetooth client is called, and the character 0 is sent through. Likewise, the blocks indicate that before picking from the list picker, its elements will be set to the addresses and names available to the Bluetooth client; after an address selection has been made, its corresponding Bluetooth module will be connected. The entire set of code blocks can be seen in Figure 5 shown in the next page.

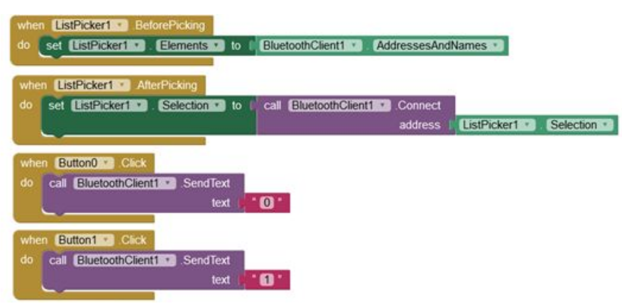

Figure 5: The mobile application's backend blocks of code for the list picker and the first two buttons.

VI. HUB MOTOR KIT

For the motor and wheel kit, we chose the "75mm 350W Brushless Hub Motor Kit" from Amazon. This model is single drive and is affordable. We originally bought a more expensive dual hub motor kit from EBay, but this arrived damaged due to poor packaging by the EBay seller. To account for the hit to the budget, we went with a cheaper replacement. This kit still performs well, achieving a top speed of about 16 miles per hour, which matches our target. This kit also has the benefit of using brushless motors, which has energy efficiency advantages over regular DC motors. In typical "brushed" DC motors, the motors are spun internally using electromagnetic brushes to create a magnetic field. However, these waste energy because the brushes create a lot of friction inside the motor. Brushless motors work instead by using an electronic speed controller (ESC), which externally generates the magnetic field by cycling voltage periodically through three input wires, usually blue, green, and yellow.

Figure 6: This is the hub motor kit we chose.

VII. ELECTRONIC SPEED CONTROLLER

In the original design, we planned to use a PN Half-Bridge to control the speed of the motors by regulating the voltage. This would have worked with regular DC motors, but the brushless motors require an ESC to generate the magnetic field inside the motors. To match with the motor kit we chose, we went with a 24v-36v Single Drive ESC. This one in particular was rated for up to 650W, which gave us plenty of room to safely operate our 350W motor kit. This electronic speed controller in particular was convenient because it came with a battery indicator, power button, and even a remote. The handheld remote that came with the ESC had a simple and intuitive design to drive the E-Skate. We started to think that using the Android application to drive the E-Skate would be a dangerous distraction, so the fact that this ESC came with a remote allowed us to focus on just controlling aesthetics with the app, namely the LED functions.

Analysis of the ESC's circuit board allowed us to find a set of pins that output 13V when powered on. By making solder connections between these pins and our circuits, we were able to power the rest of our components, including the LEDs, sensors, Bluetooth module, and the microcontroller.

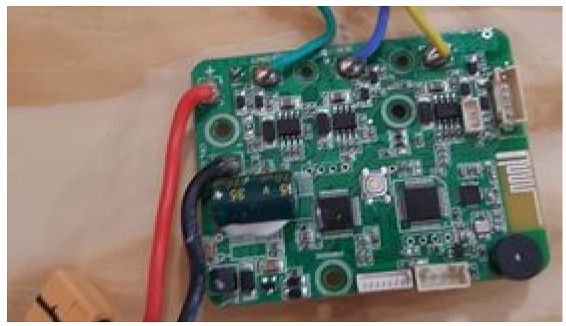

Figure 7: A figure that shows the electronic speed controller we chose. The cables at the top are the power cables of the brushless hub motor utilized; the cables at the sides are to connect a power source.

VIII. BATTERY

When going over our original design, we realized that our battery choice was unrealistic. First of all, using two lead-acid batteries would be very heavy. Aside from making the E-Skate less portable, it would also decrease the travel range because extra energy would be wasted to transport the extra weight of the batteries themselves. Second of all, we chose 12V batteries, which is low. Anything other than the lowest-end electric skateboard motors have input requirements of at least 24V. With these things in mind, we opted for a 7S2P Lithium-ion battery. The code stands for "7-series 2-parallel," indicating the battery is made of seven Lithium cells connected in series to provide the needed voltage, and then two of these sets are attached in parallel to double the capacity. This battery has a nominal voltage of

25.2V and goes up to 29.4V on full charge, making it perfect to go with our motor kit and ESC.

Figure 8: This shows the battery we chose.

IX. MOTION SENSOR

Additionally to the main sensor setup designs that we had researched, there was a simpler, and still efficient way of implementing motion sensors on the project. Considering that the skateboard will primarily move forward, a PIR motion sensor mounted under the skateboard at the front would detect movement ahead, but also in a semicircular area once a dome has been mounted on the sensor. This is ultimately the design procedure we followed, given that it would not have served much of a purpose to install sensors on the sides and/or the back; those installments would work better for cars and other bigger vehicles, with closed cabins that limit visibility.

Figure 9: PIR Motion Sensor

Figure 10: PIR Motion Sensor Dome

The sensor selected included a circuit board and a dome, facilitating the sensor's installation for our team, which has no electrical engineer. This selection was also ideal considering that a bare sensor alone could be priced much higher; it also takes away the need to buy components and put them together in a circuit on the PCB, minimizing work, while maximizing cost efficiency and time. One other great attribute of this selection is the dual potentiometer system installed on its board; it allows one to control the sensitivity and delay. Sensitivity refers to how much movement is required to trigger the sensor pin, while delay refers to how soon after a trigger the sensor is ready for the next trigger. The figure below shows the motion sensor selection, along with its pinout.

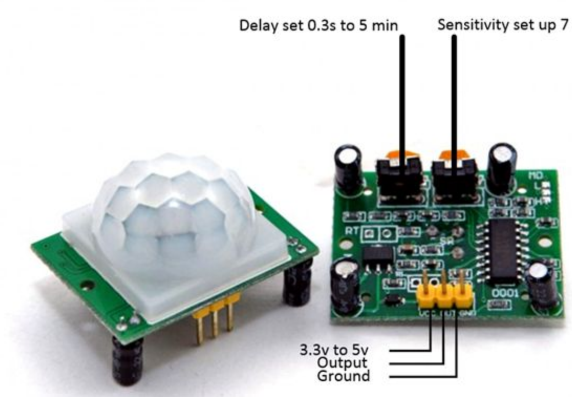

Figure 11: PIR Motion Sensor covered by its dome on the left; circuit board, including both potentiometers on the right.

X. LIGHT SENSOR

Unfortunately, we quickly discovered upon the arrival of our selected light sensor, model OPT3001DNPR, that this latter model is very small, and its pins are absolutely impossible to solder onto without the proper equipment, which must include very thin tips to prevent the pin nodes from merging. We speculate that it is produced for smaller objects, such as phones.

This dilemma took the team back to the research phase to look for project examples that used light sensors in similar environments. Then we stumbled upon the basic light sensor, a photoresistor. The photoresistor is just as good as any other sensor previously researched for this particular use case;it can also be read analogically or digitally using a microcontroller. Analog read allows our team to dim or brighten LEDs without turning them completely on or off; this functionality is not necessary though. Instead, we will read the photoresistor's output and decide at which brightness the LEDs will turn on or off. For example, the value that we decided to use was 50, which was good to perform our tests. When the environment's brightness dips below 50, the lights turn on automatically; similarly, the lights will turn off if the environment's brightness hits 50 or above.

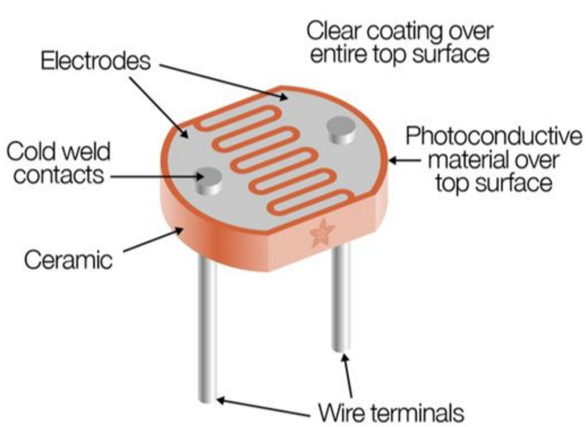

Figure 12: The components of a photoresistor.

XI. LEDS

Once the LED strip arrived, we were surprised to find out that we had bought a grand total of 12 LEDs, four strips of 3 LEDs each, at a price of about $8 to $12. This was a loss, since a whole roll of LEDs could have been bought from Amazon for about $20. We proceeded to test the LEDs received on the breadboard to make sure we understood how the circuit works with a microcontroller.

In order for a microcontroller to turn a 12V LED strip on or off, it requires the use of transistors; transistors behave like switches controlled by the microcontroller. First, the strip is connected from the 12V pin to the 12V source, which is a circuit that regulates the voltage of the battery, 25.2V. An LED strip has three different grounds, one for each color, which share the voltage in parallel; each ground is connected to a collector pin of a TIP 120 transistor, which were purchased from Skycraft at $1.50. The base of each transistor is connected to an output pin from the microcontroller. Passing enough current through this pin turns the transistor on. Oppositely, removing the current through the base will cause the transistor to turn off. At the emitter pin of each transistor, the ground is connected; when the transistor switches on, its LEDs connect to the ground, thus completing the circuit, and turning on the LEDs of the color designated to that transistor. Please refer to the schematics in Figure 14.

These schematics in Figure 14 include the headlight, which was implemented the same way in regard to the circuit. Figure 15 shows the location of the headlight and how it was made. The idea originated from the LED bars that have been appearing on trucks in the last couple of years; they are annoyingly bright and provide great visibility. During a trip to skycraft, a team member stumbled upon 2x3 LED warm white panels. We used three of these panels connected in parallel, all hooked to the same voltage source as the LED strip. When the headlight was finally ready, it was mounted with hot plastic, just like the LED strip. This can be seen in the section "Physical Construction".

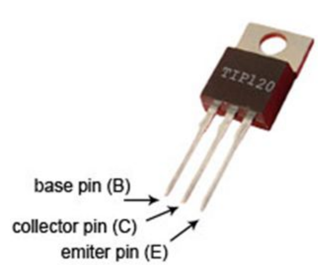

Figure13: TIP120 transistor pinout.

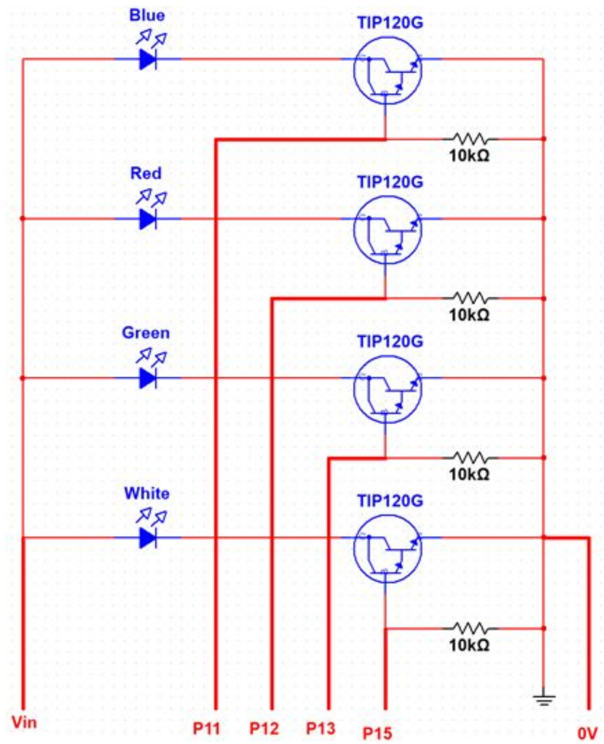

Figure 14: Circuit design for controlling all three colors of LEDs, one color per transistor and output pin.

After all the testing was done, we proceeded to get the LEDs. Around that time, a team member found a long LED strip sitting at his house; it was broken when removed from the desk on which it had been glued to, but that only constituted a loss of 3 LEDs in total. Then the time came to mount the LEDs, and we quickly discovered that working with an LED strip as a whole is an absolute nightmare, given that the E-Skate is foldable. Figure 15 is a more accurate representation of what the LED mounting diagram looks like in reality. Mounting the LEDs had another major complication; the weird bending of the strip caused the strip to start breaking in bad places, causing some colors not to work. Although we tried to take off the black vinyl, the PCB started breaking, and the team ultimately decided to cut the pieces out and resolder. Resoldering was also complicated; the soldering compound had a hard time sticking to the PCB, but ultimately, it ended up sticking when a thinner soldering tip was used.

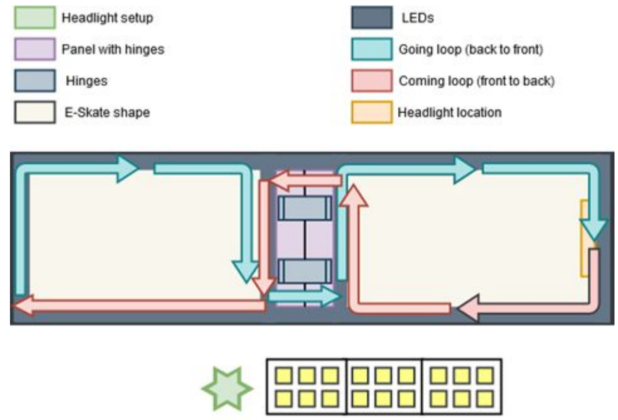

Figure15: Accurate LED system setup based on actual build

XII. PHYSICAL CONSTRUCTION

This section is concerned with the design of the deck itself, including the construction and the folding mechanism. The deck in total is 40 inches long, 10 inches wide, and ½ inch thick. These proportions can be seen in Figure 16. This was achieved by double layering pieces of ¼ inch plywood. Cutting pieces 20 inches long gives us the two sections that fold on top of each other. Figure 17 shows the bottom of the E-Skate, so the layout of components and the space management can be seen. Notice how the front wheels (left side of picture) are placed back several inches. This gives the space for the E-Skate to fold without the sets of wheels colliding. We show what the pivot point of the board looks like in Figure 17. Notice the durable steel hinges and the placement of the wires such that they sit below the surface level of the hinges. This allows the E-Skate to fold without pinching the wires, which could damage them. Finally, see Figure 19, which shows the shape of the folded E-Skate. Notice the reinforcement at the pivot of the E-Skate (right side of picture). This helps distribute the tension force caused by the rider's weight on the middle of the board, preventing it from snapping at the hinges or folding in the wrong direction. They also serve to limit the range of motion when folded completely. Otherwise, if the hinges were flush with the bottom of the deck, the folded shape would form a sharp angle that can crush the electrical components with enough pressure.

Figure 16: Top View of the E-Skate

Figure 17: Bottom view of the E-Skate.

Figure 18: View of the E-Skate in its folded form.

XIII. BUDGET

This section simply shows the costs associated with the Backpack E-Skate project. Only the parts that made it to the final design are shown. Anything we bought that we did not end up using are not included. For example, the original dual hub motor kit we bought is not included in the total cost because we ended up using a different model.

	Component	Cost
1.	Battery and Charger	$40
2.	MSP430G2553 w/ dev. board	FREE
3.	HC-05 Bluetooth 2.0 Module	$6.99
4.	Wood and Hardware (includes screws, hinges, etc.)	$45.62
5.	PCB (includes costs of various circuit components used)	$53.62
6.	Sensors (PIR and photoresistor)	$2.48
7.	Electronic Speed Controller	$50
8.	Single Hub Motor Drive Kit	$85
9.	RGB LED Strip	FREE
10.	White LED modules	$7.99
11.	Total	$291.70

Table 2: A table that describes the parts used and their costs.

XIV. GROUP MEMBER BIOGRAPHIES

Danner De La Rosa is graduating with a Bachelor's degree in Computer Engineering. He has enhanced his engineering experience through personal projects and an internship. He is excited to begin his career and plans to come back to UCF after a couple of years to earn his Master's. His passion is embedded systems.

Joshua Andrews is graduating with his Bachelor's in Computer Engineering. After his internship with the Navy at NAWCTSD over the last summer, he plans to go back after graduation as a full time employee. He has not yet decided to pursue a Master's, but he is waiting to see what the future holds. He is interested in computers at any level, but he particularly enjoyed working with the microcontroller in this project and incorporating it with the other electronics.