University of Central Florida
Department of Electrical & Computer Engineering

EZ-Scooter

Smart Scooter

Senior Design 2: Final Project Documentation
Dr. Samuel Richie

Group 14
Carolina Geary - Electrical Engineering
Jacob Jardines - Computer Engineering
Tony Naylor - Electrical Engineering

August 2nd, 2022

Table of contents

1 Executive Summary

The past few years have seen increasing public and government interest in energy issues in electromobility as a source of less environmentally harmful mobility. So, for this reason the usage of electric scooters in cities has increased in recent years. We believe that electric scooters have their segment in the transport networks of the present-day and future, offering first and last mile travel solutions that are inclusive and sustainable. However, ongoing battery technology has addressed many problems with limited travel distance between battery charging, charging time, high cost and the lack of safety for the consumer. Therefore, this project consists of designing an electric scooter capable of bringing safety, efficiency and to be more affordable than the already existing ones.

In this project we want to design and improve the normal electric scooter. Designing an electric scooter requires consideration of safety, comfort, and electrical objectives. The project began by analyzing existing products and components in the market and by sourcing components to meet the requirements and specifications commercially available for the main components of the electric scooter such as motor, batteries, speed controller, microcontroller and sensors. The system will also include features like a liquid crystal display (LCD) to display the speed and the life of the battery.

To achieve our main goals the system will have regenerative braking and the tilt sensor. In order to support the regenerative braking the right type of motor and speed controller had to be selected. The only motor that complies with the specifications that can be used for regenerative braking is the direct drive hub motor (non-geared motor). And the tilt sensor will give us a safety feature by determining when the scooter is on its side.

The electric scooter is an excellent option for mobility and represents a large piece of the problem in decarbonizing urban transportation. This project could be very helpful for future related projects to develop new smart technologies and develop a more proficient regeneration system.

2 Project Description

2.1 Project Background

The electric scooter is not a new concept. Electric scooters serve many purposes such as a mode of transportation, to make a living, or just simply as recreational purposes, but it can be expensive for the consumer. We want to create a system that not only is beneficial to the environment, but to the consumer by being less expensive than the ones that already exist in the market. To find a scooter with similar capabilities to our design, costs could easily exceed a thousand dollars. Our electric scooter will run on electric power by activating a direct drive hub motor that will facilitate the scooter's acceleration. It will also have the ability of self-charging. The scooter will have a motor on the rear wheel, a battery with a regenerative braking system as well as a wall charger, a LCD display, as and added safety feature several tilt sensors will be used to determine when the scooter is on it's side and to prevent the activation of the motor, the speed controller will support regenerative braking and will switch the motor to this mode given user input, and lastly several light sensors will be used for automation of the scooter's lights. The feature of the self-charging comes from the regenerative brake. To be realistic, the regenerative braking is not to fully charge the battery, its primary function is to provide a longer life cycle for the battery and to provide some charge when an outlet is not available. The regenerative brake will convert some of the kinetic energy into useful energy to supply it back to the power source.

2.2 Goals

- Reduce the commute time.
- Reduce the amount of energy used by the rider.
- Reduce costs to consumers.
- Reduce carbon dioxide emissions.
- Maximize efficiency.

2.3 Objectives

- Build a low cost electric scooter.
- Incorporate a rechargeable battery that charges by regenerative brake.
- Include an LCD screen.
- Saving battery energy.
- To contribute to safety adding lights and sensors.

2.4 Motivation

Nowadays, millions of people depend on motor vehicles as their means of daily transportation. Unfortunately, owning a motor vehicle can be expensive in terms of purchasing the vehicle, routine maintenance, and repair costs. Additionally, the majority of motor vehicles burn fossil fuels and emit greenhouse gasses that contribute to the current climate trend of global warming. In recent years, many countries have begun shifting to being more ecologically minded in order to protect the habitability of the planet and to ensure future generations will not be burdened with our environmental mismanagement. One of the ways they have started to implement these shifts is through policy changes and agreements, like the Paris Agreement which is “a legally binding international treaty on climate change” (United Nations), to hold countries accountable[1]. The United States, in the past years, has encouraged society to be less dependent on fossil fuels in order to achieve this goal.

The usage of electric scooters in cities has increased in recent years and it will keep growing. Electric scooters have started to appear in big cities around the world as an evolution of the basic scooter which uses the driver’s foot. Electric scooters most commonly use lithium ion batteries but this system uses a lot of power so the battery must be constantly charged, this is one of the costs associated with electric scooters. Electric scooters can be charged in a few hours and can last more than 20Km avoiding traffic jams. Although scooters won't reach high speeds like a moto or a car, it is faster than a regular scooter. Furthermore, they are a great alternative to motor vehicles although in cities like Orlando there may be a great distance between point A to point B and a scooter may not be as efficient as a car. Using a rechargeable battery combined with regenerative braking will improve the performance and mileage of the existing electric scooters. E-scooters make the commute easier and more efficient. An electric scooter also promotes a cleaner environment and makes us less dependable on fossil fuel.

2.5 Requirements Specifications

Requirements
Motor capable of both power output and generation
Can automatically switch motor off when appropriate for safety concerns
Regenerative braking & coasting
Capable of varying motor output
Detects both light levels and scooter incline
Bluetooth capable
Automatic head and tail lights
Battery level monitor to prevent overcharging

Table. 1 Requirements

Specifications	
Unloaded weight	< 12 kg
Max loaded speed	> 15 mph
Battery capacity	4400mAh
Headlight brightness	100 lux
Lux reading precision	+/- 1 lux
Motor power	350 W

Table. 2 Specifications

- Motor capable of both power output and generation: this means the motor should be a direct drive hub motor and effectively precludes the use of a geared motor due to the added complexity that would entail.

- Can automatically switch motor between output, regenerative, and offline: the scooter will use several tilt sensors to determine when the scooter is on it’s side and switch between motor output and offline. Additionally, the regenerative mode may be enabled based on speed and other safety concerns.

- Regenerative braking & coasting: the means by which the motor acts as a power generator while exceeding a safe/legal speed will be applied to braking, regardless of the tilt of the scooter.

- Capable of varying motor output: a speed controller will be implemented to regulate the power being delivered to the motor.

- Detects both light levels and scooter incline: light and tilt sensors will be included to facilitate the automatic motor and light control

- Bluetooth capable: capable of Bluetooth pairing to personal devices for navigation purposes.

- Automatic head and tail lights: head and tail lights will be incorporated with the aforementioned sensors to allow for automatic light control

- Battery level monitor to prevent overcharging: charging capabilities will be disabled if battery levels reach an unsafe threshold

2.6 Marketing and Engineering Requirements

2.6.1 House of Quality Diagram

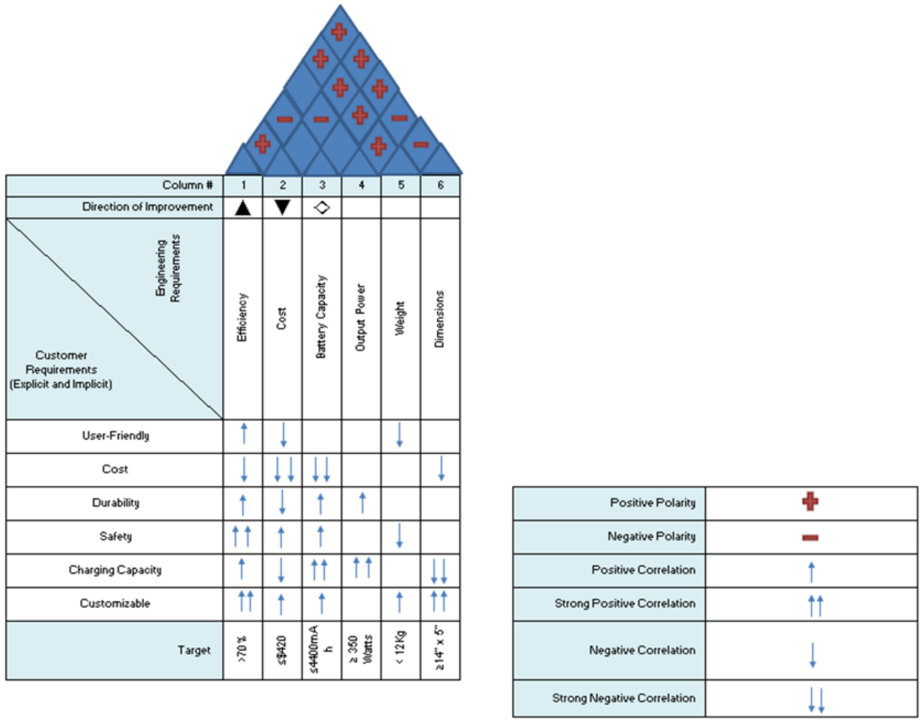


Figure 1: House of Quality Diagram

The engineering and customer requirements were selected to be able to perform at maximum performance. The engineering requirements that are listed are efficient for the overall scooter performance, cost of the EZ – Scooter as a whole, the battery capacity reflects the capacity of 2 iterations of our listed batteries connected in series, the output power of the scooter reflects the output of the hub drive motor and its ability to provide acceleration to the scooter, the weight of the scooter, and the varying dimensions for different terrain the scooter must endure. The customer requirements selected are for the best cases for the customer such as being user-friendly, cost effective, durable for long uses, safe to use for any customer, charging capacity that it is able to hold within the battery, and how customizable the scooter is for the customer. These requirements will allow us to achieve our objectives for the product.

2.6.2 Decision Matrix

When selecting our project idea, we had to evaluate different factors of each potential idea in order to pick the most suitable project. Some of the ideas we considered were e-scooter, a sand-cleaning robot, and a smart home assistance system.

A. Time: one of the main elements we had to consider was the time it would take us to develop our project. We looked at the different components for each idea and we concluded that the e-scooter could be done within the time frame given for our Senior Design 1 and 2 project.

B. Budget: since we are not looking for sponsorship, our project will be founded by the group members. Having this in mind, we had to look for ways to keep our costs low. After some research, we found that a lot of the parts required to build our e-scooter can be refurbished since a lot of them already exist in the market.

C. Accessibility: after considering the possible parts that our ideas would require, acquiring the different parts for the e-scooter turned out to be feasible compared to some of the parts we would need to develop our other ideas.

D. Motivation: one of our main goals for our idea was to work on a project that would help the environment. The sand-cleaning robot had good potential as it could be used to clean Florida beaches. The e-scooter is also an eco-friendly project as we could achieve a reduction of carbon dioxide emissions.

Below we will find a decision matrix we used in order to pick the most suitable project idea. We ranked each fact from low to high:

Project Idea	Time	Budget	Accessibility	Motivation
	-	-	+	+
Smart e-scooter	moderate	low	high	high
Sand-cleaning robot	high	moderate	moderate	high
Smart home assistance system	high	high	high	moderate

Table 3: Decision Matrix

3 Research related to Project Definition

3.1 Existing Projects and Products

The idea of an electric scooter is not a recent concept and has been around for more than a century. In today's technology, there are existing technologies that integrate self-charging and electric scooters, but with very different designs. We will expand on some designs that have been put together before.

3.1.1 Existing scooters

Hollyburn Rovers P5

The P5 model as shown in ***Figure 2*** has the motor totally enclosed and cooled with air and a 4400 watt power. The max speed of the P5 is 35 mph, it has no speed settings, the speed is controlled like the one from a moped, crank the throttle handle for a bigger speed. The maximum range is 28 miles in perfect conditions. AliCell Lithium-Ion batteries have 48.1 Volts 25.1 Ah, with capacity of 1.2 KWh, it charges in just 5 hours. good for long distance rides.

Figure 2: hollyburn P5

DIY Electric Scooter

This project concentrates on eco-friendly, the scooter counts with a brushless hub motor that was designed for a standard hoverboard, with a controller that best suits the motor, an electrical box which the user can charge the LiPo batteries. The box has a main power switch, voltage divider for the Arduino max analog is 5 Volts and controller runs off 12 Volts LiPo. Selected batteries were 6S and had 4 KmAh battery life, an LCD connected to the Arduino. To measure the voltage of the battery a voltage divider was added to the electrical box with a 47K Ω and 4.7K Ω in series. The LCD will display speed and

Figure 3: DIY Electric Scooter

odometer measurements. The resulting scooter is shown in ***Figure 3.***

Stator

Figure 4 is the Stator, the enormous wheel self-balancing electric scooter. The fully electric operates is relative silence The motor, batteries and are hidden away out of sight; the rear wheel is a Direct Drive 3 Phase Brush DC, providing a max power of 1000W giving a speed of up to 30 mph, a single piece of tubing forms the handlebars where the digital display control module is located. A lithium-ion battery of 54 Volts, giving a range of 50 miles at 30 mph. Is a havie scooter 40.8 Kg.

Figure 4: Stator E-Scooter

	Weight	**Speed**	**Price**
Hollyburn Rovers P5	80 lbs (127 kg)	35 m/h	$6,300
DIY E-Scooter	36 lbs (16 kg)	8 m/h	N/A
Stator	90 lbs (41 kg)	30 m/h	$3,995

Table 4 : Existing Product Comparison

3.1.2 Accessories

Many electric scooters have LCD finger throttles with P-setting (pre-programmed settings) that allows adjusting features like speedometer units (kph/mph), control, and zero-start. The two main types of LCD trigger throttles out in the market are *EYE throttle* and *QS-S4 throttle.* LCD throttles often include 10 to 15 P-settings, including some that should not be adjusted such as voltage shut-down, battery voltage, and protection; they can only be adjusted when understanding the functions.

The top 5 programmable P-settings in both types of LCD finger throttles are:
- Power level
- Cruise control
- Speedometer units
- Acceleration
- Start mode
- Brake strength

Many LCD throttles include USB ports. These USB ports are typically very limited in the amount of current they can output.

Figure 5: EYE LCD display

3.1.3 E-scooter Controller Design

The controller, also referred to as an electronic speed controller (ESC), is basically an electronic circuit that regulates the speed of the motor in an electric scooter. It regulates and directs power from the battery to the motor. Controllers are like the brain of the electric scooter; it links all other electronic components, electronic brakes, motor, battery, display, throttle, and speed sensor. The importance of electric scooter controllers is to take inputs from the electric components and determine the necessary signals that should be transmitted to them.

Some electric scooters controllers have an electronic braking function; they have regenerative braking function where they route power back to the battery from the motor when the regen brake is activated. Controller with regenerative brake to make sure it has this option in the controller we need to look at the specification sheet.

Controllers are categorized based on the motor type, functionality, and the voltage and current. Some of the variations are mentioned below: Brushed DC motor controller used in combination with brushed DC motor, simpler functionality primarily used in less powerful electric scooters. DC controllers have only one function: to regulate the amount of current supplied to the motor. Brushless DC Motor Controllers can only be paired with brushless DC motor (BLDC), BLDC motor controllers are reliable with enhanced efficiency, typically have three phases with at least two transistors per phase. BLDC controllers have a range of functions that are grouped considering if they do not have Hall sensor or they have Hall sensor, or if they have dual mode.

The BLDC shows how algorithms have been used to provide control of the BLDC motor. The electric scooter controller should have similar voltage and current as the battery and the motor of the scooter, this will ensure the prevention of overheating the controller. Also, to prevent malfunctions and short circuits by overheating the motor the phase current and the motor current must match. There are two different currents: the battery and phase current. The battery current is the current from the battery and the phase current is the final output to the motor. Another factor to consider are the sine and square wave controllers, it is not easy to identify because the optimal waveform depends on the scooter's wheel. They both have advantages and disadvantages. Some of this advantages and disadvantages will be mention below:

Square wave controller

Advantage
- Highly efficient in sudden acceleration and braking
- Compatible with different motors
- Not expensive
- Utilize more power voltage

Disadvantage
- Voltage sag when the scooter hits speed or acceleration
- The control is not smooth
- Produce a lot of noise
- Lower efficiency handling steep roads and heavy load

Sine wave controller

Advantage
- Have more climbing power and more efficient when having heavy load
- Produce lower noise
- More reliable and smoother control of all components

Disadvantage

- Consumes more power
- Compatible with only matched motors
- Can be more expensive

Another type of controller is the programmable controller. It is capable of different possibilities such as removing a speed limit and adjusting it at the one you like, adjusting acceleration and gradeability by increasing amps, and also increasing voltage to improve speed and enhance acceleration. In other words with the programmable controller we can reverse engineer the software to meet the requirements.

Electric scooter controller wiring connection: The black and red cable of the controller connects to the battery. The red wire connects to the positive terminal and the black cable connects to the negative terminal of the battery. The yellow, green and blue cables connect to the electric motor of the scooter, the five-pin connector (red,blue,green,yellow,black) the socket connects to the hall sensor, the three-pin connector (red, black and green) will connect to the throttle of the electric scooter. Red and white cables are the key switch cables, purple and black cables connect to the brake, green or yellow wire connects to the speedometer, black and orange are cruise control wires and the three-speed control wires: Orange, black and blue.

3.2 Relevant Technologies

3.2.1 Speed Controller

The speed controller (ESC) receives input from the throttle and precisely controls the flow of the current from the battery to the motor. A controller with higher-current, higher-voltage is capable of driving more powerful scooters. Controllers range from having sustained max outputs of tens of volts and little amps all the way up to 100 volts and 400 amps.

3.2.2 Scooter Motor

The motor power and quality will determine a lot about the scooter's performance, like its torque, its speed, the ability to climb hills, and it can contribute to its range. They are two types of motors used for electric scooters. The brushed hub motor and the brushless hub motor.

Brushed hub motor is an older type of motor where the mechanical brushes are moved along the motor to provide power to coil phases. This type of motor tends to wear over time due to friction and it can cause voltage irregularities. Although it is a low cost motor and it does not need a fixed speed controller it is less efficient and requires regular maintenance. Brushless hub motor is a newer type of motor, which is more efficient, more durable. The mechanical component is replaced by digital switching.

3.2.3 Throttle Control

The throttle for an electric scooter functions by turning on the motor to help generate the power, it controls the speed at which the motor drives the scooter, resulting in the speed at which the user travels. There are four different throttle styles: thumb, twist, trigger, and wheel throttle.

The thumb throttles are mounted on the backside of the handlebar. The user increases the speed of the scooter by using the thumb to press down on the thumb-size paddle. Twist throttle usually comes with an attached one-color digital dashboard, to accelerate, you grip and roll the entire handgrip towards you. Trigger throttle is the most common because of their capability to access and control performance settings, including torque/acceleration, and regenerative braking. And the wheel throttle is the most unusual throttle to see on an electric scooter, to be able to accelerate, roll the wheel inward and outward to engage the regen brakes.

3.2.4 Scooter Battery

The electric scooters run on electric batteries, most often Lithium-Ion batteries, their voltages start from 24 Volts, up to 120 Volts regularly. And on average, the battery has capacities between 150 Wh and 750 Wh. The average battery will take 5 hours to fully charge, and they will last between 2 and 3 years.

3.2.5 BT Module

A bluetooth module technology manages the communication channel of the wireless part. It can transmit and receive the data wirelessly by using two devices. Bluetooth is free to use in the wireless communication protocol, the range is less than other wireless communication , it operates at the frequency of the 2.41 GHz.

3.2.6 Microcontroller

A microcontroller (MCU), runs on a single chip, has a CPU, RAM and ROM memories, and interfaces inputs and outputs. Because microcontrollers are suitable for specific tasks it is better to choose the most appropriate microcontroller for our project. In order to achieve this some factors need to be considered such as: power efficiency, security, temperature tolerance, memory, software architecture, hardware interface, and cost.

3.2.7 Power Switch

A switch is an electronic device and is used to interrupt the flow of the electricity or electric current. So, switches are a part of the control system. A switch can perform two functions, one is by closing its contacts this means is fully ON, the second is by opening its contacts this means is fully OFF. Another important function of a switch is to divert the flow of electric current in a circuit.

There are numerous applications of switch, we found them in a wide variety of fields such as homes, industrial, automobiles, bikes, electric scooters and so on. They are two types of switches: Mechanical and Electronic. For our project we are going to focus on the mechanical switches.

3.3 Strategic Components and Part Selections

The component needed to do each function:

Function	Component
Supply energy	Battery
Movement	Motor
Braking	Motor (regenerative braking)
Speed moderation	Speed controller

Table. 5 Component Roles

3.3.1 Motor

The motor that we select must meet our power specifications, support the implementation of regenerative braking, and fit within our budget and weight constraints by doing the following:

- Motor has a power output rating of 350W.
- The acceptable voltage range for the motor should also be compatible with our selected battery pack.
- Motor capable of both power output and generation: this means the motor should be a direct drive hub motor and effectively precludes the use of a geared motor due to the added complexity that would entail.

- The motor will be one of our most expensive parts for this project but most still fit within our modest budget. Additionally, because the motor is one of our most expensive parts, a small percentage reduction in price can have a large impact on the overall project budget.
- Minimizing weight, as the overall weight of the scooter will be greatly impacted by the weight of the motor and, as with the price, a small percentage reduction between similar models will have a huge impact on weight.

We look up into two different motors that could meet all of our requirements.

BTER Wheel Hub Motor

This is a wheel hub motor with rubber solid tyre material, makes the motor and tire more stable and absorbs vibration. Drum-type brakes are easier to install than disc brakes. It has a Rated power of 350 W with a rated voltage of 24 V, weight of approximate 4140g. which falls within the specifications. With a maximum speed of 28Km/h, and with motor efficiency of 83%. The brushless and gearless 10-inch wheel hub motor is a good option for our project however with a cost of $162 is a little over our budget.

RBSD Electric Scooter Hub Motor

This motor is a wheel hub motor, with a power of 350 W, a rated power of 250 kW, and a rated voltage of 24 V. The wheel hub motor is suitable for front drive and rear drive which is a great option since it will give us the opportunity to decide where to install the wheel motor. This motor has an acceptable voltage range of 21-27V, a cost of $112.89, and a weight of 3.3 kg. Because this part meets all of our requirements it was selected for our project. This motor is the one we will select for our project since it meets all our requirements.

3.3.2 Microcontroller

The microcontroller that we select must be able to monitor several sensors simultaneously, supply/support input and output voltage levels that are compatible with out other components, and fit within our budget by doing the following:

- Must have multiple input channels available for monitoring several sensors simultaneous
- Can be used to send command signals to the implemented speed controller.
- Must have an operating voltage of between 3 and 5 volts.

The microcontroller that we selected to meet these requirements was the Arduino Uno REV3. Arduino Uno is based on the ATmega328P 8-bit microcontroller board. It has 14 digital input/output pins with internal pull-up resistors they are disabled by default but can be enabled with input_pullup command, 6 analog inputs pins, a 10-bit resolution on each pin, 16 MHz clock, 32 KB flash memory, 2 KB SRAM, 200 mA maximum for ATMega328 package, a USB connection (5V @ 500 mA) or using the positive 5.5mm/2.1mm Barrel Jack connection with the DC Barrel plug. This board is a perfect choice for our project since as mentioned before this microcontroller has an operating voltage of 5V, has an abundance of digital and analog input/output pins, a modest cost of $23.80, and a weight of 0.0250 kg. Because this part meets all of our requirements it was selected for our project.

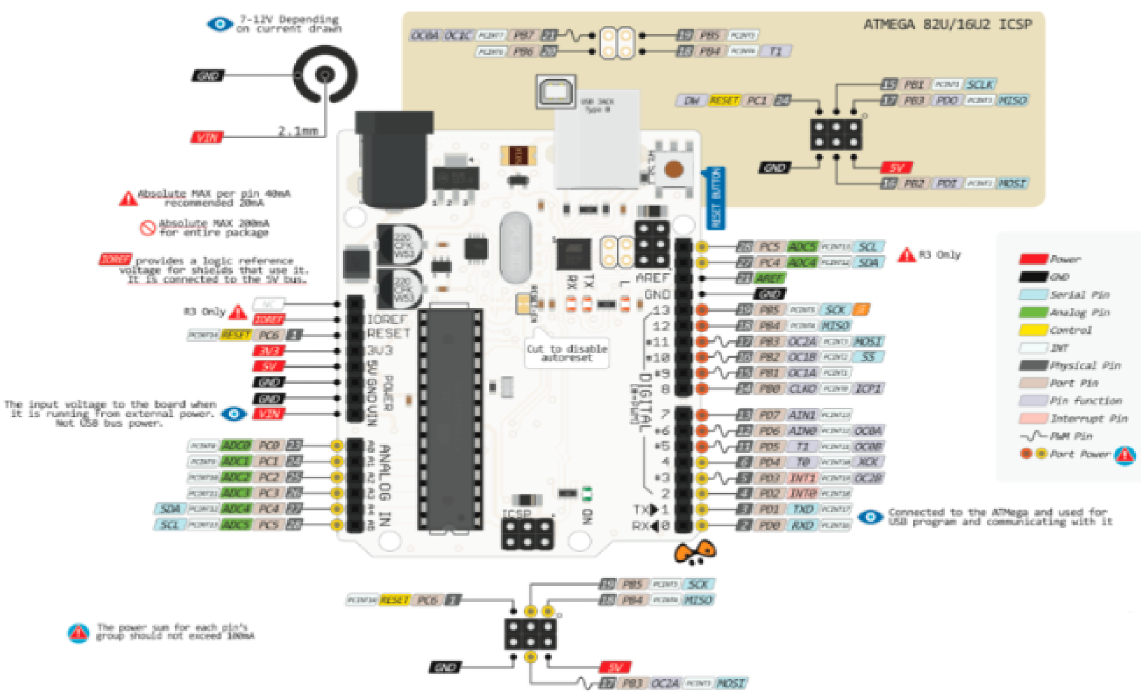


Figure 6: Arduino Uno Pins

As shown in Figure 8 the power pins are in red the IOREF, 3V3, 5V, and Vin. The IOREF provides the voltage reference which the microcontroller operates, the 3V3 is a 3.3-volt supply generated by the on-board regulator with maximum current draw of 50 mA, 5V pin outputs a regulated 5 volt from the regulator on the board the power can be supplied with the DC power jack, USB connector or the VIN pin of the board, Vin input voltage to the board.

3.3.3 Tilt Sensor

We had no shortage of options when selecting our tilt sensor and as such had several limiting criteria that narrowed our search. The tilt sensor that we select must be able to be used to determine when the scooter is on its side or at an incline and send feedback to the implemented microcontroller and must fit within our budget by doing the following:

- Must have an adjustable tilt trip.
- Must have an acceptable operating voltage range.

SW-520D Tilt Sensor Module

One of the available tilt sensors that we considered was the uxcell Angle Sensor Module SW-520D Golden Ball Switch Tilt Sensor Module. This tilt sensor is meant for use with our selected microcontroller, so capability will not be a concern. Additionally, this tilt sensor's trip incline is adjustable and is very affordable, with 5 sensors costing only $9.99. Because of this, this part meets all of our requirements and was being seriously considered for use in our project.

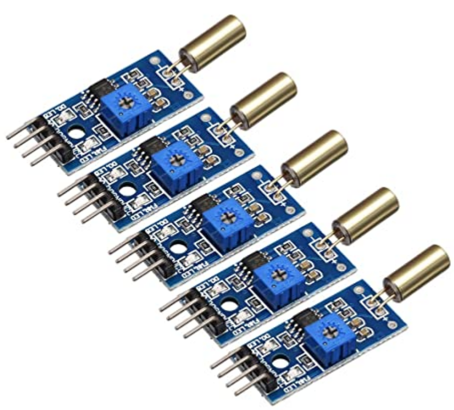


Figure 7: Uxcell Angle Sensor Module

GY-521 MPU-6050

The second sensor that we seriously considered was the Dorhea GY-521 MPU-6050 Module. This sensor was not variable in the same sense as the first sensor but rather provides a range of readings to indicate the incline. To contrast with the first sensor, which only provided a signal to represent if the sensor was inclined above a certain threshold. This greater depth of information as to the state of the scooter incline means that the implementation can be more versatile. Additionally this module includes an accelerometer, which can be used for many different features, from safe features to assisted driving.

3.3.4 Light Sensor

The light sensor is a passive device that converts the light energy into an electrical signal output. In our project we are going to use the light sensor to detect the presence or absence of light. The light sensor helps to play a vital role in the safety of our scooter. The light sensor that we select must be able to be used to determine when outdoor light levels are low and feedback to the implemented microcontroller and must fit within our budget by doing the following:

- Must have an appropriate light reading range sustainable for distinguishing between typical driving conditions (night, dusk, daylight).
- Must have an acceptable operating voltage range.

We look into three different Lux sensors the BH1750, TSL2561 and the TSL2591. The BH1750 it is a simple photodiode an ADC and an I^2C interface the spectral response as expected covers the visible wavelength the range is from 1to 65535 lx this means this sensor should work from daylight until the dusk because looking at the table at 100,000 is the direct sunlight and at 0.0001 is moonless, overcast night sky having a cost of $6.89. Then we take a look into the TSL2561it has no IR filter but it has a second channel for the IR illumination it precisely measures illuminance in diverse lighting conditions the dynamic range is 0.1 to 40,000 Lux and its voltage range is 2.7-3.6V, it also contains two integrating analog to digital converters that integrate current from two photodiodes, simultaneously, the board size is 1.7x1.3cm (approx.), with a cost of $20.00. The TSL2591is an advance digital light sensor, the built-in ADC means you can use this with any microcontroller, even if it does not analog inputs, current draw is extremely low this is good for low power data-logging systems, with a cost of $8.94.

After comparing the different light sensors, the light sensor that we selected to meet these requirements was the DAOKI 5PCS GY-302 Light Intensity Sensor Module. This module has the main chip the BH1750, it also contains a 16 bit analog to digital to convert the analog illuminance values to give a digital output in lux is the measurement of illuminance, the 662k voltage regulator, a power supply of 3-5 volts, a light intensity range of 0-65535 lx. This light sensor is meant for use with our selected microcontroller, so capability will not be a concern. Additionally, this light sensor is meant to model visual sensitivity and should be well suited for our intended use. This sensor also contains communication level conversion and is very affordable with 5v microcontroller Io direct connection. Costing only $6.89. Because this part meets all of our requirements it was selected for our project.

The way this sensor is going to be connected is to be able to interact with the Arduino Uno to measure the light intensity or the luminance of light in a given area. First, we choose our outer pins from the Arduino to be A5 for the clock and A4 for the data. So looking into the light sensor GY 302 the VCC is connected to the Arduino 5 volts pin, the next one is the GND or ground it will be connected to the GND from the Arduino, the SCL or the clock pin will go to A5 and the SDA or data pin will go to A4 and lastly the address pin to A3.

3.3.5 Speed controller

The speed controller that we select must meet our power specifications, support the implementation of regenerative braking and automated motor control, and fit within our budget by doing the following:

- Capable of automatically switching the hub motor between output, regenerative, and offline when used in conjunction with a microcontroller.
- Capable of being used in conjunction with a direct drive motor to implement a regenerative braking mode, which can be activated based on microcontroller control signals. This means the speed controller must be programmable.
- Capable of regulating the power being delivered to the motor to facilitate a range of speeds.
- Minimizing costs, while supporting both regenerative braking and integration with our microcontroller.

The speed controller that we selected to meet these requirements was the HGLRC Flipsky Speed Controller, part number: FSESC V4.12

Figure 8: HGLRC Flipsky Speed Controller

This speed controller has an operating range of 8-60V and 50A continuous, a cost of $122.99, and a weight of 0.08 kg. Additionally, this speed controller supports regenerative braking, is programmable, and can be used alongside our microcontroller. Because this part meets all of our requirements it was selected for our project.

3.3.6 LCD

When choosing an LCD we had no shortage of options. To narrow down our selections our chief concerns were easy implementation with our microcontroller and affordability by doing the following:

- Not being an over designed LCD for our intended use (keeping costs low).
- Must have an acceptable operating voltage range.
- Low current draw from the LCD to conserve as much of the battery as possible
- Cost needs to be efficient; the LCD for our project does not need to be extravagant.

The LCD that we selected to meet these requirements was the HiLetgo HD44780 1602 LCD. The 1602 is a dot matrix module to show numbers, letters, and characters. It is composed of 5 x 11 dot matrix positions and each one can display one character, the 1602 means 16 characters columns 2 lines rows. We will be using the LCD to display the speed and the battery life. The LCD has 16 pins: the first two are ground VSS and VDD pins to power the module, the VO to adjust the contrast when zero volts on that pin gives max contrast while 5 volts gives minimum contrast, next is register select pin followed by read write pin, then we have the enable pin, after we have the eight data pins, and finally we have anode and cathode of the backlight led. With all this information we can connect our LCD with the Arduino we start with ground and VCC going to Arduino ground and 5 volt pins respectively, the register select pin goes to Arduino digital pin D2, ground the read write pin this is to enable writing to the module, then connect enable pin to digital pin D3, D4 through D7 data pins go to Arduino D6 through D9, anode goes to 5 volt pin, and cathode to ground. This LCD is meant for use with our selected microcontroller, so capability will not be a concern. This LCD is very affordable with two costing only $9.19. Because this part meets all of our requirements it was selected for our project.

3.3.7 Rechargeable Battery Cells

When selecting a battery the most important characteristics were low weight, high energy density, cost effective, and easy capability with the rest of design. The battery we selected needs to do the following: Have enough energy storage to be a viable power source for our project and have an acceptable operating voltage range. As mentioned previously, there were some parameters we take into account when choosing the battery. These include:

Performance
Performance is one of the important parts for our scooter since performance in this context means how much power it is capable of generating and how much run time it will offer at each charge, some batteries have the same performance but it is possible that one perform better than the other the only way to know it is by testing each battery.

Durability
Another important factor is durability is important to consider how long the battery will last, each battery system has unique system in terms of charging, discharging, adverse temperatures we need to take into account when using in rough areas it can affect the durability of the battery because we are putting cells through large power draws more number of cycles.

Voltage
Batteries with higher voltage allow the electric scooter to produce more power but by being with higher voltage they can be heavier because they contain more cells. We also

need to match the battery voltage with the one in our motor. It is important to make sure our motor is able to support the voltage of the battery, since we are limited in our speed we do not need a very high voltage battery.

Taking into account all the parameters mentioned before, the battery that we selected to meet these requirements was the Ovonic 3S Lipo Battery 25C 2200mAh 11.1V Lipo Battery. This battery when connected in series with a second duplicate battery will be able to supply power to our motor within the correct voltage range. This battery fits within our cost constraints with two costing just $30.99. Because this part meets all of our requirements it was selected for our project.

3.3.8 Throttle Control

When selecting our throttle control the characteristics we were looking for were ease of use, a distinct implementation to set our scooter apart, dual use for regular motor operation and regenerative operation, and affordability. To accomplish this our throttle needs to do the follow:

- Requires minimum physical effort to use.
- Has a unique form from other throttles typically used.
- Multidirectional functionality.

The throttle controller that we selected to meet these requirements was a joystick, the HiLetgo Game Joystick Sensor, part number: 3-01-0492-1. The design of the joystick with x, y axis analog output, and z digital output leads us to be able to use a 3-pin special line really plugged into the expansion board. With an input voltage range of 3.3 volts to 5 volts DC, a weight of 0.03 pounds, and with a high quality. It will be connected to the Arduino by starting with the switch which is a digital pin it will be connected to D2, then the y axis into A1, the x axis on A0, the 5 volts into the 5 volts and ground to ground.

This joystick will minimize physical strain on riders during long rides. Additionally, this joystick's multidirectional range can be used as a manual means to engage the regenerative braking mode. This part is also very affordable at only $5.79, while a typically employed type throttle can cost over $30. Because this part meets all of our requirements it was selected for our project.

3.3.9 BT Module

When selecting our BT module the characteristics we were looking for were easy implementation with our microcontroller and affordability by doing the following:

- Selecting a simple, bare bones module.
- Selecting a module designed with our microcontroller in mind.

The Bluetooth module is what we need to create a wirelessly communication with our electric scooter. Since we are using Arduino for our project to send and receive data over Bluetooth, we explore some modules compatible with this one and cost effective. The BLE Link Bee and the HC-05.

BLE Link Bee

This Bluetooth module has a transmission range of up to 60 meters in free space, it is a serial to Bluetooth 4.0, it also integrates a voltage regulator that supports both 5 volts and 3.3 volts microcontrollers, wireless programming for Arduino processor is compatible with Fio, Mega, and Arduino UNO, some other specifications are the frequency is 2.4GHz, the power consumption when working is 10.6mA and the ready mode is 8.7mA, the size is 32mm * 22mm, with a cost of $10.99.

HC-05

This other Bluetooth module is designed for transparent wireless serial connection setup with Bluetooth SPP (Serial Port Protocol), it covers 9 meters of signals and works both as a master or as a slave which means it able to use neither receiving nor transmitting data. Some of the specifications for the HC-05 is a 5 volt DC, it has a 5 Pin VCC, GND, TXD and RXD which connects with the microcontroller, and the KEY which if the input is a low level the module is at paired mode and if it is input high level, it will enter to AT mode. It is very easy to use and connect.

The BT module that we selected to meet these requirements was the HiLetgo HC-05 Wireless Bluetooth RF Transceiver Master Slave Integrated Bluetooth Module. This BT is meant for use with our selected microcontroller and as such, integration with the rest of our design should be simple. This BT module fits within our budget constraints and costs just $8.59. Because this part meets all of our requirements it was selected for our project.

3.3.10 Ignition Switch

The power switch controls the flow of power to an electric device. It has an on and off position usually represented by 1 that is the on position and by o that indicates the off position. In the electric scooter we can use a power switch or a key switch. Which of the power switches is chosen will be determined by availability, cost, and the rating to suit the requirement and the ease with which it can be used. We look into some of the power switches that we are considering to integrate into our electric scooter.

Before looking at the different types of mechanical switches we are going to look for some of the characteristics. The two important characteristics are its Poles and Throws. A throw represents a contact-to-contact connection and a pole represents a contact. Another characteristic is its action, they have a Latched action or Momentary. The Momentary switches like push buttons are used for a brief time or as long they are pushed but Latched switches maintain the contact until it changes to the other position.

Throw Switch KDC2

The throw switch has different poles and throws. The single pole and the single throw is the basic ON and OFF switch; it consists of one output and one input contact, it switches a single circuit and it can be normally open or normally closed. The single pole and double throw switch consists of three terminals: one is the input and the other two remaining are the output contacts, so it has two ON positions and one OFF, these types of switches are normally used as changeover; it connects the input between two choices of outputs. The throw switch we look into is the *KCD2 is a ON/OFF power switch with black lever, it has the universal power symbols as; I = On and O = off, for push-on connectors on back side switch, is a 8 x 2 x 1.5 inches rectangular form.*

Push Button Switch Handlebar E-scooter

It is generally a momentary contact switch that makes connection as long as it is pressure, this pressure is provided by someone's finger. The working voltage is 250 VAC 3A, two wire connector, single throw SPST contact type.

Joystick Switch

It consists of a lever that moves in more than one axis of motion, the joystick are manually actuated control devices, depending on the movement when the joystick is pushed, one or more switch contacts are activated. As mentioned before, the throttle that we are using for our electric scooter is the HiLetgo Game Joystick Sensor, and is the one we are going to use for the power switch in case we go for this option.

SYWAN Ignition Switch

The SYWAN is a 2-wire ignition switch key, this cylinder assembly key includes two butt splice connectors, is a two position on-off, you can remove the key only in off position, the wires length is 9 inches and has 2.8mm male terminals, 3A power level. This type of key has a reliable ignition switch locking mechanism and it is made of metal alloy material.

3.4 Possible Designs and Related Diagrams

3.4.1 Top Level System Diagram

Top Level System EZ Scooter

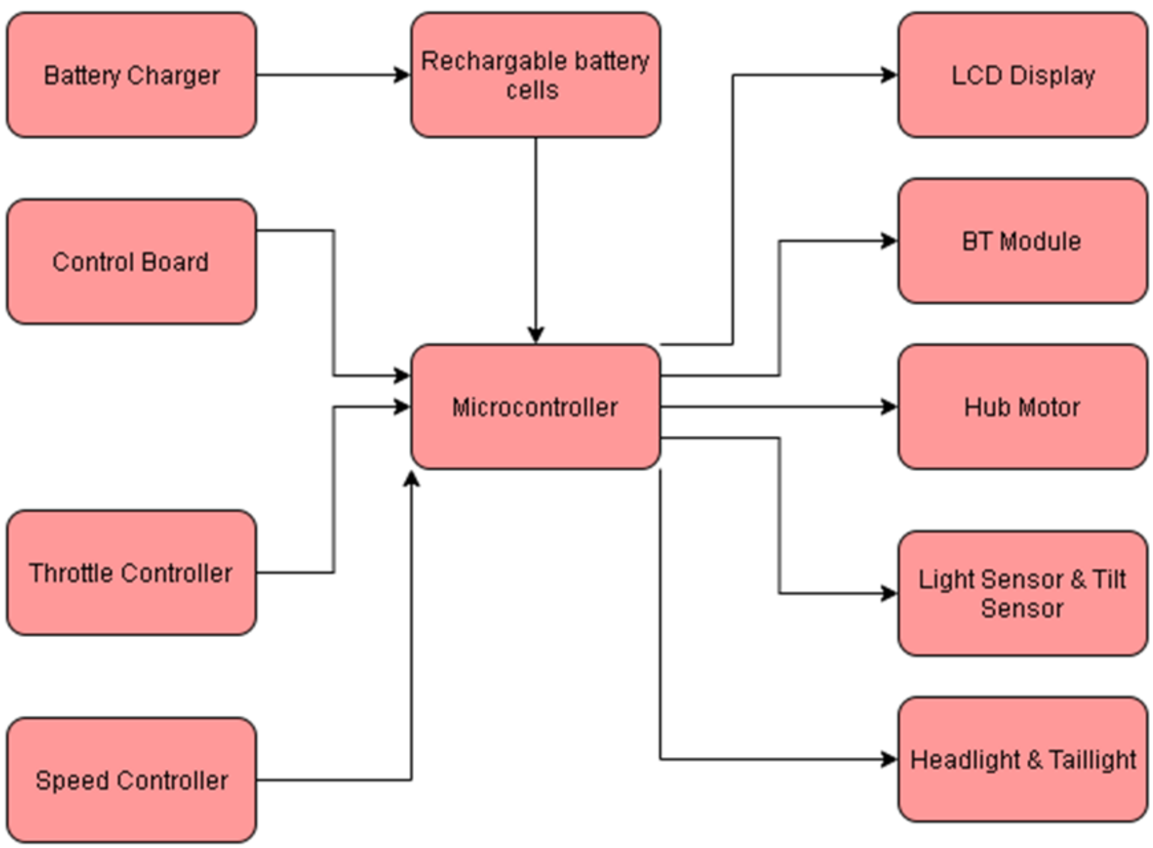


Figure 9: Top Level System

When looking at our design, for the EZ Scooter we can see that two different power systems are displayed where one power system is focused on the rechargeable battery cells powering our hub motor and the other power system focusing on the accessories such as our LCD Display, BT module, headlight/taillight, light and tilt sensor. The throttle controller that we chose was the HiLetgo Game Joystick Sensor that contains two analog that converts to digital output, high-quality rocker, long life, consistent performance, and very easy to use for the general user. The throttle will work in tangent with the speed controller to be able to determine the speed in which the scooter should be at and tilt sensors will be able to help for measuring the tilt of the scooter and multiple axes. For example, if the scooter is going fast due to a straight road and a turn is coming up the tilt sensor will tell the throttle that the scooter needs to slow down due to the turn. The microcontroller is the center piece for all of this in which it focuses on the functionality of the vehicle as a whole with both the accessories and the motor itself. Without the brain of the microcontroller, it would not work as a whole. The memory of the microcontroller will store all the programs and data that will help storage for the sensors, motor, BT module, and even LCD Display. We are looking to have the most secure, tolerant, cost efficient microcontroller that will help make our scooter stand out.

3.4.2 Speed Controller

Speed Controller Flowchart

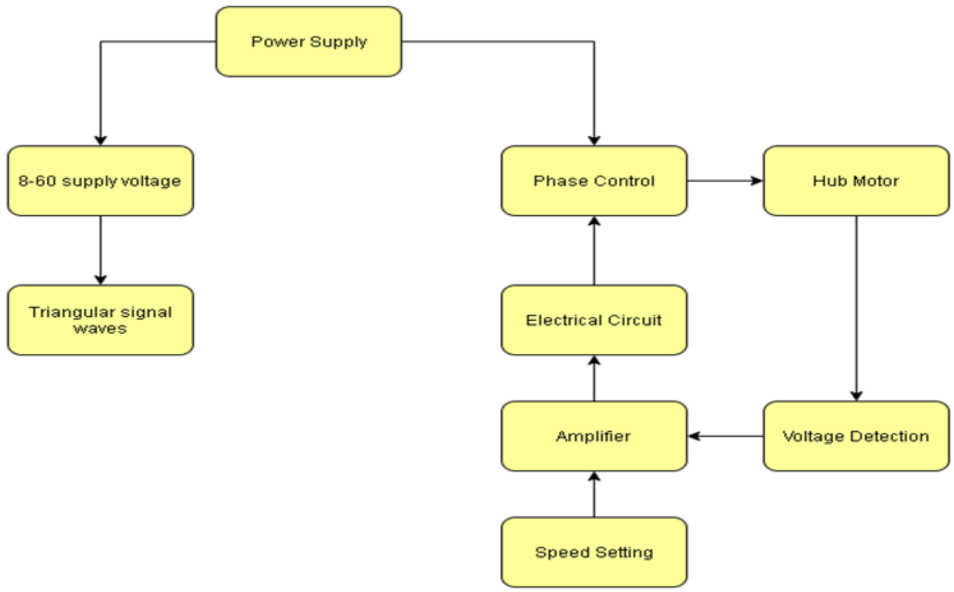


Figure 10: Speed Controller

The speed controller that we decided to use is the HGLRC Flipsky Speed Controller due to the fact that it fits within our budget and operating range of 8-60V, 50A continuous, and a light weight of 0.08 kg. The speed controller will focus on regulating the power that is being delivered to the motor and the regenerative properties when braking to be able to get more power received. The regenerative properties are referring to the power we will be regaining when slowing down and is delivered to the batteries for the scooter to be able to have a longer lasting battery life. In terms of performance, the EZ scooter will be able to perform much higher than average with our range operating range using triangular waves that will send signals to the motor and receive input from the throttle. In Figure 6 we can see the process of how the speed controller works from starting with the power supply, rechargeable battery cells, to gain voltage and send triangular waves to then sending a signal to the control unit where the speed controller sends a signal to the amplifier and from the amplifier to the electrical circuit back to the control unit where the hub motor is signaled by the control unit to determine its correct speed.

3.4.3 Throttle Controller

Throttle Controller Flowchart

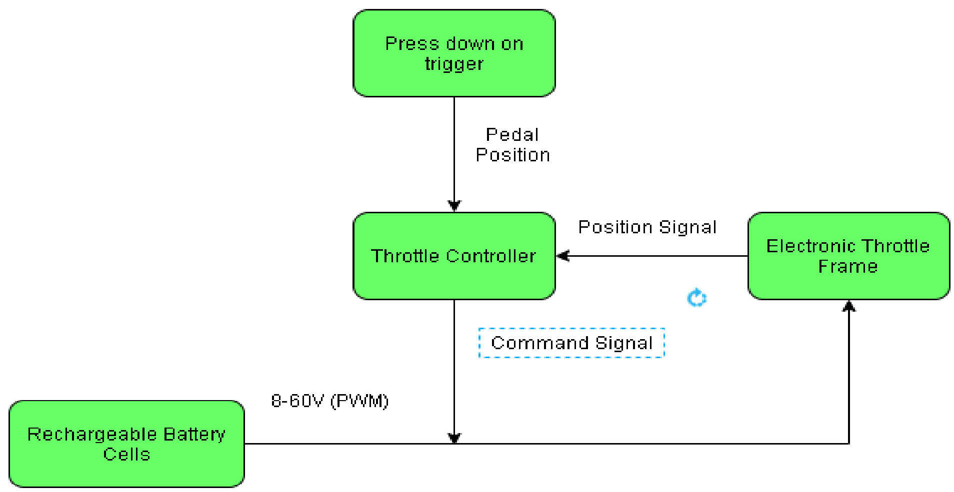


Figure 11: Throttle Controller

The throttle controller contains characteristics that will allow us to turn on the motor and generate power by being able to control the speed with the use of the thumb throttle on the EZ scooter. The thumb throttles will be placed on the backside of the handlebar which is a common placeholder for these types of scooters. Once the thumb throttle is pressed down the pedal position will be changed and will signal the throttle controller. Once the throttle controller has been signaled, the throttle controller will send a command signal to the bridge driver which is the Pulse Width Modulator (PWM) in which it will send a voltage range of 8-60 V to the electronic throttle frame to be able to accelerate. The driver is also powered by our rechargeable battery cells. The electronic throttle frame will finally loop back to the throttle controller and send a position signal to notify that the thumb throttle is being pressed down or let go of. With the operating range of 8-60 V our EZ scooter will be able to go pretty fast and save a lot of energy when braking with our regenerative brakes. The throttle is straightforward and easy to use for its multidirectional functionality that makes it stand out from other throttles.

3.4.4 Arduino Uno Rev3

Arduino Uno Rev3 Block Diagram

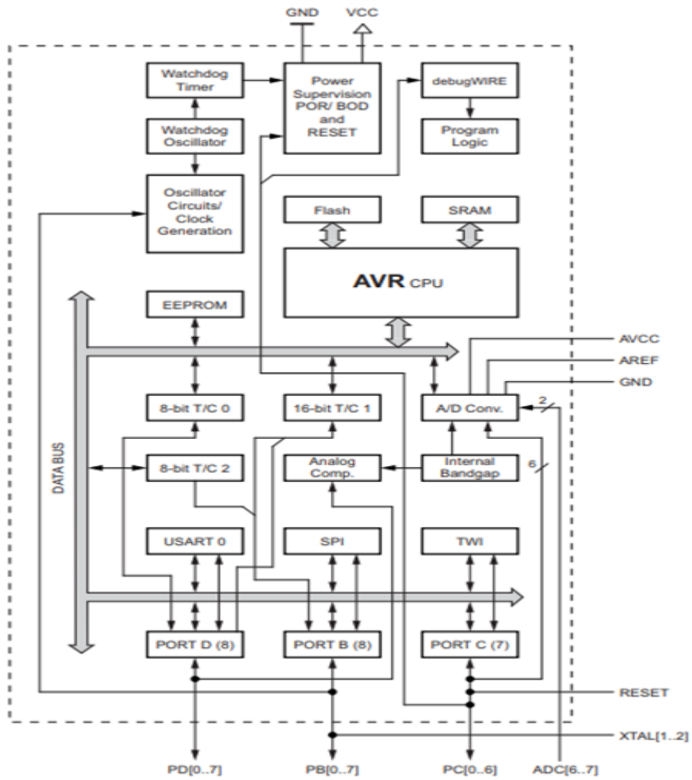


Figure 12: Microcontroller

The microcontroller is essentially the brain of the whole scooter in which it controls the functions of the embedded systems in the EZ scooter. The components of a microcontroller contain memory, a processor, and peripherals. There are many different types of microcontrollers that can be used for different purposes, but the microcontroller board that we are going with is the Arduino Uno Rev3, which is one of the best microcontrollers for tinkering with the platform and the most used board of the Arduino family. The main tasks for the microcontroller are to measure the time the wheel takes to rotate in one whole rotation which will then make the speed that is measured into miles per hour (MPH). The Arduino Uno Rev3 is really straight forward in the sense that it is very user-friendly, programming is a lot easier for the general user, and when creating a UI for the LCD display it will be easy to understand. The budget fits perfect as this presents itself as one of the most universal boards in the case we are using it. The general characteristics of the microcontroller are the fact that it has 5V for a operating voltage, 16MHz for the clock speed, a low weight of 25 grams, and a flash memory of 32KB.

3.4.5 RBSD Electric Scooter Hub Motor

RBSD Electric Scooter Hub Motor

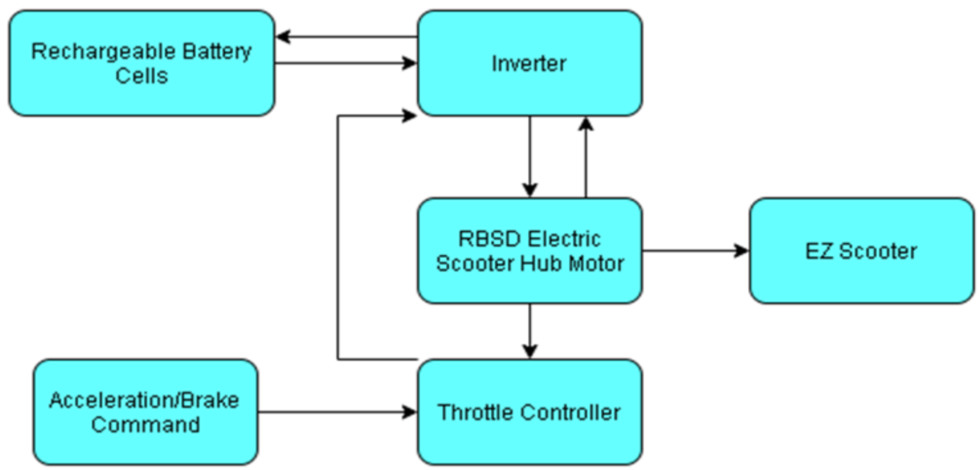


Figure 13: Hub Motor

The motor that we chose for our EZ scooter is the RBSD Electric Scooter Hub Motor which is really versatile and suits our project best due to the features and specifications that go along with it. The motor allows for front drive and rear drive, contains 350 W of power, a strong and long-lasting tire, very useful for the general user and is very easy to install when building the scooter. The motor also contains an operating voltage of 24V, which means less batteries used and therefore equates to less weight on the EZ scooter. The biggest factor when looking for a motor is the weight due to the motor being one of the heaviest if not the heaviest part of the scooter and our selected motor came to be only 3.69 Kg which is equivalent to 8.14 lbs. For the hub motor, the battery gives power to the inverter and vice versa then the inverter transports its power to the hub motor and is controlled by the throttle controller to check if you are braking or accelerating.

3.4.6 Grove - Light Sensor v1.2

Grove - Light Sensor v1.2

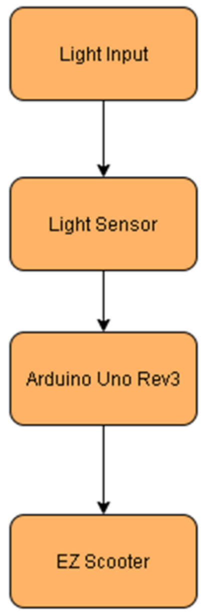


Figure 14: Light Sensor

The light sensor we chose is the Grove - Light Sensor v1.2 which is an analog module that can pair with our Arduino board, the Arduino Uno Rev3. We chose this light sensor because of its easy use of a simple plug-and-play, high reliability and sensibility, the response time is very quick ranging from 20-30 milliseconds and the range on recognizing the light has a wide radius. The light sensor is also very lightweight, only being 8 grams with an operating voltage of 3-5 V and operating current of 0.5 – 3mA. The flowchart above *Figure 9* shows the process of how the EZ scooter receives the light energy converting to an electrical signal where the light is radiating on the sensor, which signals the Arduino Uno Rev3 Board that light is being taken in and converts to electrical energy for the EZ scooter.

3.4.7 uxcell Angle Sensor Module

uxcell Angle Sensor Module

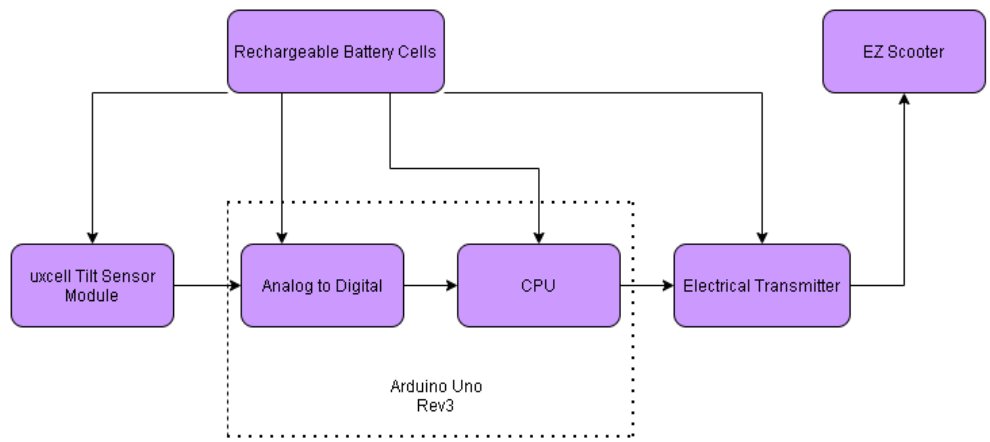


Figure 15: Tilt Sensor

The tilt sensor that we chose is the uxcell angle sensor module that focuses on changes in the body angle at which the scooter is facing. The sensor has a working voltage of 3V to 5V that is also a digital switch output (0 and 1). This tilt sensor uses a LM393 comparator which outputs a clean signal, a stable waveform, and a current of more than 15 mA. The weight of the sensor is also very small and easy to understand as long as you have a basic understanding of electronics. In Figure 10 above we can see how the tilt sensor works in action where the rechargeable battery cells power each component within the Arduino Uno Rev3 and the tilt sensor module. The tilt sensor module then sends a message to the Arduino that the scooter is tilted and for it to convert from analog to digital. The microcontroller then receives the message and transfers over to the electrical transmitter to be able to finish the message and tell the EZ scooter that there is a shift in angles.

All the listed diagrams above show the major parts and their processes. Each part is intertwined in some way shape or form and without one of the parts then it would not work and would not be up to our standards. The role of each part is important in their own ways and below is a list recapping the importance of each major part involved in our application.

- Microcontroller – the brain of the whole operation, where it controls the functions of the embedded systems of the EZ Scooter
- LCD Display – Gives us a screen in which a word or digit can be displayed on a seven-segment display.
- Speed Controller- Controls the speed of the hub motor and receives input from the throttle.
- Throttle Controller- catches the signal between the thumb throttle and the hub motor.
- Rechargeable battery cells- The power supply for the whole scooter.
- Hub Motor – converts the electrical energy that is received from the batteries to mechanical energy and allows for the scooter to move.
- Headlight/Tail Light – These lights are placed in the front and back of the scooter for oncoming traffic to know that you are driving and for the user to see better in the dark.
- Battery Charger – charges the rechargeable battery cells
- BT Module – focuses on the communications of the scooter wirelessly for easier access to different options.
- Light Sensor – detects light and creates the light energy to convert to electrical energy as a signal.
- Tilt Sensor – Measures the tilt of the scooter in multiple different axes in reference to a leveled out plane.

3.5 Parts Selection Overview

Description	Weight (kg)	Voltage (V)	Amperage (A)	Power (W)	Storage	Cost per
Scooter	4.0823					$69.99
LCD	0.2500					$9.19
Microcontroller	0.0250					$23.80
Speed Controller	0.0800	8V-60V	50			$122.99
Rechargable battery cells	0.3360	11.1V x 2			2200 mAH x2	$29.99
Hub Motor	3.3000	24 (+/- 3 V)		350		$112.89
Headlight	0.2000					$18.99
Taillight	0.1200	36				$16.99
Battery charger						$12.97
BT Module	0.0035					$8.59
Joystick module	0.0136					$5.79
light sensor (x5)		3-5				$6.89
gyro-accel (x5)		3-5				$10.99
Other Components						$250.00

Table 6 Parts Selection Overview

The table above gives an overview of all major parts selected for our electric scooter. This selection of parts will enable a design to meet the requirements and reliability needs, and also it will help to control parts to ensure that the design of the electric scooter will not be compromised.

4 Related Standards and Realistic Design Constraints

4.1 Related Standards

Related Standards	
CPSC	**Consumer Product Safety Commission**
DMV	**Department of Motor Vehicles**
FDOT	**Florida Department of Transportation**
SIG	**Bluetooth Special Interest Group**
IEC	**International Electrotechnical Commission**

Table. 7 Related Standards

4.1.1 CPSC

Is a governmental agency which is responsible for the safety of consumer products, which includes e-scooters. They promote safety by developing uniform safety standards and examining possible product-related safety concerns.

The CPSC *Consumer Product Safety Commission* “protects the public from unreasonable risks of serious injury or death from thousands of types of consumer products under its jurisdiction.” It was created by the U.S. Congress in the early 1970s, the agency often assisted by the yardstick of ASTM International standards. For our project to focus on safety we should comply with the requirements in ***16 CFR Part 1512*** for braking, structural integrity, and reflectors. Another consideration to standards which is related to the design of the EZ-scooter is ***UL.*** CPSC has an accredited laboratory UL assisting with the testing of a certification by ***ASTM F963-11. UL 2849*** standard for electrical systems for electric scooters like cell charger, and battery it tests for overcharging, temperature, inputs, and short circuit among others.

UL 1642 will test for the strength and rigidity of a battery it has to resist the abuses that it may be subjected to without resulting in a risk of fire. Therefore, the steps designed in this standard allow for the safe in the battery. In summary, the step is to understand how often and how long the battery will be in use.

The design impact of CPSC is that we must ensure that the components that we implement within our product meet minimum safety guidelines for metrics such as the impact rating of the battery. Additionally, we must implement safety measures within our design to prevent overcharging and monitor operating temperatures.

4.1.2 DMV

Is a governmental agency which is responsible for the registration of vehicles and licensing of drivers. These Government agencies may be organized at different levels and with different scopes. For the United States for example DMV is organized on a state level, and thus all regulations for each state that a vehicle is intended to be used in must be considered. These DMVs provide the guidelines as to what is considered an e-scooter and what performance thresholds and characteristics that necessitate a license be required to operate a given scooter.

The design impact of DMV regulations is that in order for our e-scooter to be considered a motorized vehicle, subject to the same laws as a bicycle it needs to In the requirements as per Florida Statutes section 316.003(48). We must ensure that our procured scooter must have “not more than three wheels”. Additionally, this means that the speed of our scooter should be limited to ensure that it is “not capable of propelling the vehicle at a speed greater than 20 miles per hour on level ground”.

4.1.3 FDOT

Is an agency which is responsible for the public transportation of florida. This agency’s responsibilities include the maintenance of roads, bridges, and several park and ride lots throughout Florida. This agency also is responsible for the regulations surrounding transportation in the form of traffic laws.

The design impact of FDOT regulations is that, assuming a vehicle speed below 20 mph as mentioned above, bicycle regulations must be followed while testing and demoing our project as laid out in Florida statute 316.2065. However, it is worth noting that the provisions detailed in this Florida statute that “by their nature have no application” will not impose legal restrictions upon our design. One such example of a provision which has no application is the requirements that a rider be “astride a permanent and regular seat” and such laws that by their nature would not apply to a scooter. Some of these design restrictions imposed on our project by following this Florida statute will depend on the circumstances of our testing environment. After we have a more solidified idea of our planned testing execution we will once again revisit this Florida statute to ensure that we abide by all transportation laws that are applicable to our testing environment.

4.1.4 Bluetooth SIG

Bluetooth SIG, short for Bluetooth special interest group, is a standards organization. To put it shortly these organizations fill the role of providing uniformity among consumers, manufacturers, government, and other applicable parties. This provided structure and uniformity could be related to terminology, specifications of products, and procedures. This means that Bluetooth SIG is an organization whose primary objective is producing technical standards and maintaining them through revision, amending, interpreting, and etcetera. More specifically it is a non-profit organization that is responsible for the development of bluetooth standards and licensing of bluetooth technologies.

The design impact of the SIG bluetooth standard is that our project's bluetooth implementation must follow the SIG standard and specifications as listed in the SIG core specifications 5.2 documentation. One such specification that we will be following is ensuring that ourBluetooth capable devices support a Bluetooth frequency of approximately 2.4 GHz. Additionally our design will need to support the typical range use case of Bluetooth which is approximately 10 meters.

4.1.5 IEC

International Electrotechnical Commission or IEC Is an international organization that develops and publishes standards in a wide range of technology fields. All of the international standards that they prepare and publish fall within the broad categories of electronic, electrical, and related technologies.One such related category of standard that relates to our project because of the published standards in the battery field. The second category of related standards is semiconductors, which relate to our project because of our implemented MOSFETs.

The design impact of the IEC standard is that during the battery selection, handling, and procurement we need to be mindful of the IEC classification of any batteries so that it 1) meets design specifications 2) we exercise proper handling proceeds. Lastly, IEC semiconductor standards impact our design by giving us guidelines for testing of reliability and accuracy. Additionally, we should be mindful of ensuring the semiconductors that we procure meet IEC standards for reliability, sustainability, performance, and fault resistance. That way we can have greater confidence in the reliability and performance of the semiconductors that we Implement in our design and the performance of our project as a whole.

4.2 Realistic Design Constraints

Constraints		
Man-hours	Motor power	Dimensions
Costs	Battery capacity	Safety
Weight	Power generation	Legal

Table. 8 Constraints

4.2.1 Man-hours

Our team is constrained by our available man-hours and the labor that man-hours represent. Firstly, we are limited in the number of man-hours that we are able to put into this project. This limitation is due to other responsibilities that we must divide our attention among (i.e. family, work, other classes, etc). Secondly, the labor that a man-hour represents is another dimension of this constraint. This is because the circumstances surrounding our work impacts it's quality and quantity. For example, spending 30 man-hours over the course of one week towards our project would likely result in a greater quantity and quality of work when compared to working 30 consecutive hours. Thus we must balance this constraint in order to ensure that we are consistently making progress on our project while also maintaining our mental and physical wellbeing.

4.2.2 Cost

Because this project is being funded at the expense of it's group members, cost is a very important constraint to keep in mind. We must make sure that our project incorporates parts that have the required functionality while also minimizing cost. This is a constraint that is connected to the man-hour restraint because oftentimes cost and labor required to implement are inversely related. Discounting the cost of parts already owned by team members we will attempt to keep the project cost below $600.

4.2.3 Weight

A very important restraint for the majority of devices, especially those which require acceleration such as in our project is weight. However, this constraint doesn't only impact acceleration, it also impacts the battery range. This is a constraint that we were especially mindful of while selecting our heaviest components, where even a slight percentage difference between similar components had a large impact. We will continue to remain mindful of this constraint as we continue to add auxiliary/miscellaneous parts to integrate our components.

4.2.4 Dimensions

The dimension constraint was very important to be mindful of while selecting our components. This is important for a number of reasons. Firstly, because available space on the scooter is limited and we have to make sure we have adequate space to mount and install all components. Secondly, the dimensions of several of the components have an impact on speed and maneuverability of the scooter, chiefly the motor.

4.2.5 Motor power

Our device will be constrained by it's available motor power because commercially available direct drive hub motors have a limited range of output power. Additionally, there is interplay between this constraint with the weight and cost constraints.This connection comes from the fact that a greater load will require a more powerful motor to reach comparable speeds, which usually means acquiring a motor at greater cost. Motor power will be constrained to 250-350W. Another interesting related constraint is the legal constraint. This constraint is related because in order for our scooter to fall under the same regulations as a bicycle it must not be capable of exceeding specific performance specifications, as well as having other characteristics that we will mention in Greater detail in the preceding sections of this document.

4.2.6 Battery capacity

The capacity of the rechargeable battery pack places a constraint on the implementable power consumption of our device if we want to maintain a desirable range per charge. Battery capacity will be limited to about 2600mAh

4.2.7 Power generation

We are constrained in the amount of mileage that power generation can extend our range. Additionally, we are constrained by the available power generation options. This is due to the generation needing to be easily controllable to not interfere with Motor output.

4.2.8 Safety

We are constrained by safety in the sense that safety must always be taken into account at every point in the project's design and execution. Some of these considerations will include things like limiting max speed, braking force, and motor Shut conditions. Furthermore, just because we might be able to be more aggressive in our design to boost performance, safety must always be paramount.

Because of the safety constraints we must implement safety countermeasures. Planned safety precautions include but are not limited to the following:

- Battery hardening
- Automated motor braking and shutoff
- Limiting Braking Force

These safety countermeasures will be implemented in a number of ways. Firstly, battery hardening, both of our batteries will be in a robust casing to protect them from damage during use. The batteries will be cased either together or individually based on available mounting space and the dimensions of the batteries when encased. The automated motor braking and shut off will be implemented through the pairing of the accelerometer gyroscope with our Arduino microcontroller. Using the gyroscope functionality of our accelerometer gyroscope we will monitor the state of the scooter to determine when it is on its side, at which point the motor operation will be cut off. By using the accelerometer functionality of the accelerometer gyroscope we will allow the microcontroller to automatically switch the motor to regenerative braking mode when the scooter is accelerating more than an experimentally determined threshold acceleration without the motor driving the acceleration of the scooter. An example of this would be when the scooter is accelerating at an unsafe rate down a steep incline. Lastly we will limit the braking force of the scooters regenerative braking to a safe limit, again this will be an experimentally determined braking force. This is not an exhaustive list and during our testing of the scooter we will likely find other safety features that we would like to implement to protect the safety of the operator.

4.2.9 Legal

We must always be careful that our testing, implementation, and project itself fail within the applicable laws surrounding our project. In particular we must be especially mindful of classification criteria that constitute a motor vehicle and a motorized vehicle. To limit the burden and complexity of the legal constraints on our project, we must ensure that our escooter is considered a motorized vehicle and subject to the same regulations that govern bicycle ownership and usage. To ensure this our project must:

- Not have a seat or saddle for the use of the operator.
- Not be designed to travel on more than three wheels.
- Be capable of speed greater than 20 mph on even ground.

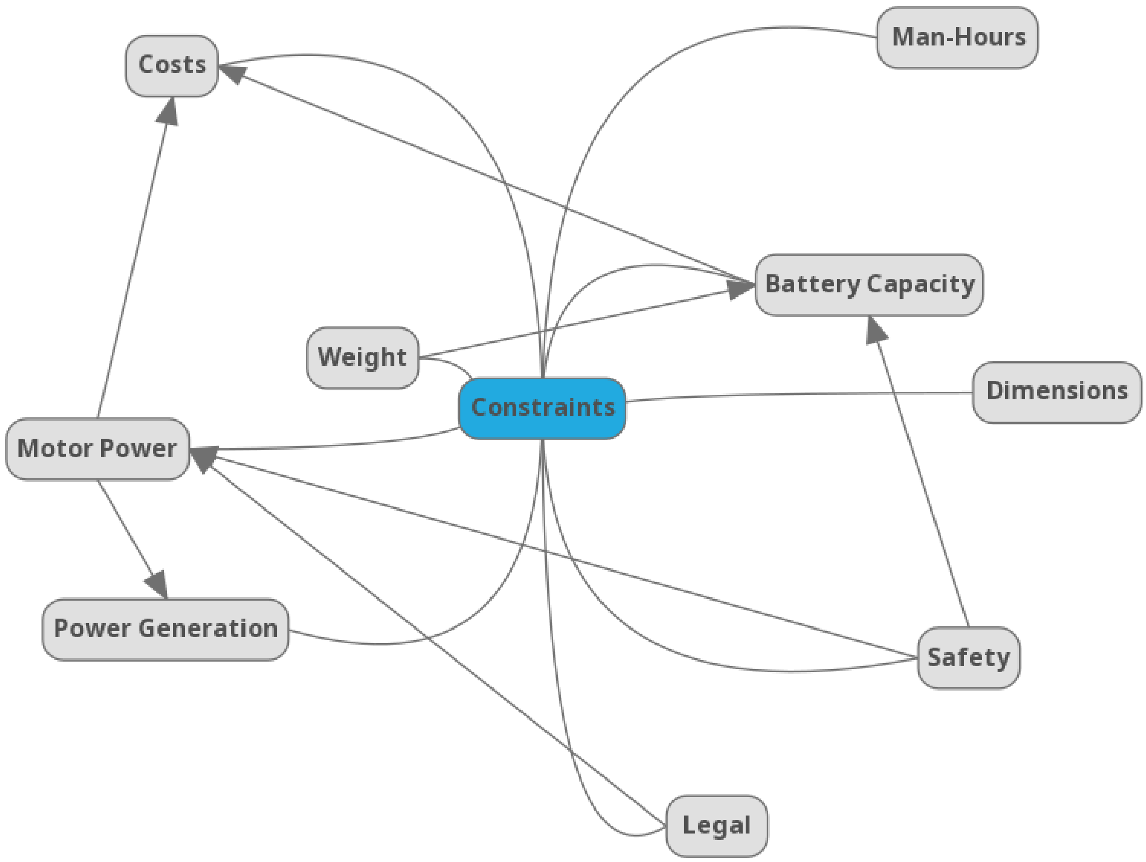


Figure 16: Constraint Interplay

5 Project Hardware Design Details

5.1 Initial Design Architectures and Related Diagrams

A vital starting point to the design of our EZ scooter is to consolidate a flowchart of initial components which would support the list of desired implemented features. As we continue to develop our project and do more in-depth research and testing of our design it may become necessary to make changes, some of these changes might be minor cosmetic/structural changes while others may be drastic feature deletion or addition.

Additionally, the assigned responsibilities may change during the course of the project due to available man-hours, balancing of workload, and available group members.

Our initial design vision may change radically and be quite different from the final implemented design. However, our initial hardware design is nonetheless a very important aspect of our project planning. Most of the research required has yet to be performed and as we are implementing our design we may come upon certain components that are incompatible and that implementation of certain features turns out to be too expensive or impossible within the given time constraints. With that said, our initial design provides firm ground from the start and gives us a proper heading for our future project planning and a clear path to our desired implemented design, which meets the aforementioned specifications.

In the following subsections, the various design aspects of our intended implemented hardware for integration of these components will be covered, as it stands based on the results of her current research. However, it should be noted that our testing, as of yet, has not begun because we are still waiting on the procurement of the majority of our components.

5.2 First Subsystem - Scooter Motor and Throttle Control

The first sub-system of our e-scooter that we will be looking at is the hub motor battery throttle control subsystem. The motor is a direct-drive hub motor which we selected specifically because it will support our goal of including a braking system that is capable of supporting regenerative braking. Our selected battery system is actually two 11.1V batteries connected in series, that will be able to supply the needed power to our motor via the Arduino microcontroller acting as a speed controller. Lastly, the throttle control will be a joystick mounted on the scooter handlebar which will indicate to the microcontroller via a DC voltage The position of the joystick. This DC voltage reading will be used as an indication that the motor should either accelerate or decelerate the scooter.

5.2.1 Motor Design

Our currently selected motor is a three-phase motor with hall sensors. Our first step will be to incorporate our motor with our microcontroller, which will be acting as the speed controller. To achieve this we are going to need to implement several MOSFETs which will be acting as electric switches, alternating which phase is receiving a neutral, high, or low voltage signal at a given instant. Additionally, we will have to implement several drivers, dc-to-dc converters to step up the microcontroller voltage within an acceptable range for the implemented motor, which is from 21 volts to 27 volts. This stepped-up voltage will be then applied at the MOSFET gates and delivered to the motor phase wires. The motor input is a 24 volts nominal voltage, the motor is an 8'in rear wheel hub motor, and it is rated for a power of 250 kilowatts. To control the motor output we will need to connect the microcontroller to the hall sensor wire to be able to determine the current state/position of the motor. This will be done by monitoring the hall sensor for a voltage "zero point" which will indicate that the output state of the microcontroller-motor system should be incremented. These states will continue to cycle and repeat as the motor continues to spin.

The exact implementation of the microcontroller-motor speed control will involve a lot of trial and error. Extensive testing will have to be done to determine the best way to control the motor outfit, such as a variable voltage input or varying the speed of the motor 3-phase switching. This kind of testing will be done by changing the microcontroller's code and trial and error to see which option produces the desired response from the motor. It's also possible that with the use of the joystick we could have the joystick act as a fine tune adjustment to the microcontroller phase alternation speed or motor input voltage.

Lastly, the motor will be powered by two battery packs Of 11.1 V connected in series. With the voltage being delivered to the motor after the joystick is pushed into the forward direction via the microcontroller speed controller implementation. With it exact means by which the motor output is adjusted depending on the results of the previously mentioned testing. When the joystick is placed in the backward position, the motor will be placed into regenerative braking mode. Testing will again have to inform our exact implementation, to account for any changes in the motor behavior (polarity changes for example).

5.2.2 Throttle Design

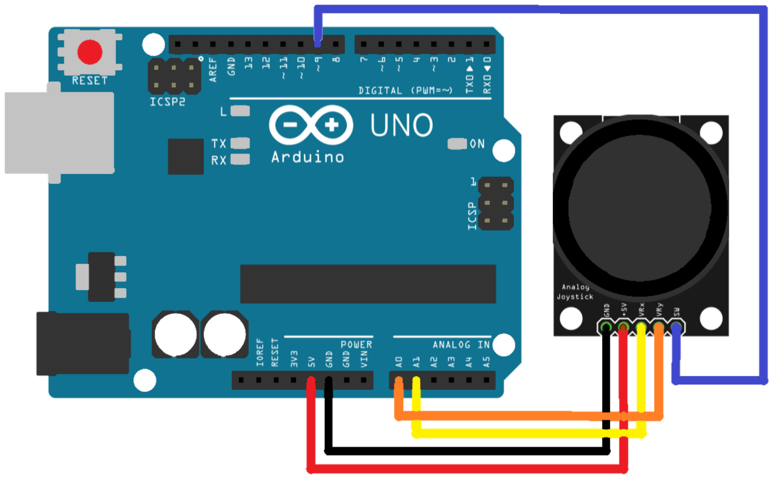


Figure 17: Throttle design

When coming up with a throttle design the first thing that we had in mind was whether or not the throttle would be bi-directional, in order to make switching between motor output and motor brake/regeneration as simple and easy as possible for the user. with that in mind, it became very obvious that a joystick would be most effective in meeting our design goals while remaining very comfortable for the end-user. Its implementation within the rest of our design will be such that as the joystick is pushed forward the motor will accelerate and when the joystick is in the middle position the motor will idle and then when the joystick is in the backward position the motor will work as a regenerative brake until the motor comes to a standstill. At the point where the joystick is in the backward position and the scooter has come to a stop we had an interesting design decision on whether the motor would continue to drive the scooter backward or if the motor output would be cut off. Due to the realistic use of the scooter, its weight, and the ease with which an operator could move our scooter with human power, we decided to implement a motor cut-off after the scooter has already been stopped.

To integrate the joystick into our design we will need to connect the joystick to at least one of the input pins of our Arduino, assuming only bidirectional use, and have the microcontroller monitor the voltage input from the joystick. Given a voltage over a certain threshold a microcontroller will begin to deliver power to the motor via its implemented role as a speed controller. Likewise, the microcontroller will control the motor as a regenerative braking system given a voltage input below a certain threshold from the thumbstick. The above logic of the voltage thresholds and thumbstick direction is assuming that the maximum voltage delivered from the thumbstick is when it is in the most forward position and the most negative voltage output is when the thumbstick is in the most backward position. At this point, there is no planned function for the left and right directions for the thumbstick throttle control. However, something may be implemented for the left and right directions such as a motor cut-off given a certain combination and duration of thumbstick Direction signals. Such an addition to the throttles function would be easy to implement and would only require an additional input pin to be used on our microcontroller and additional code to be written for the intended feature.

5.2.3 Microcontroller Design

The microcontroller will act as the brain of the electric scooter, monitoring sensor data, Computing logic, and driving the motor. The microcontroller will monitor the light sensors in order to automatically control the headlight and taillight of the scooter. The microcontroller will also monitor the accelerometer gyroscope sensor to monitor the tilt and acceleration and deceleration of the electric scooter to automatically control safety features and limit certain parameters of the scooter's performance depending on the selected operation mode and applicable laws. Lastly, the microcontroller will provide output signals that will be stepped up via dc-to-dc converter and act as the gate to the MOSFETs which will feed into and drive the motor. Additionally, the microcontroller will provide the computational power for the aforementioned logical operations, which will be necessary to know the current state/ position of the hub motor.

Some of the aforementioned safety features will include automatically shutting off the motor output when the scooter is on its side, limiting the regenerative brakes' ability to decelerate the scooter to a safe maximum, and providing some degree of automatic braking when the scooter is accelerating without motor output (down a steep hill), and depending on the selected user mode. Additionally, the microcontroller will enable several different user modes that will provide different default behaviors for the scooter. One such behavior will be an assisted scooter mode where the microcontroller will monitor the accelerometer and when the user uses their foot to drive and accelerate the scooter the microcontroller will drive the motor to provide assistance to the user while accelerating. Another intended mode will enact strict safety limits on the performance of the scooter, which will make the scooter more suitable for use by an inexperienced and/or young user. The last intended mode will remove all restrictions not put in place to meet our legal and design restraints and will operate fully without the foot drive assistance feature enabled.

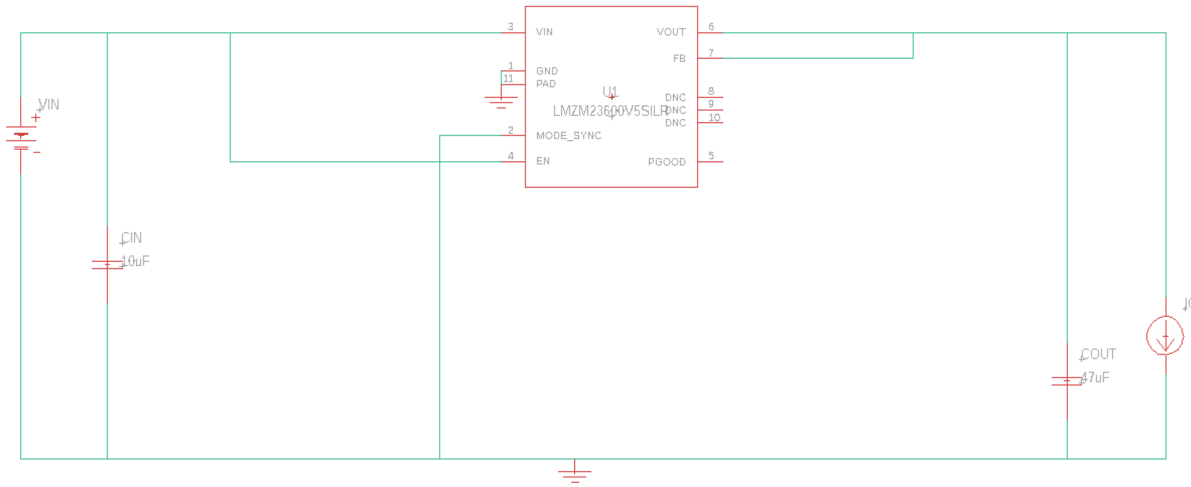


Figure 18: Battery to Microcontroller Regulator

5.2.4 Power Switch Design

When thinking of an implementation to turn on and off the electric scooter we looked into some of the methods they have on the market such as the key switch, power switch and the button switch as mentioned on section 3.2 but none of the choices were good to fit our design, the only one that fit our design was the joystick, making it easier for us since we had the solution in our design already: the joystick. At this point, there is no planned function for the left and right directions for the thumbstick throttle control.

We chose to implement the joystick as our power switch. First to be able to use the scooter, it must take input from the joystick X and Y directions as a password. It is a four direction password and it has a delay of inputting each direction of two seconds, if the password is incorrectly inputted or took too long to input it resets after two seconds and waits for user input once again however once the password is inputted correctly it takes two seconds to turn the scooter on.

5.3 Second Subsystem - Electric Speed Controller Design

In order to design our speed controller, we first need to understand what we need to achieve. An electronic speed control follows a speed reference signal to change the speed of a switching network of field effect transistors. By changing the duty cycle of the transistors or the switching frequency, the motor speed can be changed. First, we look into our brushless DC motor. The brushless DC motors are inside out, the permanent magnet is on the inside, it's the part that spins. The electromagnetic coil is now on the outside of the motor, this gives a lot of efficiency and way more control, the commutator is not fixed this means that we can control it anytime, the electric parts the wire is outside so now it can spin that magnet much faster, it has a triple phase input which means it has three wires as an input, so because of the triple phase we cannot apply a normal differential voltage to it. The three wires create three coils inside the motor, the coil creates a magnetic field and the motor spins due to repulsive forces between the permanent magnets and the coil. In order to make the motor rotate we need to switch the polarity and in order to switch the polarity an electronic device needs it to fast switch the input of the motor such as the Electronic speed controller.

We will design the electric speed controller to make the motor rotated and the most common configuration for sequentially applying current to a three phase BLDC motor is the use of MOSFETs. Using three pairs of power MOSFETs arranged in a bridge structure, each pair controls the switching of one phase of the motor. The high-side MOSFETs are controlled by using PWM. The pulse width modulation converts the input DC voltage into a modulated driving voltage. The PWM frequency is a trade off between the switching losses that occur at high frequencies, to prevent damage to the motor the PWM frequency is used in at higher order of magnitude than the maximum motor rotation speed.

5.3.1 MOSFET Design

As mentioned previously, current is switched to the windings by electronic switches, usually MOSFETs. MOSFETs are one of the most important parts because that is the interface between the software. The MOSFETs are the ones doing most of the work, they take that small level of signal from the microcontroller and drive it up, to actually spin the motor itself. So, the characteristics of these six MOSFETs are really critical for our project. The MOSFET need to be connected to the motor windings, when the motor turns it consumes electricity but it also generates so when we turn off the supply to the motor it keeps spinning and this generates electricity back so the power MOSFET needs to be able to tolerate that voltage coming back from the motor into it. So, it needs to be capable of operating in a reverse mode.

Secondly when bringing the motor up to speed the PWM is use, basically turning the MOSFET off and on really fast for each coil so that electricity comes on slowly and turns off, not full off or full on constantly turning it off and on in this nice slow smooth pattern to get smooth operation for. This pulse width modulation is probably occurring at about 10 to 20 kHz it can be faster, this kHz means that the MOSFET needs to be able to switch off and on that fast without creating noise to be able to give full control of the motor.

Looking at our project the applications we need are two batteries and we need to make sure that the MOSFET can tolerate the maximum voltage that the battery is going to put out. So that the next important aspect is the voltage, the voltage of the MOSFET has to be matched to the requirement of the motor since we are talking about battery powered applications, MOSFET must be rated for the maximum voltage of a fully charged battery. In our project we are going to use two of the HiLetgo *HD44780 1602 batteries*, looking at the operation of the battery being 20 volts depending if it is discharged or fully charged in this case we need something with more tolerance than the maximum voltage of the battery we need at least 30 volts MOSFET. We also look for the low $R_{DS(on)}$ is the resistance from the drain to the source when the part is on, the more resistance it has the more of the electricity it is going to burn off in the form of heat so the MOSFETs are going to get heat. So, the goal is to get them as low as possible. The best MOSFETs today are less than one milli ohm, the lower $R_{DS(on)}$ means more current can be put through the motor.

So, after we sorted some different MOSFETs we took a deep look into some requirements in the datasheet. We first look at the breakdown voltage this is the voltage that we can operate on so we are peaking a 30 or 40 volts MOSFET, then we look at the $I_{D(Max)}$ is the maximum current for the part that tell us how much power we could flow through the device without exceeding its thermal, putting current through a MOSFET is not the difficult part is getting the heat out of the MOSFET for that we look at the thermal performance, we look for a low thermal resistance R theta which is the resistance from the chip inside the part to the outside of the package itself the lower that number is the faster the heat can get out of the part, and the T_j (max), the higher junction temperature is how hot can the piece of silicon get before it melts.

Parts	V_{DS}	$R_{DS\ (on),\ max}$	I_D
IRF3205	55V	8.0mohms	110A
BSC014N06NS	60V	1.45mohms	240A
PSMN0R9-30YLD	30V	1.09mohms	300A
IRF540	100V	0.077ohms	110A

Table 9: MOSFETs Selection

From the above table we have four different good choices we opted for the IRF540 since it gives us a good tolerance over the maximum voltage of the battery, the good side of this MOSFET is that the $R_{DS\ (on),\ max}$ is in fact the lowest resistance of all of the four, we know that the goal is to get them as low as possible to allow better performance at higher temperatures, however is in the range of a good resistance MOSFET, and for the I_D it give us 110A it is enough power capability for our application.

As mentioned previously the one that feats the best for our electric speed controller design is the IRF540 N-channel. This MOSFET can drive loads up to 23A and can support pick currents up to 110A. The good thing about this MOSFETt for our project is that since we are using the atmega328p this MOSFET has a threshold voltage of 4V, this means that it can be driven by low voltages like 5V. They are controlled devices; they turn off and on by supplying the required Gate threshold voltage.

5.3.2 ESC *Design*

To start the design, we took a look at the parts needed for the electronic speed controller. First, we have the microcontroller which is the brain of the entire module, then we have the triple phase bridge made with six transistors we will have to switch the six transistors in order to create our three input signals for the brushless motor the top transistors are connected to a positive voltage and the button MOSFETs are connected to ground, so in order to create the sequence for the brushless motor we have to control two transistors in each state. In order to inter connect the microcontroller and the triple phase bridge we need it a driver, this is because the MOSFET acts like a switch we apply voltage at the gate and it will give us voltage at the source but the voltage applied at the gate has to be higher than the threshold voltage which is usually as 0.7 volts, so since the microcontroller can only give us a maximum voltage of 5 volts in this case without the driver our voltage source will be 4.3 volts but we really want 20 volts at the source, so for this reason we need to add the driver, the driver will get a low voltage signal and it will give it at its output but with a higher voltage. Now having the knowledge on how to design our electric speed controller we know we will need an Arduino Nano microcontroller, some dual MOSFET drivers, our six MOSFET transistors, a voltage regulator, capacitor, and resistors.

5.3.3 Schematics

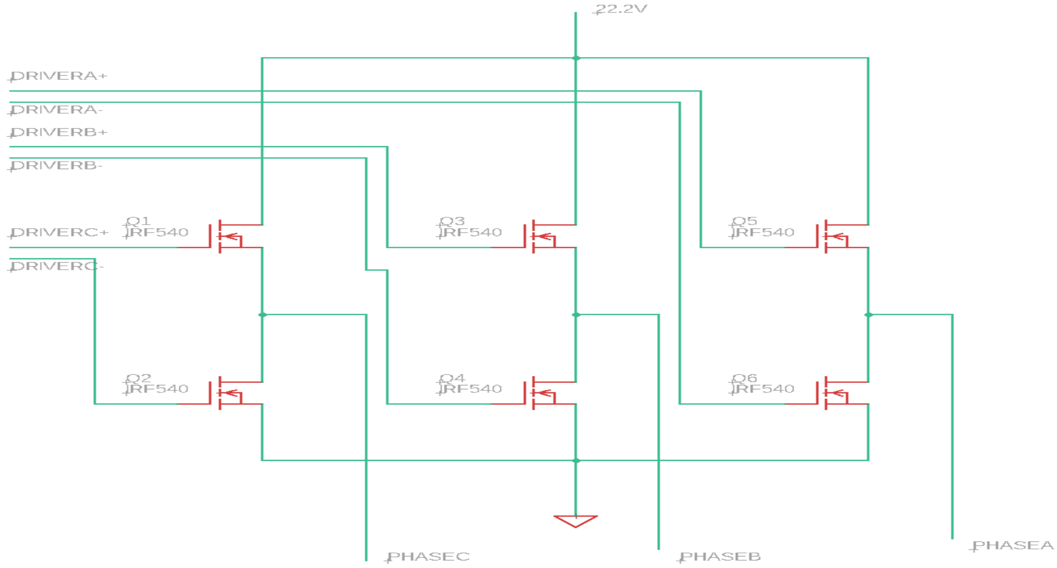


Figure 19: Mosfet Schematic

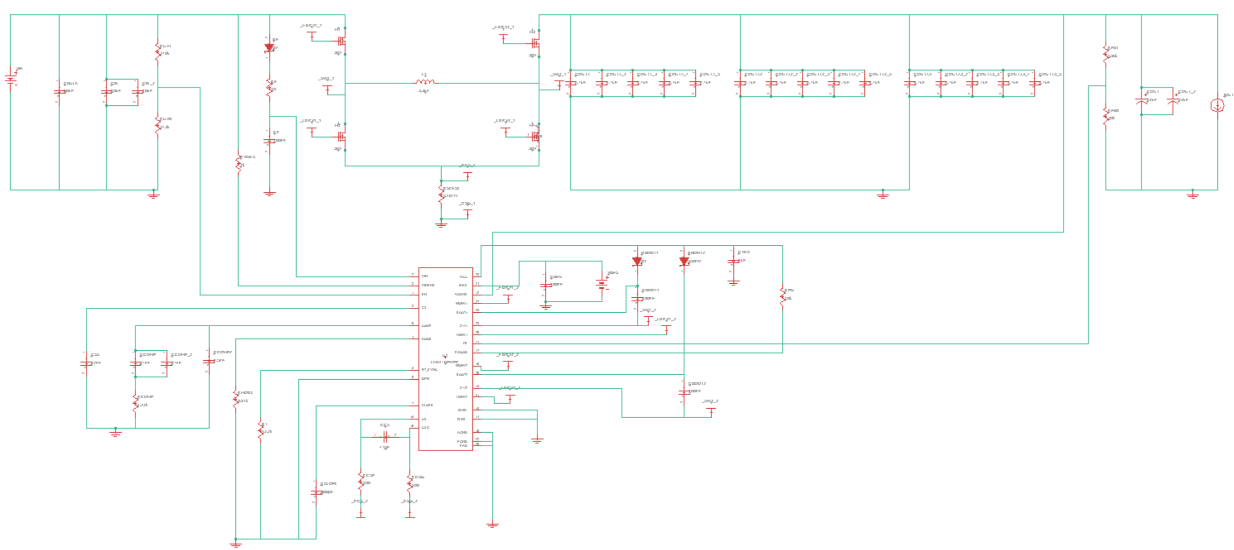


Figure 20: Driver Schematic

5.3.4 Taillight and Headlight Design

The taillight and headlight are a very important safety feature for our electric scooter. We implemented the lights by using the MOSFETs, amplifiers and voltage regulator. The Mosfets are implemented to facilitate the taillight and headlight automatization. Because the lights were substantial loads voltage would be dropped on the opposite side of the MOSFETs and 5V is on the edge of the effective voltage range for the lights, adding the amplifier there would have a substantial effect on the functioning of the lights.

The operational amplifier that we use is the TL084 is one IC that includes four separate operational amplifiers in which each works independently, if any part of the amplifier stops working then unused op-amp uses more power, so the efficiency may be reduced. Its input bias current is low, includes a high slew rate, and trimmed input offset voltage. Its input offset voltage is 3m V, the minimum operating voltage is 7 V, typical operating voltage range +18 V to -18 V, and the input bias current is 20p A.

The lighting system is used to provide automated control over the head and tail lights. This is done by monitoring an externally mounted light sensor module and given a sufficiently low reading, sending a HIGH control signal to the input of an OP-AMP which sets up the voltage to 9V at the gate of a pair of MOSFET gates. These MOSFETs provide a path to ground when opened, from the battery high terminal, through the head and tail lights, to ground. We included two mosfets into the design to allow for the option of controlling the lights individually, as a new braking during daylight hours but due to time constraints we had to prioritize other features.

During our initial design of the lighting system included voltage doublers, which were connected to a regulated 5V. This would allow for greater voltage to be applied to the loads of the head and tail lights, reducing current drawn from the batteries. However, while testing our design we discovered that the voltage doublers LOW output voltage level was ~5V. This meant the MOSFET gates were always open and lights were always always drawing some amount of current, even though 5V wasn't sufficient to power them on. To eliminate this issue we went with the OP-AMP, where the LOW voltage level was 0V, which allowed the gates to properly close.

5.3.4.1 Bread Board Test

The bread board testing was implemented to verify functionality of the headlight and the taillight using the MOSFETs. The layout of the testing on the breadboard is included in **Figure 21,** showing below.

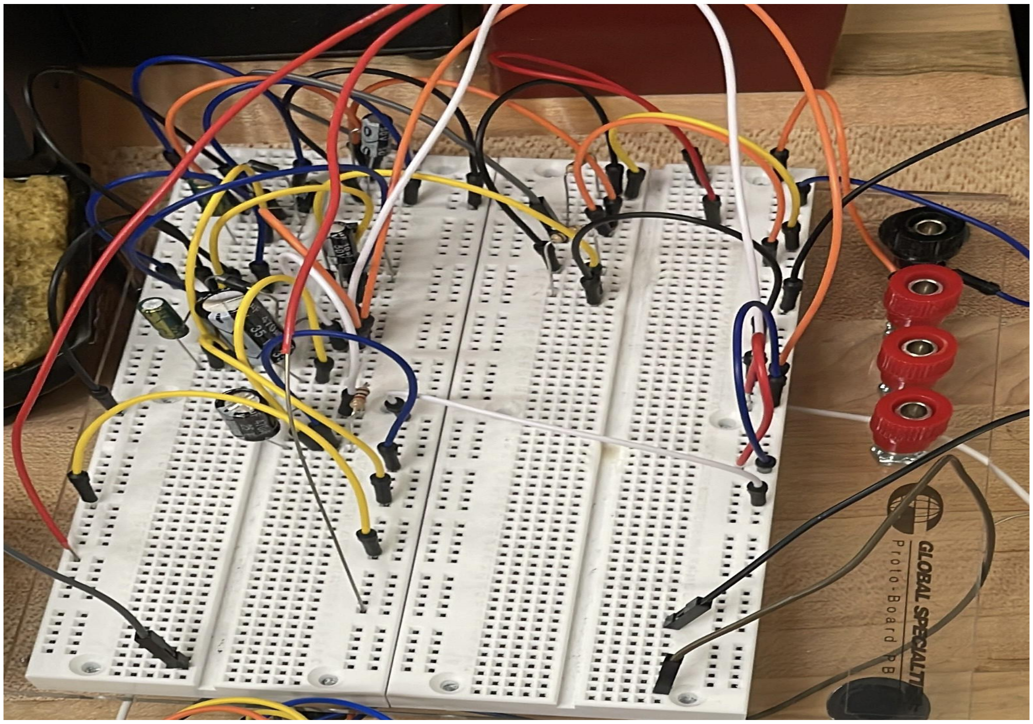


Figure 21: Headlight & Taillight Circuit Test

5.3.4.2 PCB Design

The circuit shown in **Figure 21** is transferred to the pcb, however the final pcb was damaged and since we decided to modify our circuit due to time constraints we transfer our design into a soldering pcb board. The results of the layout are shown in **Figure 22.**

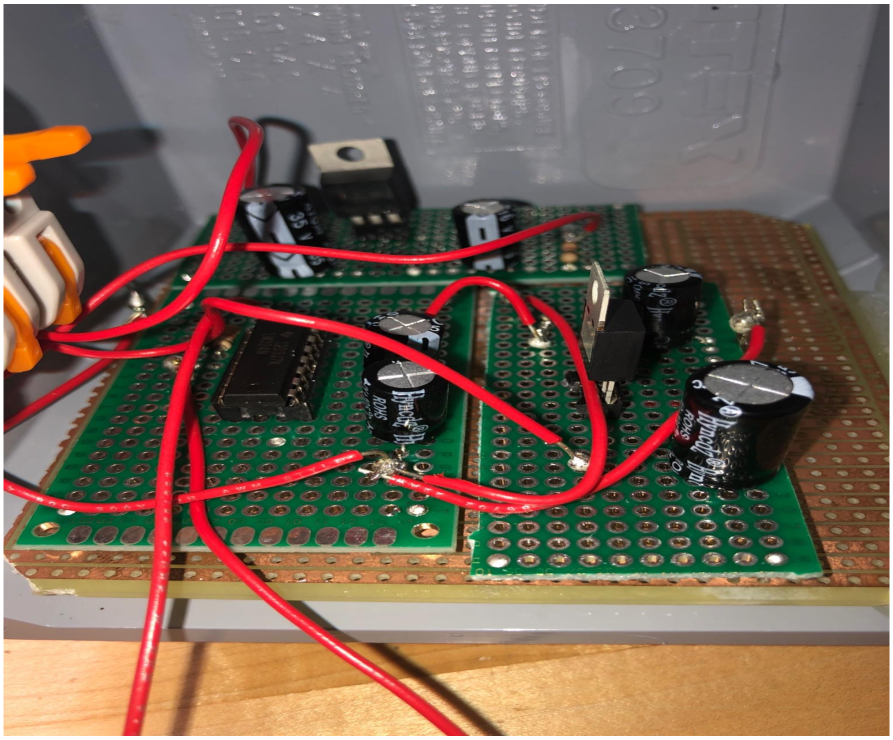


Figure 22: PCB Headlight & Taillight

5.4 Scooter Chassis Configuration

The scooter chassis that we chose for our implemented design has 8-inch wheels, which match the diameter of our hub motor, a 23-inch long deck, 17-inch wide handlebars, and is foldable. This scooter is meant for adult riders and can bear a heavy load of 300 lb. This will give us plenty of leeway with our added components and the weight of an adult Rider. Another attractive feature of this scooter is that it has a considerable amount of ground clearance under the bottom of the deck and the sides of the handlebars support are relatively clean, this gives us a space to Mount our components.

For the assembly of the E scooter, the direct-drive hub motor will replace the back wheel. This should be relatively easy because the tire matches the size of the currently installed tire and a simple replacement is all that would be necessary. The LCD will be mounted to the handlebars and will be angled in such a manner that the operator can easily view the screen. The throttle will be mounted to the right portion of the handlebar and will be placed in a location that will enable the operator of the scooter to easily reach the thumbstick. The batteries will be encased underneath the scooter if there is sufficient clearance for the batteries and the battery casing. Otherwise, the batteries will be encased separately but still connected in series and mounted to opposite sides of the bottom of the handlebar support. This will allow for the next best-case scenario for weight distribution. The light sensor will be mounted close to the LCD positioned upwards. The microcontroller, accelerometer gyroscope, Bluetooth module, and PCBs which include the voltage drivers and MOSFETs that will facilitate the microcontroller acting as the speed controller will be housed and mounted together on the lower portion of the handlebar support facing the operator. And lastly, The headlight will be mounted on the handlebars facing away from the operator and the tail light will be mounted on the back portion of the scooter facing away from the operator.

5.5 Additional Hardware Features

There are additional hardware features that we have an idea of for our EZ Scooter. These features can be included in many different ways of using the PCB or even additions to the overall structure for the connections of the system. Some of the features will benefit the overall use of the scooter and other features will allow for the user to understand the features better in a more direct information approach.

5.5.1 Battery Voltage to Microcontroller

Oftentimes we are going to need to know the battery voltage level within the microcontroller itself when it is charging and when it is not charging. This will allow us to have a general idea if the rechargeable battery cells need to be charged or not and will prevent any type of faulty to the charging system. We are considering adding a PIC microcontroller that will contain a 4-digit seven segment display and USB charger. The 4-digit seven segment display will be used to show the voltage of the battery which will be able to read voltage of 24V and the USB charger is used for getting the battery voltage to the ADC pin of the PIC microcontroller.

5.5.2 Headlight

Almost any type of motorized vehicle contains a headlight to be able to see during the nighttime and it is also used as a safety precaution for oncoming traffic such as cars, bikes, people, etc. it can be very hard to see at nighttime and pedestrians will know to move when you are driving towards or away from them.

The headlight that will be obtained is by a brand named Cyc tree that will contain a control switch to turn the headlight on and off on command. The headlight will also use the voltage from the rechargeable battery cells and will need a current and voltage to be able to keep a consistent lux for visibility up to 15 meters.

5.5.3 Tail Light

In relation to the headlight, the taillight is just as important for the ability of visibility during the nighttime. The purpose of the taillight will be to provide info to other motorized vehicles or pedestrians that we are in front of them with the use of braking a red light will appear signaling braking. The taillight will be able to show a consistent amount of lux that will illuminate for the safety of the user and any other vehicles nearby.

The taillight will have to be analog with the use of a timer pulse circuit that will have the ability to send signals to the taillights whether it needs to turn on or off. The signal will most likely be sent through the capacitor that would connect to the brake trigger that would signify it to turn on the red light. A test circuit will need to be created for this instance to test its functionality.

5.6 Summary of Hardware Design

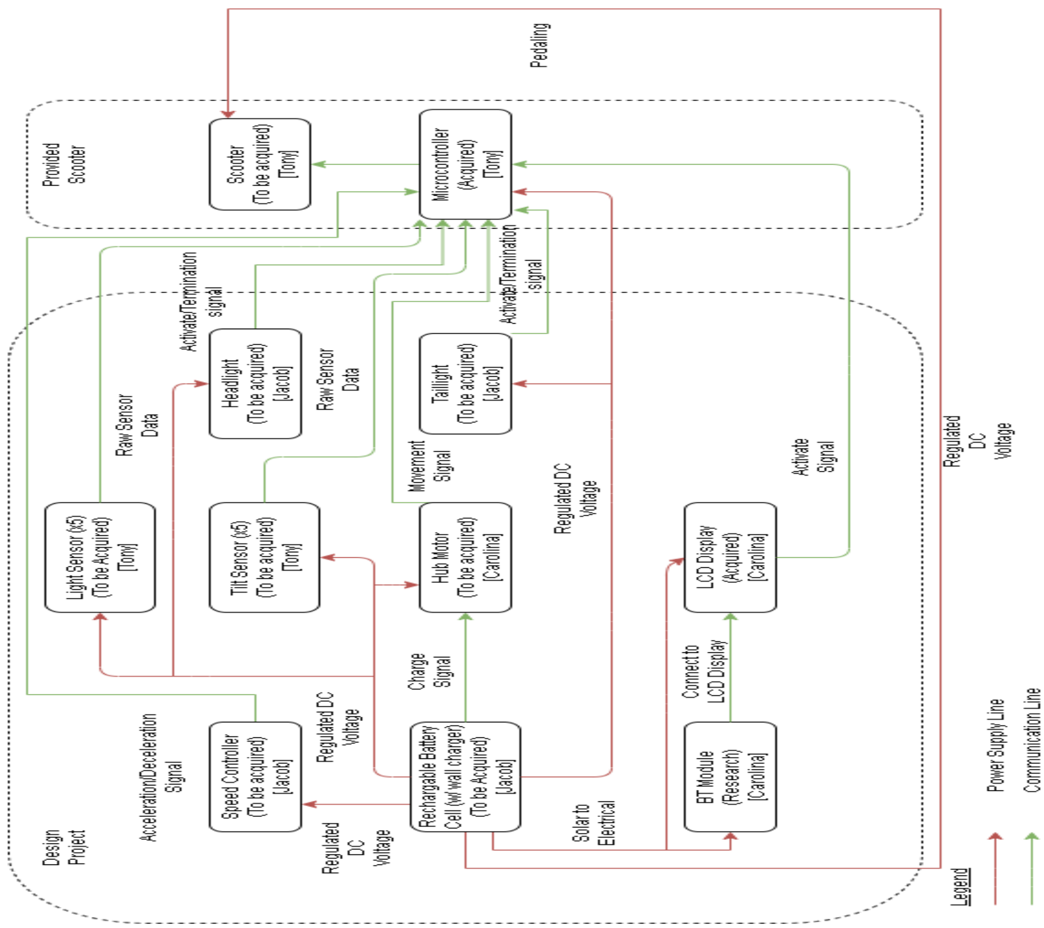


Figure 23: Hardware design

This block diagram shows and describes the flow and necessary components within our EZ - Scooter needed to complete our project. Our biggest focus is the sensors being able to work with the controllers that we will install and will continue our research in which is best and efficient for our project.

6 Project Software Design Details

The software part of the EZ smart scooter will be implemented on the Arduino Uno REV3 will be able to take in different types of inputs and outputs that will be utilized within our code. The code that will be implemented will have different tasks that will keep the scooter functioning such as, sensor input from the HGLRC Flipsky Speed Controller to be able to read the incoming speed, the speed for the scooter will then be displayed on the LCD display for the user, and the motor will be receiving signals in rhythmic waves that will give info in relation to the speed.

6.1 Software Functionality

The purpose of our software is mainly oriented around two main tasks with one being able to display the information on the LCD display for the user to read and having the ability to control the motor in accordance with sending signals. The functions and processes of this are done through the ATmega328P which is compiled and sends the code from Arduino Integrated Development Environment (IDE).

6.1.1 Sensor Input Calculation

The sensor allows the microcontroller to calculate the speed of the scooter, which will be read from the HGLRC Flipsky Speed Controller. The Arduino Uno REV3 will be able to take in the input by using a function called pinMode and we are going to use that function to call it “pinMode(flipsky, INPUT)”. The speed controller also known as the flipsky will be defined using the #define a function which will be connected to an analog input. Timer interrupts will be included as well for the scooter such as TCCR1A, TCCR1B, and TCNT1 will be set to 0. The main tasks that need to be assigned are the flipsky counter, the timer setup and the dimensions of the microcontroller.

In this next part we will be calculating the speed of the scooter using the interrupt function ISR(TIMER0_COMPA_vect) which will be always running and needs to be continuous due to the fact that the speed of the scooter needs to be constantly updated. The flipsky number will be read as a binary number being a digital HI or a digital LO that uses the digitalRead() function. The flipsky counter value for HI will calculate the miles per hour that use a table of time but if the value comes out to be positive then the count will decrement and the LO value will increment except when the timer is greater than 3000 and the mph will deplete to zero.

6.1.2 LCD Display

The LCD display will show the calculated speed values from the sensor input calculation. There will be functions that will help format the LCD display to its best degree. The function that I am referring to are these:

Function	**Task**
Serial.write(“String”)	**Look at the required string that needs to be printed and will then be printed on the LCD display.**
Serial.write(12)	**This clears the LCD Display entirely**
Serial.write(13)	**This allows for the next line to be shown on the LCD Display.**

Table 10: Serial.write functions

The LCD display will have two different pages, the home page and the GPS page. On the home page that will show the title of our scooter “EZ Scooter” on the first row of the 16x2 display. In the second row, it will show the miles per hour with it constantly being updated by the gps module every second and the battery percentage being constantly updated every second to allow the user to know if the scooter needs to be charged or not. On the GPS page, it should display the user’s current location using the latitude and longitude on the first row of the display. On the second row, it will show an active clock that uses military time that is being updated every second as well. The different pages are accessible by pressing down on the joystick button at any moment.

6.1.3 Tilt Sensor

The tilt sensor will be focusing on when the motor will turn off depending on the angle it is on. In our code, we put in values to show that if the scooter bends to the side at a 30 degree angle or higher then the hub motor will turn off for safety precautions. If the angle of the scooter is less than the 30 degree angle then it will continue to wait for an input from the user to input forward, back, or nothing on the joystick. When testing this I had to use led lights signifying if the values are read correctly when the tilt sensor is tilted at a certain angle and within the specified range. When the led lights turn on then the angle is at 30 degrees or higher and if it is off then the tilt sensor is within the safety range.

6.1.4 Light Sensor

The light sensor focuses on signaling the headlight and tailight to turn on and off by measuring a specific lux value range. If the lux range value was higher than 1000 lux then the lights would turn off accordingly so it would save battery life. If the lux range value is less than the lux value of 1000 lux then the lights would stay on until the lux value has gone over 1000 lux again. An alternative I was looking for this part was to leave the light always on no matter what, so we would not have to worry about any specifications malfunctioning at a bad time, but I went ahead with the first option due to the fact that it saved more energy for our battery and therefore allows for our scooter to last longer from 1 hour to 2 hours.

6.2 Algorithm Description

This code will be represented in two different sections that will work towards two different tasks and obviously each task that is assigned will contain variables and constants that need being defined. Anything related to the inputs and outputs will be referred to the headers and library and these will always be at the top of the code like all other languages. Below is a representation of how each task will be compiled shown with descriptions.

6.2.1 Speed Calculation

Task: The Flipsky constantly reads and calculates the speed values that is using the ISR(TIMER0_COMPA_vect)

- Headers/Libraries
- START
- Read the digital flipsky signal
- If there is a voltage detected from the flipsky speed controller
    - If the flipsky counter is equal to 0, then it must calculate the miles per hour (mph) and the resets the timer to 0 and when resetting the counter we must put it as its max value Else if the speed counter is greater than that of 0 then the counter will decrement
- If the timer value gets bigger than 3000, then the mph must be set to 0.
- Else increment the timer

· END

6.2.2 LCD Display Function

Task: The LCD Display will be able to read and display information that is being sent from the speed controller.

· Headers/Libraries

· START

· Clear display

· Print the Odometer which displays the speed

· Print the speed that was originally calculated and found to through the speed calculations function

END

6.3 Coded Flow Chart

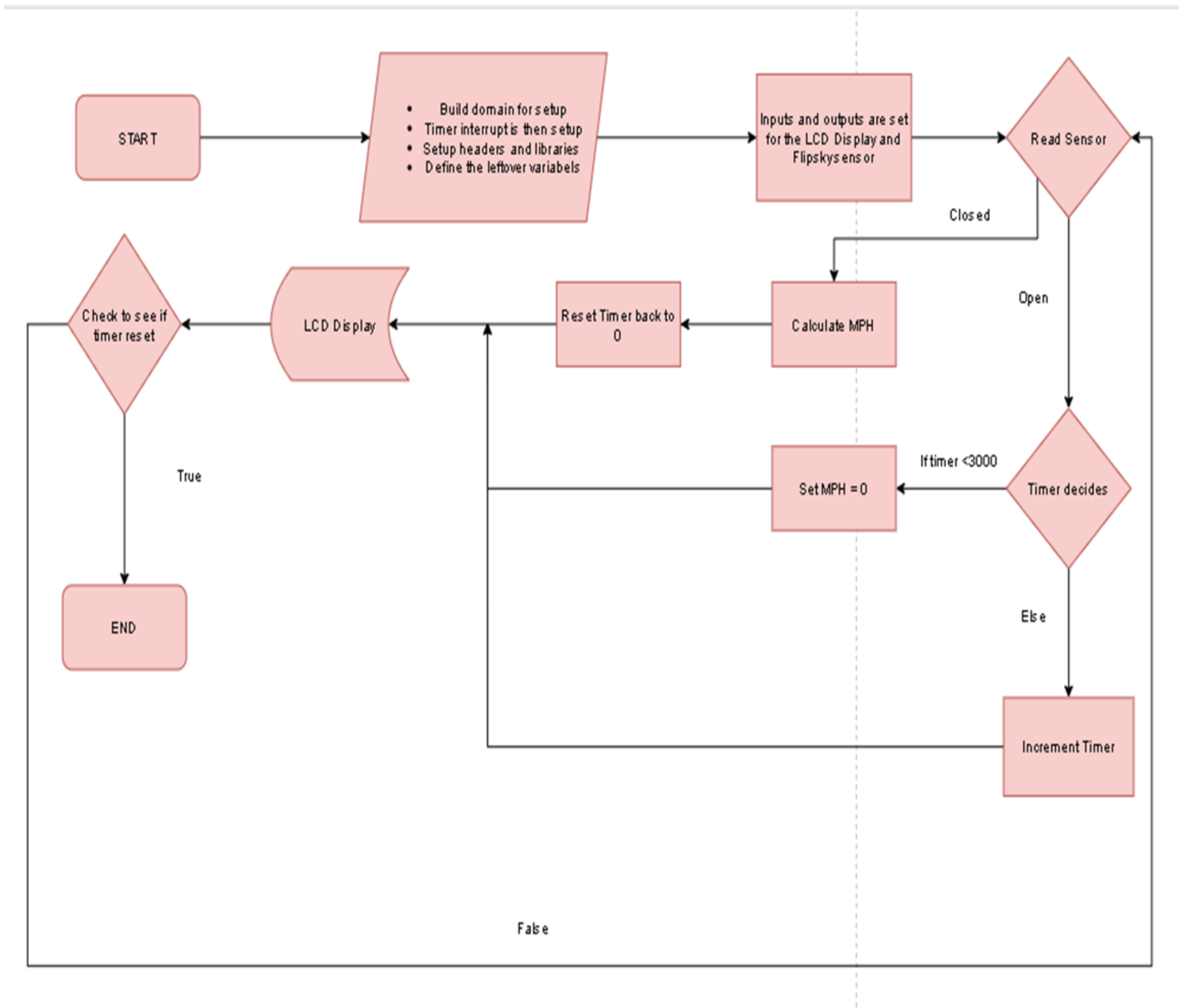


Figure 24: GPS Module Calculation Flow Chart

6.4 Control Signal Testing

For this section, we will be focused on the hardware aspect of testing the software functionality to make sure the tasks that we created from the code are working correctly. For this testing process, we will be using the ATmega328P to test out the software functionality. This is to ensure that we do not run into any future errors within our code or software in general. We will also not be using the Arduino UNO in the final design of our project.

6.4.1 Throttle Control Testing

One of the first tests that we had performed was to make sure that the PWM signal being generated from the ATmega328P is producing properly and able to be controlled the throttle, which is the joystick module in our case. To simulate the throttle control, we must first connect the DC voltage to the analog input of the joystick to read any signals. The provided Oscillscope will then be used to observe the output of the digital outputs for the relevant PWM signals that are registered. First, we will leave the joystick disconnected and will code built in digital throttle values to test if any type of PWM signal will be produced. One of the main parameters we had to set as well was to check the frequencies of the pulses, in our case we chose to test it with 8000 Hz as the set frequency.

The next test will be using the joystick and an adjustable DC supply that will vary values from 0V to 5V, which are the same parameters as the joystick. The positive terminal of the DC supply will be connected and focus on the analog input on the ATmega328P while the negative terminal of the DC supply will be connected to GND of the ATmega328P. The code will then be modified to fit with the new features of the joystick and will be compiled and uploaded to the ATmega328P. We will then adjust the voltage from 0V to 5V to see if any type of increase in the positive pulse is showing on the oscilloscope while keeping the stable set frequency of 8000Hz. There is a chance of the DC supply distorting the PWM signal when the the DC supply starts to reach 5V. To be able to prevent this occurencem we can increase the throttle divider in the code by 10 to not have a high duty cycle.

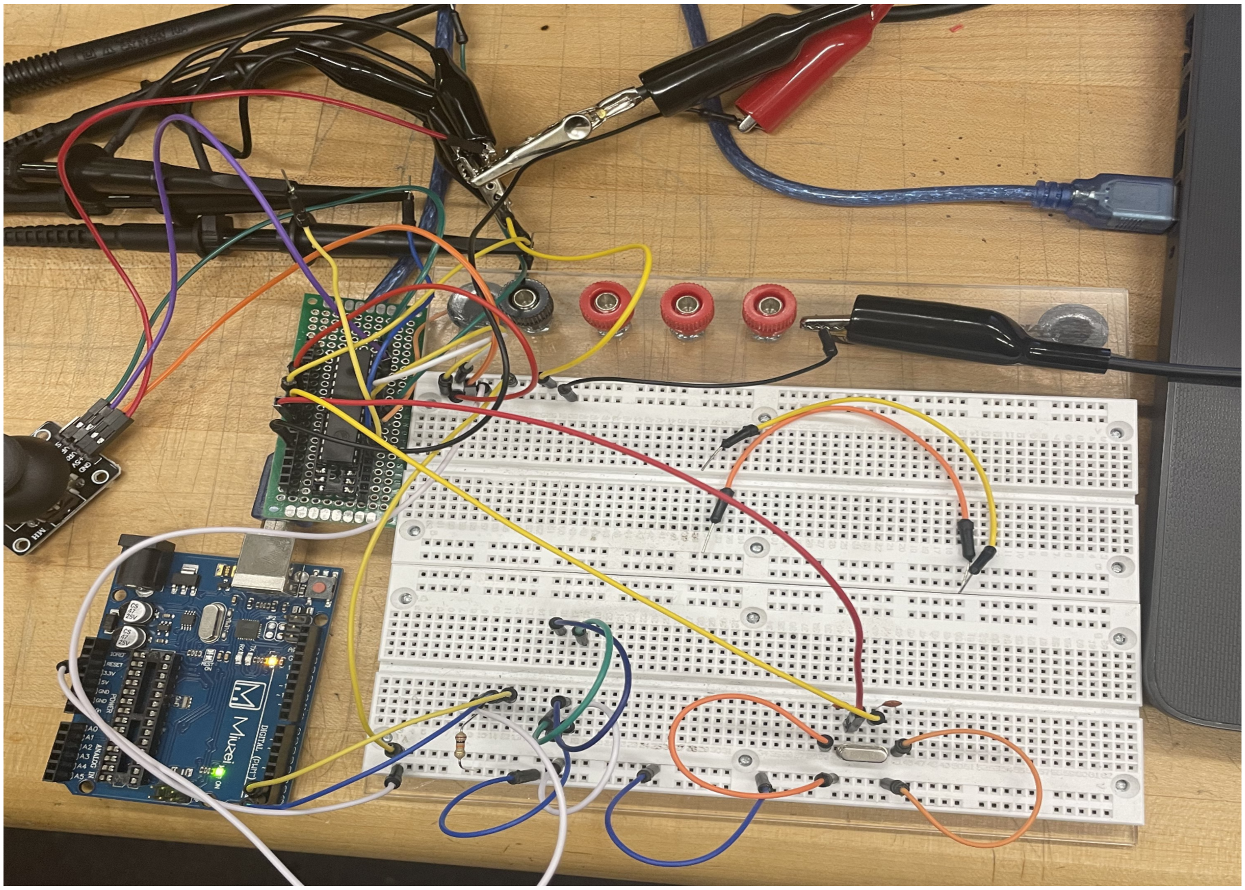


Figure 25: Joystick Throttle Testing Setup

During the testing the oscilloscope should show a waveform with a PWM signal that has a small pulse width when setting the DC supply to 0V. As the voltage is increasing, the pulse width should be getting bigger since the voltage of the joystick is increasing as well. Finally when 5V is reached, the pulse width should be at its limit and will show if the code is working as it should or not.

Along the way, we did run into some issues that set us back a little bit with one of them being the oscilloscope showing a different frequency value than the set 8000 Hz that was set within the code. After troubleshooting and changing some values around we realized that the time it takes to process the void loop is lagging due to the varying duty-cycles and calculations within the loop to be able to help with the PWM signals. This makes the compilation longer and causes the ATMega328P to signal a HI for the digital output when it was trying to go back to a LO signal within the loop.To fix the issue we went ahead got rid of some unnecessary calculations that were in the loop and instead put it in the void setup portion of the code. After testing again with these changes, we found that the frequency was reading correctly as 8000 Hz, which is the same amount as the given value within the code.

6.4.2 LCD Testing

The LCD display is one of the most important features to test as that is our main source of displaying the title of our scooter “EZ Scooter”, the miles per hour, the battery percentage, the location using latitude and longitude, and an active clock that uses military time. To make sure that the LCD is displaying correctly we can simply create a serial function such as Serial.write(“Hello World”) which will be used within the code. After compiling and uploading, the LCD should display the text “Hello World” to show that the LCD is connected to the breadboard correctly in accordance with the ATmega328P and that the code works well.

6.4.3 Tilt Sensor Testing

The tilt sensor is a main component for the user to feel safer and more secure when riding our scooter. When testing the tilt sensor, we will connect the the tilt sensor directly to the breadboard and wire the connections to the proper parts of the ATmega328P and make sure it is powered by a DC supply of at least 5V and grounded. Within the code, I will use analogWrites to write the value to the analog inputs of the tilt sensor within if and else statements. There will also be led lights connected to the breadboard each having 1k resistors to limit the current and represent the motor turning on or off. If the angle is more than 45 degrees or -45 degrees than the led lights will turn on signaling that the motor is now off and will not continue to move, else the led lights will be off signaling that the motor is still on.

6.4.4 Light Sensor Testing

Testing the light sensor will allow us to determine whether the headlight and tailight should turn on or off depending on the lux value that it reads when it is properly connected to the ATmega328P and at least 5V and grounded properly. We will connect the light sensor to the breadboard in which the ATmega328P is connected to and also connect some led lights to act as our headlight and tailight with 1k resistors as well to control the current. Within the code, we will again add analogWrite functions to say if the lux value is greater than 1000 lux, then the led lights will turn off (LO), else if the lux value is less than 1000 lux, then the led lights will turn on (HI). The led lights turning off will show that the rider is most likely outside during the day where the light will not be needed. The led lights turning on will show that the rider is either inside or riding at night time. These two different options of it turning off or on is helpful in controlling the battery usage within our scooter and will allow the user to ride longer. The lux values will be programmed within the code to be able to ensure that the output is made correctly.

6.4.5 GPS Module Testing

The GPS Module is one of the key components to the EZ Scooter in which it allows for the LCD to display the updated miles per hour, location, and time. To test this we are going to need to use the TinyGPS++ library that gives us access to different GPS functions that will allow us to display the output based off the signal output. First we must program in the ATmega328P to use a gps function that can help us find our longitude and latitude as an example for us to make sure the GPS module is working properly. The function that we will be using is the gps.f_get_position(&lat,&lon); function and a few serial prints to print out the result of the latitude and longitude. These results will be printed out on a string value of up to 6 decimal points. Finally we will need to take the setup outside for the GPS module to be able to find a signal and see the results when moving around outside and seeing if the latitude and longitude is constantly being changed

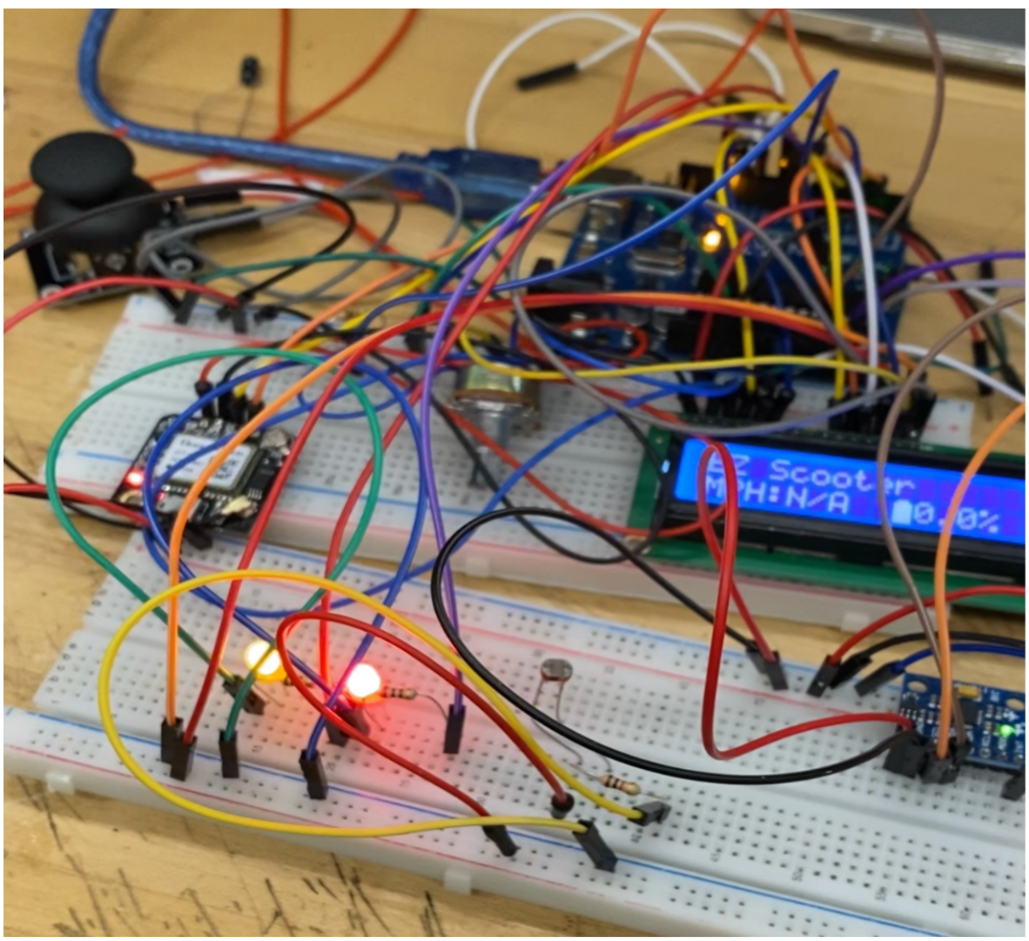


Figure 26: LCD, Tilt/Light Sensor, and GPS Testing Setup

6.5 Additional Software Features

When it comes to adding additional features we thought of a couple that would make the experience riding the EZ Scooter a lot better with some of them able to be added to the code for the Arduino Uno REV3. These features are more oriented towards quality of life and allows the user to have a better and easier experience when riding the scooter. These features include rear sensor for traffic, emergency alert, LCD for battery charging, and GPS. Keep in mind that these features are not to be expected but ideas to use if we have spare time/money for the project to expand upon.

6.5.1 Rear Sensor for traffic

The feature of adding a rear sensor is not needed but it helps with any user, especially new people that are trying to learn how to ride the EZ Scooter. Having more security helps the user feel safer and would assist them in parking the scooter in spots and warn the user when it is close to an object with a beeping noise. The beeping sound will increase in speed the closer they get to an object, which will signal the user to slow down or stop before hitting it when backing up. It should not cost too much and is a high priority on our additional feature list that we hope to bring to the table.

6.5.2 Emergency Alert

This feature allows for emergency situations to feel more at ease. Let's say you are on a stroll with our scooter and an accident happens where you fall off the scooter or hit someone else with a vehicle, you are able to press a button that will alert local policemen and ambulance to come as soon as possible to assist you in need such as a Life Alert but for the ability to use it anywhere anytime at any given time. The scooter is filled with secure features that will allow the user to feel right at home and safe when going for a ride.

6.5.3 LCD Display for Battery Charging

This feature is very straight forward with what we would like to bring to the table with being able to see how much the batteries have been charged so you'll be able to know how long you can go on a ride for or how much longer you will need to charge the scooter for before leaving. This is a little more complex and more expensive, but for the ease of the user it is definitely worth it rather than using a LED light to indicate if the scooter is charging and when it is done. For example a simpler and cheaper solution would be using LED lights with one being green for being fully charged and one being orange sharing that it is currently being charged and red for when its low on battery or about to run out of battery.

6.5.4 GPS

A GPS can be either a very hard task to achieve or very straight forward depending on the approach you want to go with it. You could create/code a basic GPS system for the Bluetooth module to be able to display on the LCD Display and that represents a certain range in which people can travel to. The other approach is what's probably best and will save us time and money, which is buying a GPS system and attaching it onto the scooter to share where you are and how far you are from the designated location the user enters. The GPS can also only be accessed when having a signal and a signal is usually found when you are outside for it to be able to reach a satellite signal. Once connected then the LCD Display will be able to work properly and display the calculated measurements through an assortment of GPS functions and code such as "gps.module.mph" that focuses on calculating the miles per hour.

7 Project Testing and Prototype Construction

For our EZ Scooter project, a PCB is needed to test and measure a similar model to the real-life model. This should be able to test the different functions within the MCU and PWM to be able to act as a power management hub for where the code is read and processed through the LCD Display and result in giving off PWM signals. These tests will entail the hardware use and software use of its capabilities in a real life setting.

7.1 Prototype PCB

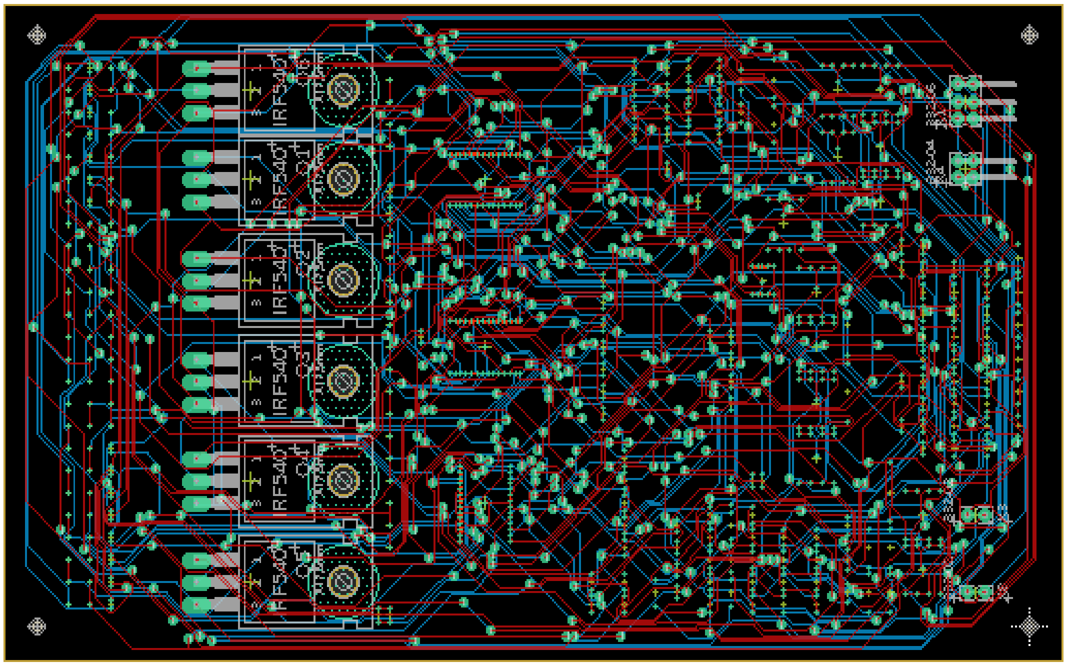


Figure 27: Example PCB Prototyping for MCU and PWM Signal

7.2 Hardware Testing

Hardware testing is one of the most important aspects of the project where we will make sure the hub motor works properly for the EZ Scooter. The hardware testing will first be approached on Multisim™ Simulation for circuit simulation to see if there are any errors we did not see originally when creating the circuit. Going over any errors will allow us to avoid any problems in the future with damage to the different components. We will test all of our components working in unison to be able to successfully ride the EZ Scooter. The electrical side of things with hardware will be brought inside using lab equipment.

Motor Testing

Objective: The objective of testing the hub motor is to make sure that the EZ Scooter will properly work and run safely for all users. These different tests will require the use of a Digital Multimeter that we can use from the UCF lab. The positive probe will be put in the VOhmHZ terminal, while the negative probe will be connected to the Common terminal.

Environment: The hardware testing will be done at a mix of places with one being at my household for basic functionality, but for measuring in depth measurements for the electronics we will need to use the UCF laboratory located in Engineering 1, Room 456. The lab equipment that is given within the lab are:

- Tektronix Oscilloscopes
- Tektronix Dual Arbitrary Function Generators
- Tektronix DMM 4050 Digital Multimeters
- Keithley 2230-30-1 Triple-Channel Power Supplies
- Dell Precision 3420 Computers
- SMD Rework Station
- Soldering and Desoldering Stations
- Digital Microscope Inspection Station

Procedure: To test the hardware for the EZ Scooter, the steps for the procedure are:

1. Make sure that everything is properly connected to the terminals, and nothing is loose that will disrupt the testing.

2. The digital multimeter will measure the rechargeable battery cells with the positive probe on the positive terminal of the rechargeable batteries and the Common lead on the negative terminal of the batteries. The hub motor takes up to 24 V so the multimeter should read around that number as its maximum. If the batteries are read for being lower than 12 V then we will need to charge the batteries.

3. For the hub motor, we will need to connect the oscilloscope to the positive terminal of the motor and the negative terminal to the negative end of the hub motor.

4. For the throttle, we are going to want to turn on the scooter and proceed to hold down the throttle at a steady rate until the hub motor moves. Once the motor moves then the oscilloscope will show the variety of waveforms within the PWM (Pulse Width Modulation) that reaches up to a maximum of 24V and a duty cycle that reaches up to 10% at most. At this low speed (low speed signal), each pulse should come up around every 11 microseconds or so.

5. Now pull down on the throttle to increase the speed of the hub motor to its almost max potential to see what kind of waveforms will appear for a high speed (high speed signal) with a duty cycle of a max 90% and PWM of 100 microseconds.

6. For the headlight and taillight, we will just need to turn on the EZ Scooter and turn on the switches on the handles of the scooter and check to see if the headlight and taillight will turn on and off.

Conclusion: The readings on the oscilloscope are correct then the hub motor should work perfectly along with the sensors and if any errors occur then we must troubleshoot.

<u>Throttle Testing</u>

For throttle testing, we can first look at the hall sensor that we will adapt to the scooter. The hall sensor focuses on helping the controller to drive the motor. With the testing of the hall sensor we must connect the voltage source and digital multimeter to the positive volt source (Pin 3) and connect Pin 2 to ground. The output was then measured by the digital multimeter on Pin 1 which is connected to the positive terminal of the digital multimeter.

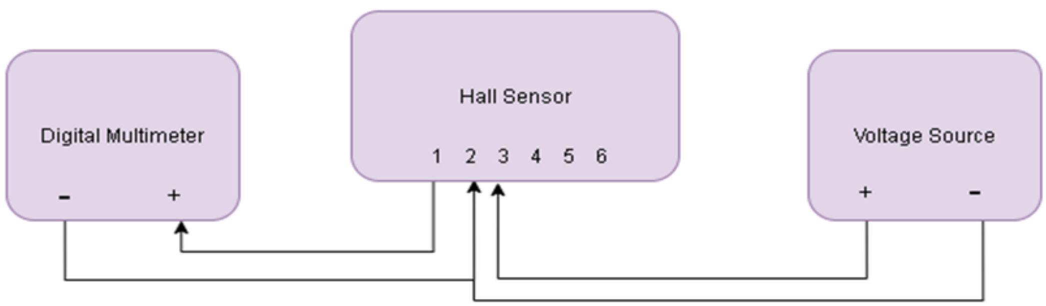


Figure 28: Hall Sensor Testing

The next part we are going to look for throttle testing is the battery level display that will be attached to the throttle where the LCD display will be. In *Figure 27*, it references how the testing will be set up to measure the battery display. The positive terminal of the voltage source will be connected to Pin 3 while ground will be connected to Pin 2. The battery level will display different levels of voltage using LEDs as reference. Red being low battery (6V or lower), yellow being half battery or lower (12 V or less), green indicating full battery or high battery level (greater than 12V all the way up to a maximum of 24V), and no LED for when it is empty (0V).

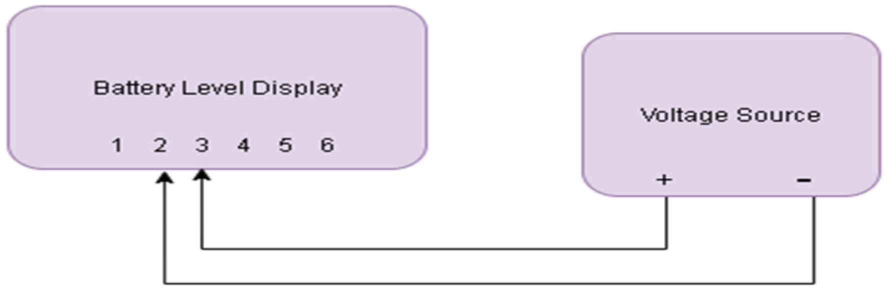


Figure 29: Battery Level Display

In the table below we can see what voltages are required for a specific LED to light up accordingly showing the battery level display of red, yellow, green, or nothing depending on the voltage level. With the hub motor having a rated voltage of 24V needed to rotate at full battery. When the battery level starts getting really low close to nothing then the hub motor will not rotate the wheel at all. It is highly recommended that the battery stays above 12 V or above for maximum efficiency during the ride and not having to run into any problems while riding the EZ Scooter, which will also give the best and safest experience.

Battery Level Display	Minimum Voltage Required
None (No color)	0V
Low (Red LED)	6V
Half (Yellow LED)	12V
High (Green LED)	13V

Table 11: Battery Level Display VS. Minimum Voltage Required

Another important part that we test using the throttle in this case for our project is the joystick. We are not only using the joystick to control the motor but to lock and unlock the operation of the motor. The implementation of this operation is to add in our software a safety feature, by adding a joystick combination. The joystick combination starts with a combination with right or left and this is because if you are already operating the motor and want to shut off the motor we do not want the driver to push the up or down because that is to go forward and the down position is to break. So for this reason the combination that we create and we test is right, right, left, up, left and then the motor will operate. We can test if the code works unlocking the motor by looking at the waves on the Oscilloscope as shown in **Figure 28.** We will know if the motor is unlocked even if the motor was not operating.

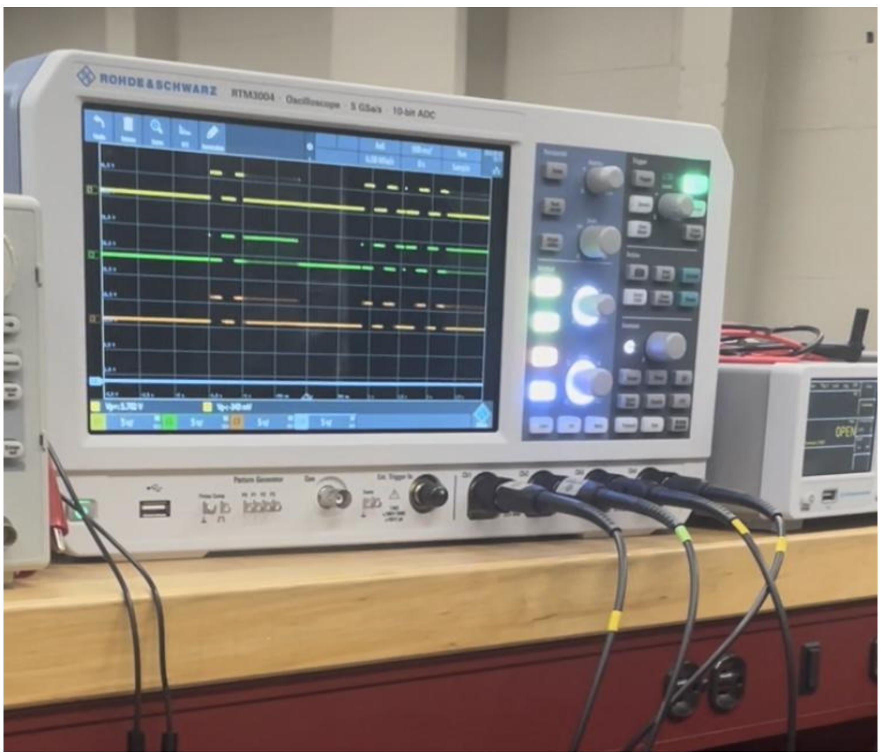


Figure 30: Joystick Test

<u>MOSFET Driver</u>

The final piece of the puzzle are the MOSFET drivers that will have the goal of having similar frequencies and duty-cycles of the PWM signal that the microcontroller gives off

(Arduino Uno Rev3). This will be done by using a 1kOhm resistor to be able to simulate and relate the hub motor impedance and a power supply that uses DC and will be tested with 36 V. The oscilloscope that will be used at the UCF lab has the responsibilities of measuring the amplitude, frequencies, PWM signals, and duty-cycle, which will output through the Arduino Uno Rev3 and 1kOhm resistor.

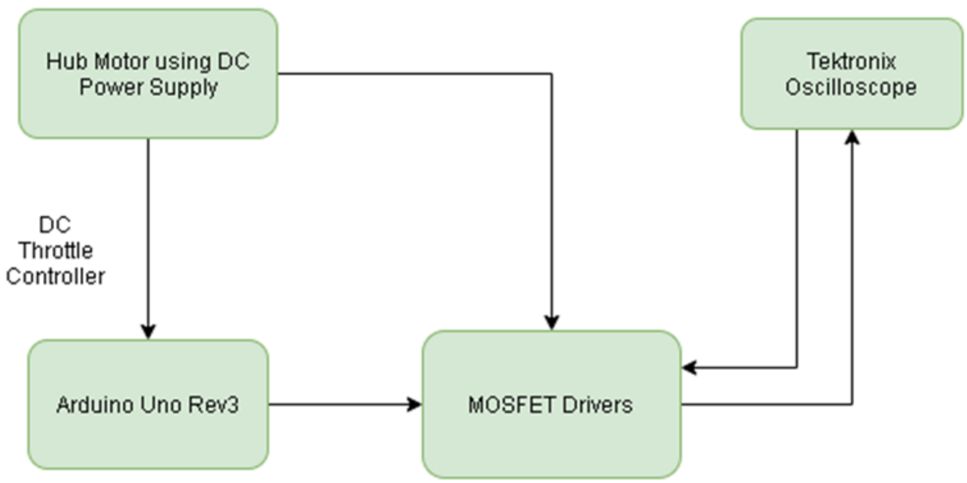


Figure 31: MOSFET Driver Testing

7.3 Software Testing

Testing the software of the components are essential to completing and measuring the system. The software system needs to be able to read the sensor signal for the electric scooter to read the surroundings of incoming objects, the ability to be able to power on from the rechargeable battery cells, the output signal is then measured with the in between each MOSFET and PWM signal. The ATMega328P needs to be able to calculate the speed to send the information and correctly display the calculated speed to the LCD Display. The frequency in which the signals are sent must match with the PWM signals to be able to display correctly on the LCD. If any errors occur within the system, then it will need to be displayed on the LCD as well. Multiple lines will be at use for different features, with the first line focusing only on MPH display and then proceed to test the clock using military time, battery percentage, and location using the longitude and latitude of the user's current location.

Objective: The objective of the software testing is to make sure that the code I created works successfully in tangent within the system.

Environment: The software testing will be done mainly at my household on my desktop (PC) using and running the Arduino IDE to edit and test the code I created for the microcontroller (Arduino Uno Rev3). The UCF lab will also be used for testing for the oscilloscope and other lab equipment.

Procedure: Testing the software portion of the EZ Scooter follow these steps:

1. Turn on the processor of the microcontroller and proceed to wait for a display of characters that should display “Welcome to EZ Scooter”.

2. Make sure the display is able to turn off with from a switch.

3. Check to see if the LCD displays the MPH on the first line and if it correctly shows that the scooter is stationary with reading “0 MPH”.

4. The software needs to be able to calculate the MPH correctly so we will need to align the sensors on the back wheel of the scooter.

5. Allow the wheels to turn 360 degrees 5 times and record the time by calculating it using the formula of MPH = (284 * circumference)/(Delta time) to see if the calculated value is the same value that is being displayed on the LCD.

6. The calculated speed and displayed speed being the same means the software is working properly. If the speed does not match within the 4% tolerance, then I must go back into the code and fix it accordingly.

<u>PWM Testing</u>

1. We will be focusing on using the oscilloscope to see what kind of PWM signal that will be created from using a PWM test function within the code.

2. Frequency of the motor should be able to take up to 8kHz and the positive duty cycle can be set to 45%.

3. Record the waveform that the oscilloscope created with a HI amplitude of 24 V for 30 microseconds and LO amplitude of 0V for 30 microseconds.

4. Keep track of the waveform to make sure it is consistently at 8kHz.

Conclusion: If the procedures work out as planned and done in a correct fashion then the scooter should be working well and acceptable for use. If any errors occur such as the display not working then fixing the code accordingly must be done urgently.

8 Administrative Content

8.1 Milestone Discussion

This section will illustrate the time and due dates from the initial stage of Senior Design 1 down to the final presentation time of Senior Design 2.

Semester	Task	Due Date
Senior Design 1		
	Idea	01/14/ 2022
	Project Selection	01/24/2022
	Research, Design, and Documentation	01/31/2022
	Initial Document Divide and Conquer 1.0	02/04/2022
	Review Document	02/11/2022
	Updated Divide and Conquer 2.0	02/18/2022
	Review Document	03/17/2022
	60 page Draft Document	03/25/2022
	Review Document	04/01/2022
	100 page Report Submission	04/08/2022
	Order Parts	04/08/2022
	Review Document	04/20/2022
	Final Document	04/26/2022
Senior Design 2		

	Assemble	May/2022
	Testing	May/2022
	Committee Selection	June/2022
	Final Presentation	June/2022

Table 12 Design Roadmap

In order to hold ourselves accountable, we created the table shown above, a schedule to plan the progression of our project. This table outlines the dates we need the completion of certain tasks. We will work to stay ahead of this schedule and if we feel that we are falling behind, we combine our efforts to stay on track. This table is divided into two sections one is Senior Design I (SD 1), and Senior Design II (SD 2).

Senior Design I was focused on the process of research, documentation and the purchasing of the parts. This semester is very important since it is the foundation of the project. Senior Design II is the focus on development and construction of the project. The hands-on semester is where we test, implement, document and troubleshoot until the project achieves its requirements.

8.2 Senior Design Lab

The senior design lab located at UCF has supplied us with all of our testing and reworking equipment. The lab provides us with breadboards, alligator clips, probes, breadboard friendly components such as resistors, capacitors, some texas instruments components as well, it also contains a triple output DC power supply, Multi-meter, Function generator, and a signal oscilloscope, and have a soldering station available for building our circuits. All these help us on better visualization ideas for the completion of the group project.

8.3 Budget and Finance Discussion

This project is fully funded by the students creating and designing all aspects of the project. The team will be splitting the total cost equally amongst each other. With a maximum budget of $800 amongst all three of us, we will look for cost effective without compromising quality parts for lower cost. Most of our project parts can be found on Amazon at a reasonable lower price. The table below shows the cost of the parts needed for the project. We understand that costs might vary when the final design is complete.

Description	Model	Cost per	Status	Remaining Cost
Scooter		$69.99	Pending	$69.99
LCD	HiLetgo HD44780 1602 LCD	$9.19	Acquired	$0.00
Microcontroller	Arduino Uno REV3	$23.80	Acquired	$0.00
Speed Controller	HGLRC Flipsky Speed Controller, Part number: FSESC V4.12	$122.99	Pending	$122.99
Throttle Control	Joystick HiLetgo Game Joystick Sensor, Part number: 3-01-0492-1	$5.79	Pending	$5.79
Rechargeable battery cells		$29.99	Pending	$29.99
Hub Motor	RBSD Electric Scooter Hub Motor	$112.89	Pending	$112.89
Headlight		$18.99	Pending	$18.99
Taillight		$16.99	Pending	$16.99
Battery charger		$12.97	Pending	$12.97
BT Module	HiLetgo HC-05	$8.59	Pending	$8.59
light sensor (x5)		$6.89	Pending	$6.89
Gyro-accel (x5)	GY-521 MPU-6050	$10.99	Pending	$10.99

Table 13 Budget Overview

8.4 Project Roles

For our design project we opted for the EZ scooter, we were named Group 14 and it consists of three members. We are a group of a diverse mix of University of Central Florida, we are students with varying degree paths. Carolina Geary is a student of Electrical Engineering, Jacob Jardines is a student of Computer Engineering, and Tony Naylor is a student of Electrical Engineering. Our project division roles are dependent on what areas we feel comfortable working on and on what our degree of pursuit is. As shown in **Figure 30** the tasks have been divided among group members based on software and hardware, however group members are expected to work together to fulfill the task.

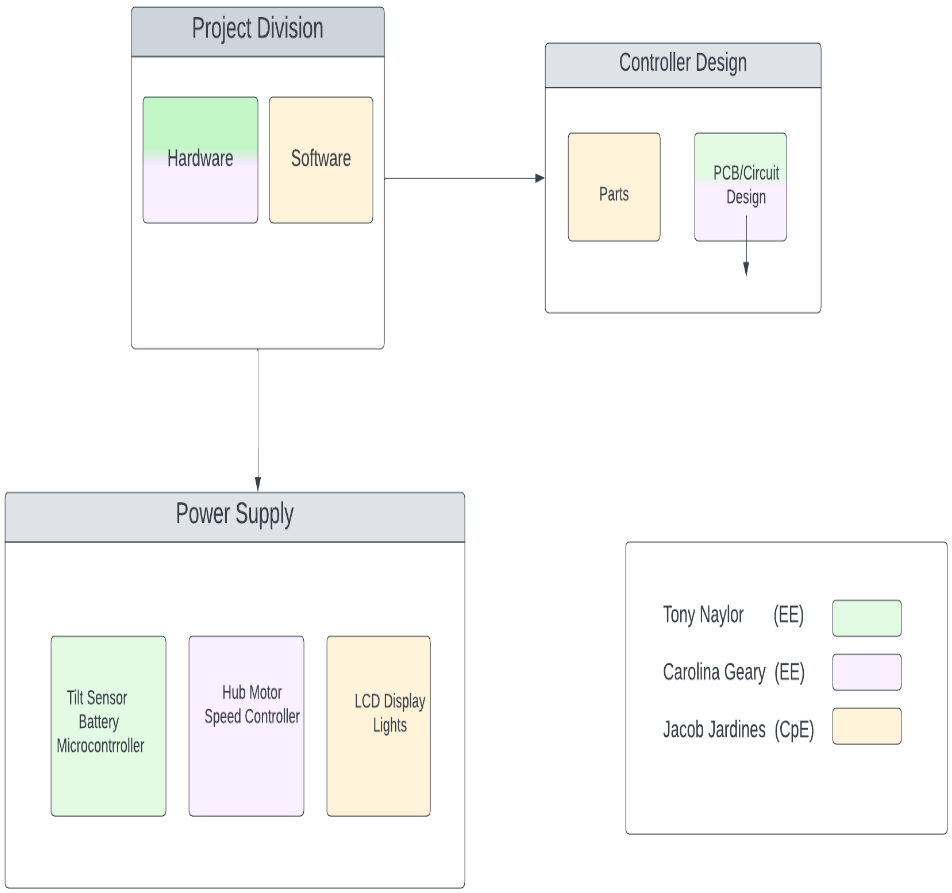


Figure 32: Block Flow Diagram

8.5 Design Deviations

During the design of our project, we deviated from our original design in a number of ways. This section will address these deviations, explaining the causes, reasoning behind these changes, and the iterations of these changes. These deviations had a variety of causes ranging from component failures to redesigns required by unforeseen challenges. Though some of these deviations were the result of challenges that forced our hands, some of these changes both reduced costs and improved the efficiency of our design.

The first deviation I will address is the mcu to motor drivers. Originally, our motor drivers were going to be facilitated by two LT1014 4-channel op-amps, with six of the outputs connected to the MOSFET gates that controlled the current path within the 3-phase motor. However, during the testing of our motor we noticed that the MOSFETS seemed to occasionally not properly close. This is because our MOSFETs have a turn on voltage of 4 volts and require 0V gate input to turn back off. While using our amplifiers we observed that a LOW output would not actually be 0V relative to our circuit's ground. This was leading to power leakage while idling the motor, heat buildup in the MOSFETs, and reductions in motor power and efficiency. It was at this point that we took another look at the driver implementation and began to rethink their use in our project. However, our microcontroller can provide a maximum voltage of 5 volts and can drive an output channel down to 0V, which means it can avoid this issue. This along with the fact that effectively zero voltage will be dropped on the other side of the MOSFETs because it will be effectively tried to ground means that we are able to drive a very high current through the motor without the use of amplifiers. To further reduce the likelihood of the Voltage applied at the MOSFET gates from locking high and to prevent this issue from persisting we also added 10k Ohm resistors between the MOSFET gates and sources. These added resistors provide a high impedance path to ground to allow the voltage output to decay to 0V if it is not being driven HIGH by the mcu.

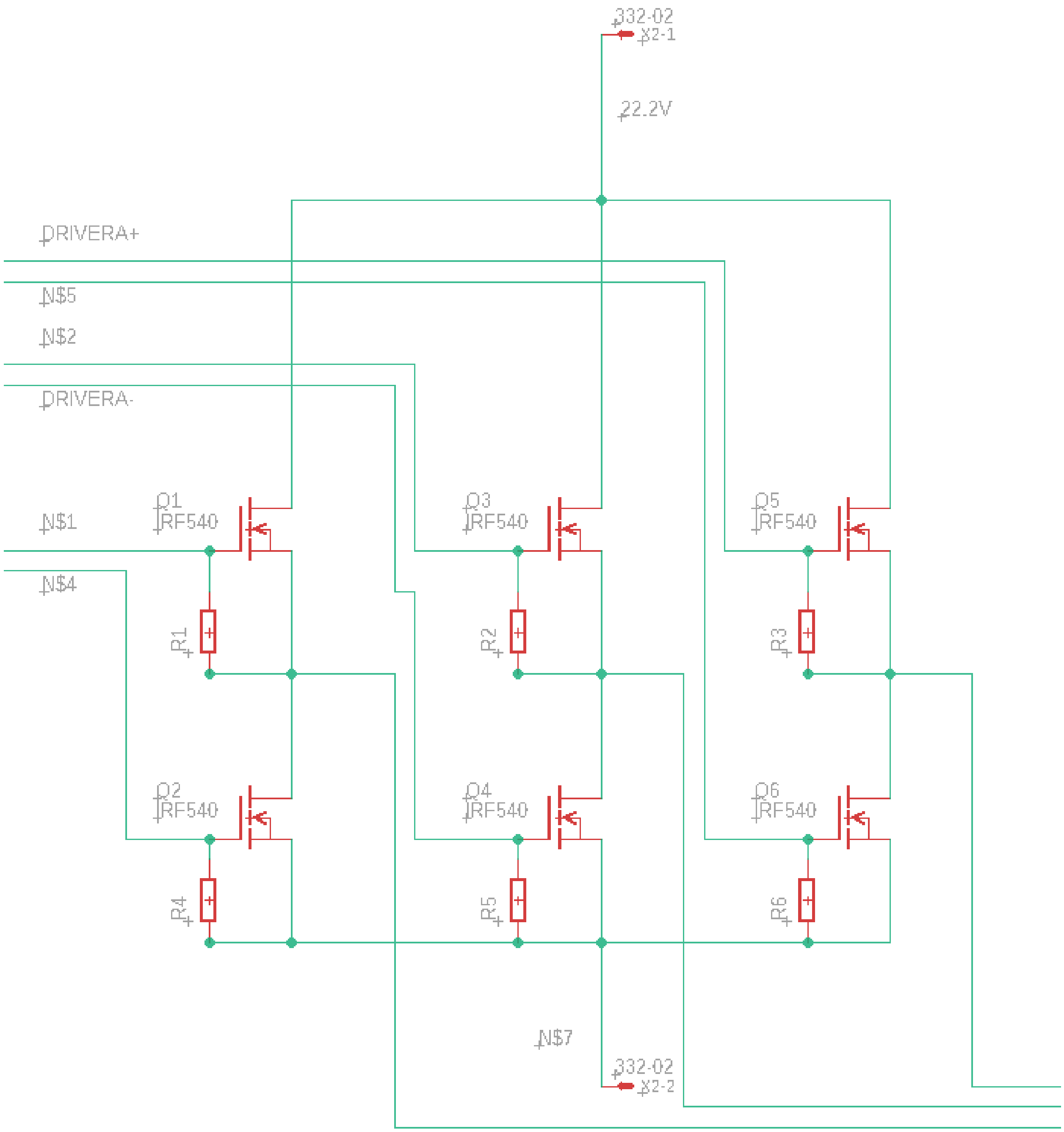


Figure 33: Redesigned Motor-Mosfet Circuit (1)

While testing the resolution of the first design resolution, a reduction in heat and leakage current, and improved motor performance, we observed improvements in all of these domains. However, the heat buildup in the middle pair of MOSFETs, pair B, had persisted, albeit less severe, despite our previous attempts. At this point we began considering drastic measures to permanently address the heat problem. Our solution wasn't an elegant one but it would be able to guarantee the elimination of the heat problem. I decided to do a redesign that would duplicate every MOSFET, placing them in parallel with each other. This would double our number of MOSFETs in our mcu to motor integration but would split the current passing through each individual MOSFET while it was in use. Because of time constraints this redesign had to be completed by hand soldering all the duplicated components on a new PCB. This took an entire day and night to complete but after finishing there was a fault in our hand soldered PCB and because of time constraints we were unable to troubleshoot the cause and repair our revamped PCB design.

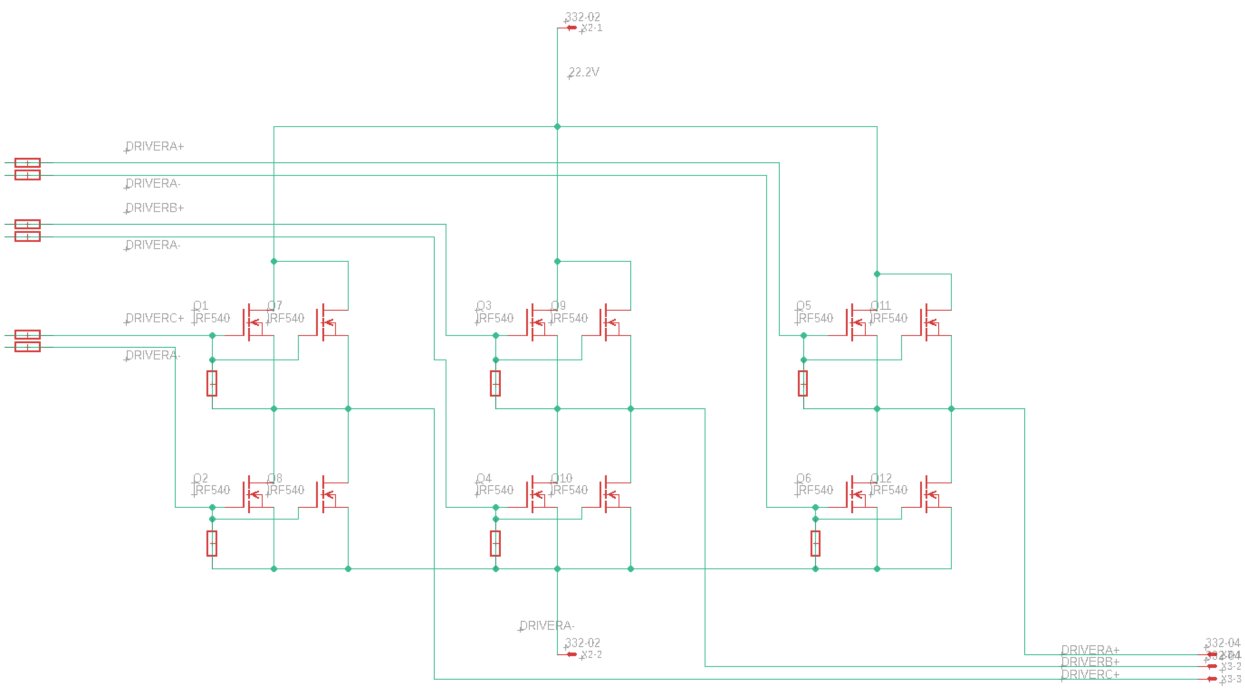


Figure 34: Redesigned Motor-Mosfet Circuit (2)

The third deviation was similar in nature to that of the first and stemmed from a problem with our MOSFET setup controlling the headlight and taillight. However, the situation with the lights differed in two main ways. Firstly, our planned means of providing necessary gate voltages were two constructed voltage doubler, using two TC428s. These voltage doublers did an even worse job of closing the MOSFET gates, with a LOW being around 5V. Secondly, the headlight and taillight offered substantial enough resistance that the current through the MOSFETs was not great enough to damage our components and voltage was being dropped on the opposite side of the MOSFETs and needed to be at least 6V for proper operation of both the headlight and taillight. However, considerable heat was still being generated and power was still being wasted while the MOSFET gates were not properly closed. To combat this, we took the same measure employed with the motor MOSFETs, where the gates and sources are linked with a 10k Ohm resistor. However, because of the voltage requirement for operation of the lights, we were unable to avoid having a voltage setup. To accomplish this we replaced the voltage doublers with one of the 4-channel op-amps that was removed from our mcu to motor integration. While they showed imperfect operation, their performance was superior to the voltage doublers. After making these alterations we observed proper functionality of our light control subsystem, with a great reduction in heat and power drain.

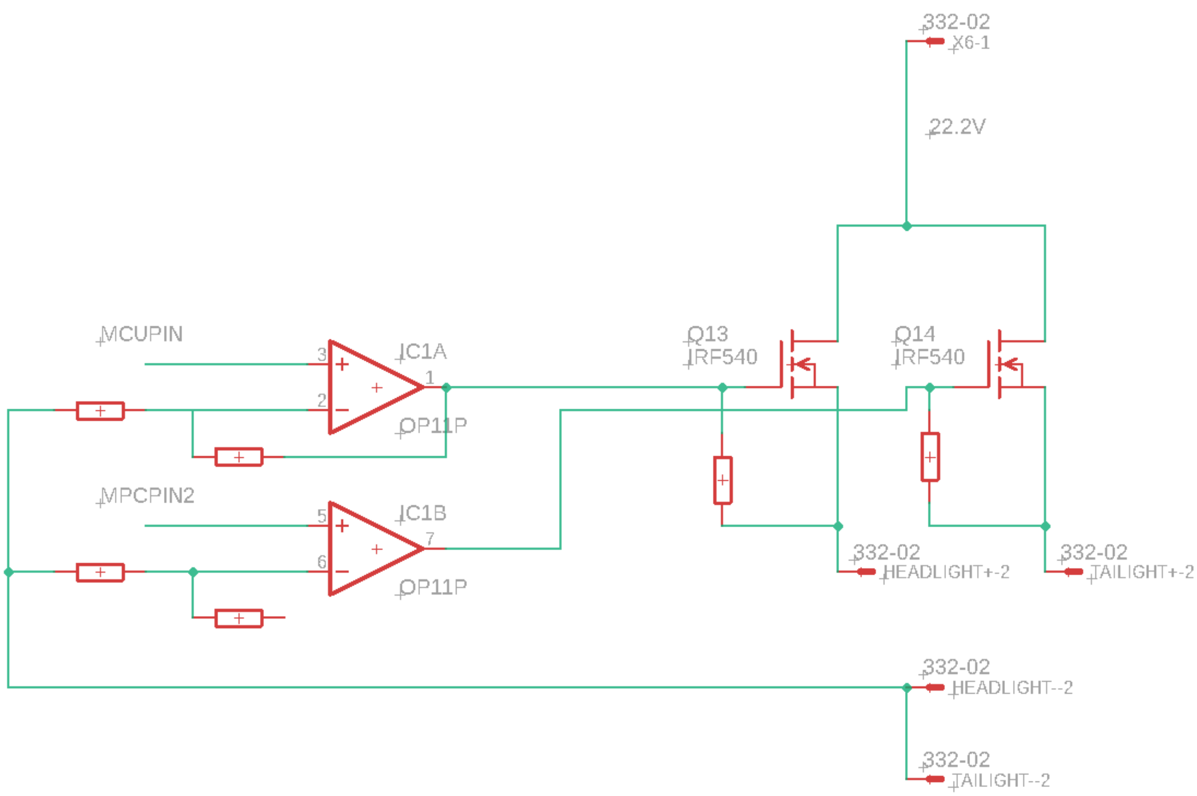


Figure 35: Redesigned Light-Mosfet Circuit

The last deviation from our design was our number of implemented mcus. While it remains possible to implement our design with the use of one mcu, we choose to increase the number to three for a number of reasons that I will address in turn. Firstly, the motor forward and reverse loops, the portions of our code that were responsible for the motor operation, had the highest priority inorder to allow for proper operation of the motor. However, we had already noticed considerable lag during the code's execution that raised serious doubts about the performance of our mcu. This made us hesitant to further divert the mcu's resources from the motor operation. Secondly, we had a scarcity of available analog input pins. Both of these issues could have been handled by implementing a scheduler into our code, checking sensors and changing port usage periodically, otherwise prioritizing the motor's operation.

Appendix A Works Cited

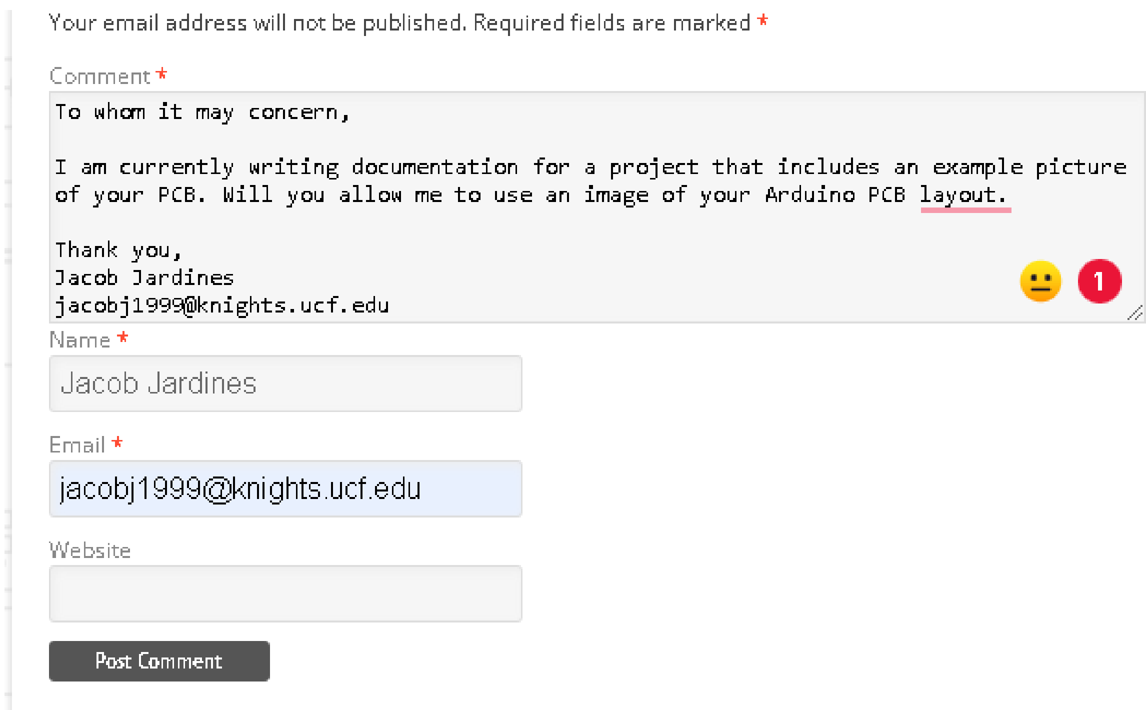
1. "The Paris Agreement." *Unfccc.int*, United Nations, https://unfccc.int/process-and-meetings/the-paris-agreement/the-paris-agreement
2. MacArthur, John, Jennifer Dill, and Mark Person. "Electric scooters in North America: Results of an online survey." *Transportation Research Record* 2468.1 (2014): 123-130., https://journals.sagepub.com/doi/abs/10.3141/2468-14
3. Apostolou, Georgia, Angèle Reinders, and Karst Geurs. "An overview of existing experiences with solar-powered e-bikes." *Energies* 11.8 (2018): 2129., https://www.mdpi.com/1996-1073/11/8/2129
4. "Motorcycle, Motor Scooter, Moped and Motorized Scooter." *Florida Department of Highway Safety and Motor Vehicles*, 21 Mar. 2022, https://www.flhsmv.gov/driver-licenses-id-cards/motorcycle-rider-education-endorsements/motorcycle-motor-scooter-moped-and-motorized-scooter/.
5. *, N. (2021, November 30). *Choosing the right electric scooter throttle: Thumb vs trigger*. Varla Scooter. Retrieved April 26, 2022, from https://varlascooter.com/blogs/knowledge/choosing-the-right-electric-scooter-throttle-thumb-vs-trigger/
6. ago, yida 2 years, Yida, posts, yidaS. author's, & posts, S. author's. (2019, November 15). *Arduino uno alternatives - which microcontroller to buy?* Latest Open Tech From Seeed. Retrieved April 26, 2022, from https://www.seeedstudio.com/blog/2019/11/15/arduino-uno-alternatives-which-microcontroller-to-buy/
7. *Amazon.com: Hiletgo Game Joystick Sensor Game Controller ...* (n.d.). Retrieved April 26, 2022, from https://www.amazon.com/HiLetgo-Controller-JoyStick-Breakout-Arduino/dp/B00P7QBGD2
8. *Build a DIY electric scooter*. Nuts and Volts Magazine. (n.d.). Retrieved April 26, 2022, from https://www.nutsvolts.com/magazine/article/build-a-diy-electric-scooter
9. George, D. (2014, June 3). *Ten best light sensors used in Robotics • Smashing Robotics*. Smashing Robotics. Retrieved April 26, 2022, from https://www.smashingrobotics.com/10-best-light-sensors-used-in-robotics/
10. *Hub Motors - Learn*. ebikes.ca. (n.d.). Retrieved April 26, 2022, from https://ebikes.ca/learn/hub-motors.html
11. *Regenerative braking of series-wound brushed DC electric motors for Electric Vehicles*. IEEE Xplore. (n.d.). Retrieved April 26, 2022, from https://ieeexplore.ieee.org/abstract/document/6360991

12. Toll, M. (2018, November 5). *The crazy big-wheeled US-made electric scooter that you can't buy... yet*. Electrek. Retrieved April 26, 2022, from https://electrek.co/2018/11/05/stator-electric-scooter/
13. | *citation machine*. Citation Machine, a Chegg service. (n.d.). Retrieved April 26, 2022, from https://www.citationmachine.net/bibliographies/ff3a16af-3615-4c23-bc01-5fb09f872685
14. Siahaan, Andysah Putera Utama, Solly Aryza, and Muhammad Dharma Tuah Putra Nasution. "Arduino Uno-based water turbidity meter using LDR and LED sensors." (2018).
15. Choron, Rachel L., and Joseph V. Sakran. "The integration of electric scooters: useful technology or public health problem?." *American journal of public health* 109.4 (2019): 555.

Appendix B Copyright

1. *Build a DIY electric scooter*. Nuts and Volts Magazine. (n.d.). Retrieved April 26, 2022, from https://www.nutsvolts.com/magazine/article/build-a-diy-electric-scooter
2. *How to use LCD 1602 display with Arduino*. Arduino Project Hub. (n.d.). Retrieved April 26, 2022, from https://create.arduino.cc/projecthub/mdraber/how-to-use-lcd-1602-display-with-arduino-b14981
3. *Hub Motors - Learn*. ebikes.ca. (n.d.). Retrieved April 26, 2022, from https://ebikes.ca/learn/hub-motors.html
4. Toll, M. (2018, November 5). *The crazy big-wheeled US-made electric scooter that you can't buy... yet*. Electrek. Retrieved April 26, 2022, from https://electrek.co/2018/11/05/stator-electric-scooter/

Appendix C Permissions



Your email address will not be published. Required fields are marked *

Comment *

To whom it may concern,

I am currently writing documentation for a project that includes an example picture of your PCB. Will you allow me to use an image of your Arduino PCB layout.

Thank you,
Jacob Jardines
jacobj1999@knights.ucf.edu

Name *

Jacob Jardines

Email *

jacobj1999@knights.ucf.edu

Website

Post Comment