Project E-Bike

Group 12

Group members

Christopher Rumpf	Computer Engineering
Victor Petrillo	Computer Engineering
Carlos Curbelo	Electrical Engineering
Alexander Gilbert	Electrical Engineering

1.0 Executive Summary

Very recently there are large pushes for people to begin switching from combustion engines to electric vehicles. Many people claim electric vehicles are better for the environment, especially for a period in time when combustion engines are taking a toll on the planet and its temperature. However, there are a number of people who don't have the money, means, or the necessity for a large vehicle to get places or to move different items around, especially in highly population-dense, urban city communities. There are plenty of places within modern society where large vehicles aren't necessary or, in some cases, are even not allowed.

Rather than using vehicles for transportation or to haul small items such as food, groceries, etc., we instead propose using electric bicycles (e-bikes). E-bikes are a modern solution to the age-old issue of transportation. While e-bikes may not be a proper solution for many people, they are still a quick and relatively cheap solution for many. The cases where e-bikes will not work is due to many countries' modern infrastructure being designed around automobiles; wide, multi-laned roads, complex highway systems, even having longer distances between cities are all factors of an automobile-driven society. In different corners of the world, however, the distance between buildings and even cities are not as far apart as they are either here in America or another automobile-driven society. Within these places, a bicycle, and even an e-bike, would make for a more convincing argument.

This report will detail the steps taken to design and plan the process to change a traditional bicycle to an electric-enabled version which, with the power of a motor and battery, will enable quicker travel. Roughly the first report will be a description of the project itself: what goals are being considered for the present and the future, requirement specifications, general overviews, diagrams, and technology investigation. The technology investigation will also include the selection for each technology that was investigated, along with the arguments for each serious consideration.

The latter half of the report delves more into how the project will be constructed, along with design constraints, executive materials involved in this project (summaries, bills of materials), part/software testing ideology, and references.

2.0 Project Description

The goal of the project is to create an e-bike conversion kit that will allow usage through an app connected via Bluetooth. The e-bike will include a cruise control that will be activated on the app along with having battery level gauge sent to your phone. This will allow easy access to the e-bike's modes and status.

2.1 Project Motivation

People need to use automobiles to go nearly everywhere nowadays, needing to go to school, work, or the grocery store. Everyone uses cars because everything is far from each other and the use of a bike either takes too long or there is not enough space. An e-bike gives more freedom to the user. Compared to electric vehicles, an e-bike is less expensive, and a better alternative means of transportation in a city. The inexpensiveness of an e-bike compared to a vehicle allows more users compared to the cost of an automobile.

2.2 Goals

Goals for this project include, but are not limited to:
Fundamentals:

- Being able to use the bike without pedaling.
- Providing enough speed to maintain balance.
    - (Hopefully not fall off from going too slow)
    - The amount depends on the environment.
- Engineering Goals:
    - App
    - Throttle control for speed
    - Cruise control
    - Ensure a constant speed that the user decides.

Stretch/advanced goals that are not currently in the active scope, but can be worked on after main goals are met:

- Building an application that is paired via Bluetooth to send and receive information to and from the bike.
    - Such information will include, but is not limited to:
        - Battery level
        - Time spent on the bike for a specific event.
        - Current speed
- Create a lighting system that would allow riders to always ride safely in the day and night.
- Providing an assist mode.
    - This mode will help you when pedaling.
        - For example, when going up a hill the force required to continue going up the hill increases.
        - When using assist mode, the motor will make up for the increase needed to go up the hill.

2.3 Requirement Specifications

These are the design specifications. We plan to follow these closely as we design and create our project. Though these specifications are subject to change at any point during the project, they will aid us in structuring the design and guide us in the right direction for success. Also, these specifications are subject to change due to any errors that are found during assembly and testing phase.

Component	Spec	Requirement	Actual
Whole bike	Bike Weight	≤28kg/~62lbs	24.75kg
Whole bike	Carry Weight	≥150kg/~331lbs	TBD
Whole bike	Dimensions	No req, just listed dimensions	1.71m x 0.67m x 1.05m
Battery	Distance/Time	30km/2hrs (approx)	48km/2hrs (approx)
Bluetooth Module	Built-in Antenna	Included	Included
Bluetooth Module	Range	100m (open space)	100m (open space)
Application	Bluetooth-capable	Yes	Yes
LED Lights	Voltage	12V	12V
LED Lights	Lifespan	A long time	50,000 hours (long enough)

Table 1: Specifications

2.3.1 - Production Cost

- We are making efforts to keep the cost of all the components to under 1000 dollars.

Item	Cost
Bicycle	Free
Power	$329
Motor	$60
Bluetooth	$8
PCB	$40
MCU	$38.3
Throttle	$15

Table 2 : Project Budget

2.3.2 – Weight

- The total weight of the project is going to be 35 lbs. or 15.88 kg.

Item	Weight
Bicycle	20 lbs or 9 kg
Motor	2.5 kg
Power	4.74 Kg or 10.45 lbs
Bluetooth Module	TBA
Brakes	TBA
Throttle	TBA

Table 3 : Project Weight

2.3.2.2 – Power

- Batteries that are about 500-watt hours can weigh up to 6 – 8 lbs. or 2.7 - 3.6 kg.
- The battery will last around 2 hours

Item	Power usage
Bicycle	None

Motor	250 – 350 Watt
PCB	TBD
Bluetooth	3 V
Throttle	None

Table 4 : Power Usage

2.3.2 – Dimensions

Tires	28x1.95
Motor	10 x 2 x 2 (in)
Power	365 x 90 x 110 mm
PCB	TBD
Throttle	TBD
Bluetooth	12.9 x 15 x 2.2 mm

Table 5 : Project Dimensions

2.3.3 – Speed

- At least walking speed (2-4 miles per hour)
- At most: electric bike Class 2 standards of 20 miles per hour.

2.3.4 - Power

- 250-350 Watts
- Rear mounted
- Rechargeable 36v Lithium Li-ion at 20ah

2.3.5 - Motor

- A brushless motor with hall sensors will be used for efficient operation.
- We are trying to have the motor carry someone of about 250 lbs.
- Hub Motor

2.3.6 - Microcontroller

- Different sensor input detections (throttle, battery)
- Output comparison component

- Control of the brushless motor

2.3.7 - Speed Controller

- Since in our design, we are using a brushless motor with sensors, we have the option of using either a sensor or sensor less speed controller.
- The use of an input signal to change the speed of the motor.

2.4 Block Diagrams

2.4.1 High Level Overview

This diagram covers the main pieces of the bike as individual components. A central control module will use electrical power that can also drive the bike motor. Peripheral inputs, such as throttle control, brake levers, and mode selector also feeds signals into the controller to determine the bike's behavior. The controller also outputs data and information to an app that can be used as a display or log that data. Each portion is given a color that is coordinated with each person that is interested in the portion but is subject to change at any point.

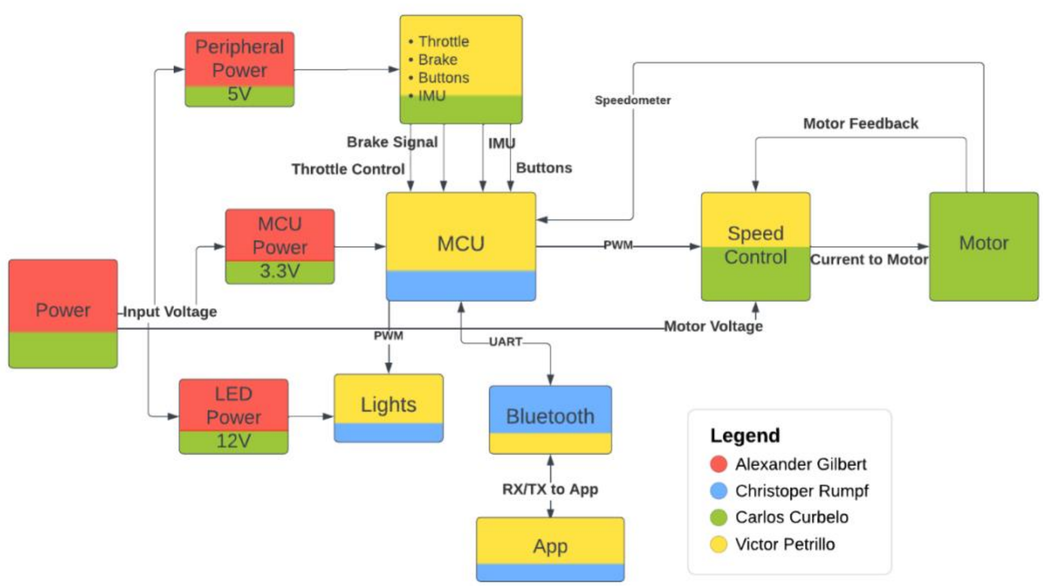

Figure 1: Hardware Diagram

2.4.2 Controller Diagram

Within the control board the main logic of the bike will be a microcontroller (MCU) that will take in all the signals and data. The MCU will be powered by the bike's main input power using a voltage regulating circuit. Motor control is done via a speed control circuit. The speed controller will use feedback from the motor to regulate speed based on the bike's signals and feedback. Any data that is logged, displayed, or input signals from the app will communicate to the MCU via a Bluetooth module.

2.4.3 Logical State and Flow

This state diagram shows the logical flow of the bike's MCU. When initially powered on the bike would be in an idle position where the bike can then be used as a bike with no electrical input the user can then switch into a powered mode using the motor. Using the throttle would put the bike in a driven state where the bike is powered by the motor regardless of the rider pedaling. An assist state is also available where the motor is used only to help the rider pedal such as making going up a hill require less work. When the brake lever is pulled, we will shut off the motor.

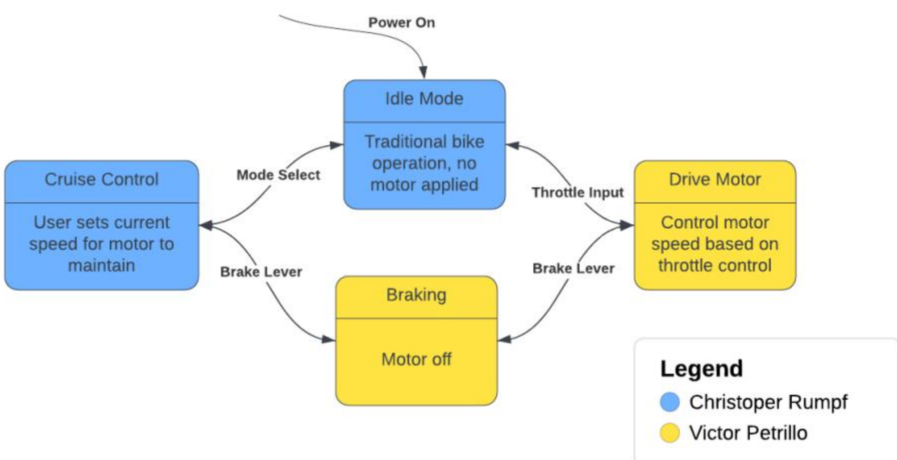

Figure 2: Control states

2.5 Project Budget

Item	Estimated Price
Motor (1 unit)	$350
PCB (3 – 5 units)	$150

Batteries (1 unit)	$350
Sensors	$50
Circuit components	$150
Forecasted Total	$620

Table 6 : Budget Breakdown

2.6 Project Milestones

These are the milestones set for our group as directed for both Senior Design 1 and Senior Design 2. Completion of each milestone is subject to change as each due date passes. Any updated dates will be changed in the updated version of the document. In addition, we will be adding our own milestones that we as a team feel like we are able to achieve.

2.6.1 Tentative Senior Design 1 Milestones

Due Date	**Milestone**
Week 1 (1/29/23-2/4/23)	Divide and Conquer
Week 2 (2/5/23-2/11/23)	Divide and Conquer Meeting
	Editing and making changes of the Divide and Conquer
Week 3 (2/12/23-2/18/23)	Divide and Conquer website update
Week 4 (2/19/23-2/25/23)	15/60 Pages
Week 5 (2/26/23-3/4/23)	30/60 Pages
Week 6 (3/5/23-3/11/23)	45/60 Pages
Week 7 (3/12/23-3/18/23)	Spring Break & 60/60 Pages
Week 8 (3/19/23-3/25/23)	60 Page Draft
Week 9 (3/26/23-4/1/23)	60 Page Feedback | 75/120 Pages
Week 10 (4/2/23-4/8/23)	PCB Designing | 90/120 Pages
Week 11 (4/9/23-4/15/23)	60 Page website update | 105/120 Pages
Week 12 (4/16/23-3/22/23)	Review, Editing, Polish BOM | 120/120
Week 13 (4/23/23-4/29/23)	120 Page Final Report
Week 14 (4/30/23-5/6/23)	In-between weeks
Week 15 (5/7/23-5/13/23)	In-between weeks

Table 7 : SD1 Milestones

2.6.2 Senior Design 2 Milestones

Dates	Milestone
Week 1 (5/14/23-5/20/23)	Build
Week 2 (5/21/23-5/27/23)	Build
Week 3 (5/28/23-6/3/23)	Build
Week 4 (6/4/23-6/10/23)	Building / Testing
Week 5 (6/11/23-6/17/23)	Testing
Week 6 (6/18/23-6/24/23)	Middle Term Demo
Week 7 (6/25/23-7/1/23)	Make Changes
Week 8 (7/2/23-7/8/23)	Conference Paper
Week 9 (7/9/23-7/15/23)	Build & Edit Paper
Week 10 (7/16/23-7/22/23)	Test & Edit Paper
Week 11 (7/23/23-7/29/23)	Final Presentation and Demo
Week 12 (7/30/23-8/5/23)	Senior Design Web Exit Interview

Table 8 : SD2 Milestones

2.7 House of Quality

The most important engineering criteria and the critical marketing requirements will be systematically laid out in the house of quality that we created. This product needs to meet several important criteria in order to be marketable. We identified six areas where we would like to meet clients after carefully examining the market of the product. Regarding how we will meet the marketing needs, engineering requirements are the key focus. To pique consumer attention, we must address six crucial components of engineering needs.

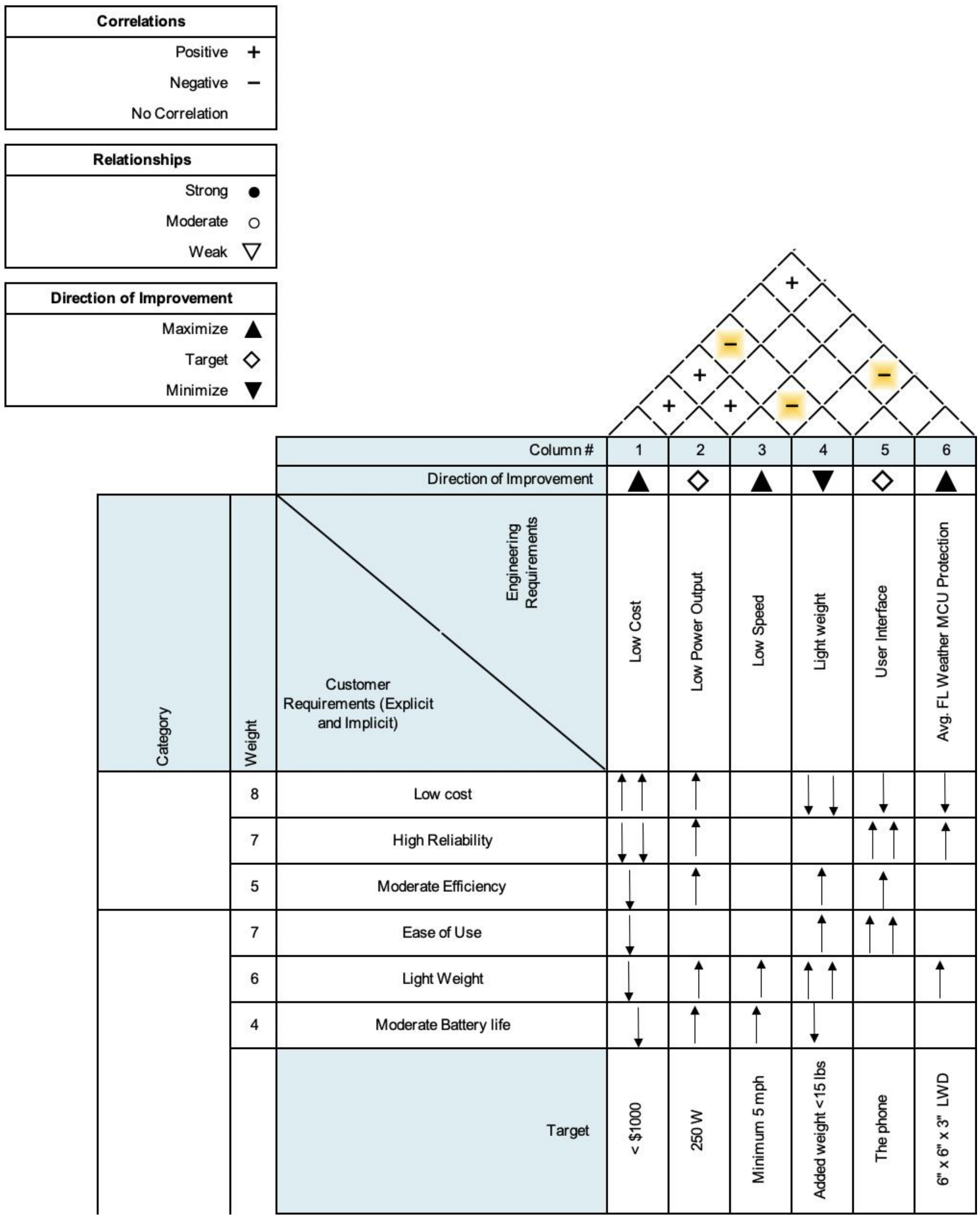

Figure 3: House of Quality

Keeping the development cost low for the group typically has a negative correlation with certain customer requirements. Having a lower power output correlates positively with most of the customer requirements because having a lower power output can result in increased longevity and efficiency, a lighter weight, and a lower cost. The speed of the bike doesn't correlate much with the customer requirements at all, other than how heavy the bike is and how much battery life is left.

3.0 Research related to project and Part Selection

In the divide and conqueror document the group did a preliminary block diagram for what we believe will be needed for the entirety of the project. An updated and complete version of a hardware block diagram will be added to this document as more information is learned and understood throughout the length of this project. The subsections below will contain all technology that was found and is relatable to the completion of this project. There will also be product investigation and with that a pro and con chart for the different product available that may be used in the future.

3.1 Batteries

There are primary and secondary batteries. The former is a single use while the latter is the more common type we see where the battery is rechargeable. The reason for the differences is that the electrode materials are made in a way that is not reversable while the electrodes in a secondary battery are reversed when applying an electric current. In the case of this project we will need to use a secondary type so the battery on the e-bike will be able to be reused.

Batteries transfer charge by using a process called oxidation and reduction. This was first discovered by an Italian named Alessandro Volta using copper and zinc separated by a saltwater solution. Oxidation is when a substance loses electrons (the zinc from the previous example) and transfers over to the substance gaining electrons which is called reduction (the copper). This process is one time use for primary batteries where the oxidation and reduction process is not reversable. In secondary batteries, however, it is reversable but only for a certain number of charge cycles. This is accomplished by transferring the electrons back to the metal so the oxidation process can occur another time. Every time this process occurs there may be irregularities in the metal which keep the metal from oxidizing correctly. This may reduce the max charge or even kill the battery outright and no longer be able to hold a charge. An anion is a negatively charged ion.

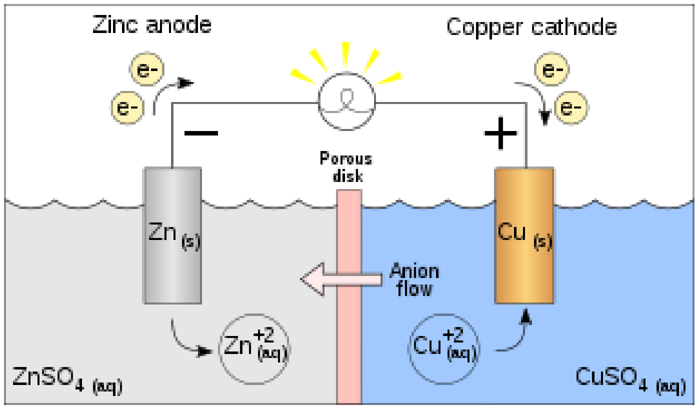

Figure 4: Galvanic cell flows by OhioStandard and AntiCompositeNumber

Batteries are affected by a few different characteristics. One is temperature. There are a few other characteristics such as the material used for a battery. This is primarily a concern for lithium-ion batteries which have a few different chemical makeups as opposed to the other secondary batteries that will be spoken of in 3.1.1 Secondary battery types – chemistry.

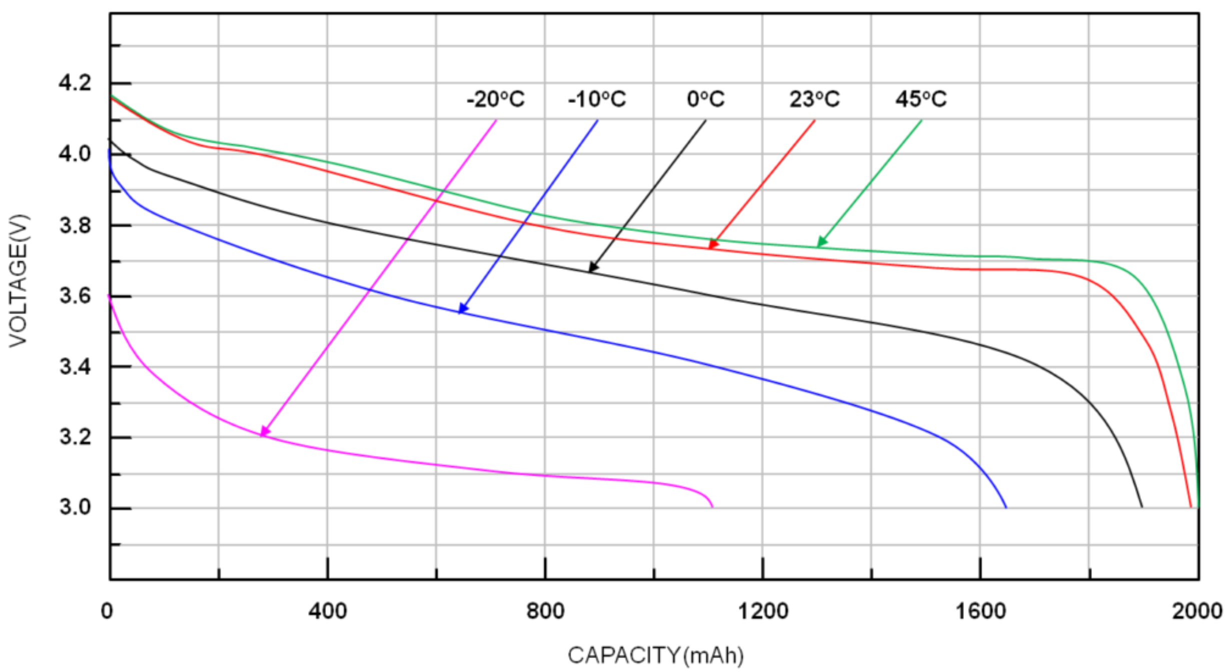

Figure 5: Voltage output at temperature variances (copyright pending)

As can be seen above the voltage output of a battery decreases as temperature decreases. This means there is a nominal temperature for a battery to operate at, however, seeing as we are in Florida and the specifications for the project call for normal Florida weather there is no need to consider any large changes in output voltage.

Size has the main effect of changing the capacity of a battery (measured in amp hours Ah). This is a measure of how long a battery can last for one hour when pulling at that rate. An example of this happening can be seen in the figure under 3.1.1.2 Nickel-Metal Hydride batteries.

3.1.1 Secondary Battery Types - Chemistry

A basic visual for a secondary battery can be seen in the previous figure above in section 3.1 Batteries. There are many different chemistries for secondary batteries, however, there are specific chemistry types that are more well known than others. There are three main battery chemistries that are used today, one being lithium-ion, these are found in portable devices such as phones, and laptops. The second is a lead-acid battery, these are found in combustion engine vehicles, more recently, however, in electric vehicles the battery chemistry is lithium-ion. The third type of battery is Nickel-Metal Hydride (NiMH), which is found in rechargeable batteries from companies such as Energizer.

3.1.1.1 Lithium-Ion Batteries

Having a battery being Lithium-Ion is very broad because the electrodes have a few different options in regard to technology with different benefits and main applications. The anode is fairly consistent using some form of carbon. The cathodes are made with a metal oxide that always includes lithium. The most popular chemistries contain cobalt because of the stability of the element when reducing. However, cobalt is an expensive element along with nickel (which can be seen in other chemistries) so other alternatives are being looked at actively as a replacement. For example, a cathode of lithium iron phosphate (LFP; $LiCoO_2$) was discovered by Arumugam Manthiram and John B Goodenough and could be used as a cathode.

These batteries are considered to be the best secondary battery available because of a few factors. There are three main reasons to this. The first is the higher specific energy (other than LFP) of the battery being around 460 kJ/kg while the next most common would be the NiMH battery being at 360 kJ/kg. The Li-ion battery also has low rate of self-discharge from 0.35% to 2.5% depending on state of charge with an output voltage of approximately 3.6 volts. There are safety concerns for this battery that will be further explored in section 4.1.

3.1.1.2 Nickel-Metal Hydride Batteries

This chemistry is most often found in single celled rechargeable batteries used in video game controllers or places where a AA and are used as a replacement for alkaline batteries because alkaline batteries are nominal at 1.6 volts while NiMH operate nominally at 1.2 volts. The discharge rate is dependent on if the NiMH battery is a low self-discharge version at a cost of approximately 25% capacity. The discharge is 13.9% to 70.6% while the slow discharge rate is 0.08% to 2.9% which is comparable to Li-ion but at a lower energy density. Below is a discharge curve for an Energizer™ NH15-2300 rechargeable battery.

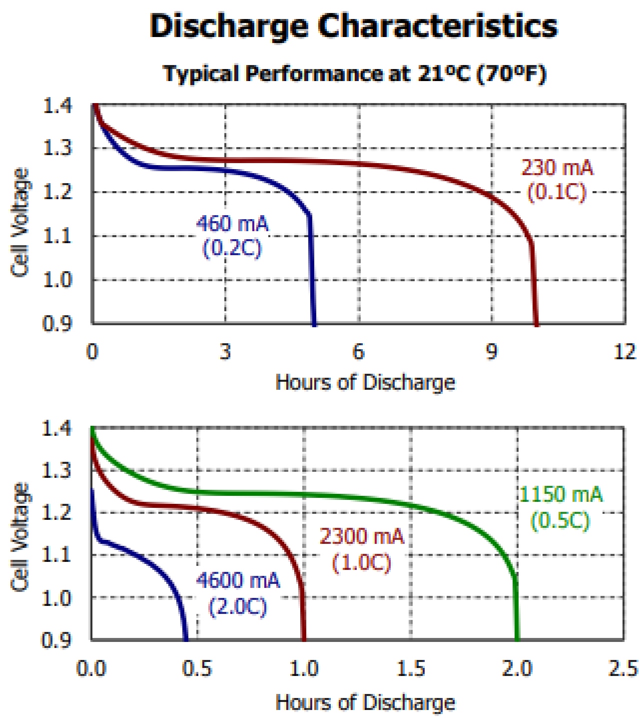

Figure 6: NiMH discharge characteristics (copyright pending)

As can be seen in the above figure, cell voltage drops dramatically in the first 10 minutes of discharge but then depending on rate of discharge can be fairly consistent until a point in time when the battery no longer is capable of oxidizing and then the cell voltage immediately drops to 0.9 volts.

3.1.1.3 Lead-Acid Batteries

Normally used in vehicle operations because of the high surge current needed for start up a motor. However, these have a high discharge rate and few battery discharge cycles comparatively to lithium ion and NiMH. They are also very cheap comparatively so they can be used on other applications as well.

The chemistry consists of having a lead plate on both sides with a sulfuric acid solution in-between. The acid solution changes depending on if the battery is charged or discharges fully. Below is the figure for a fully discharged plate.

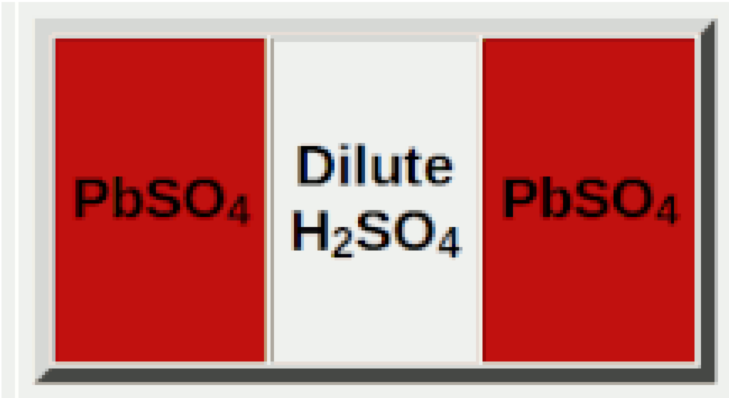

Figure 7: Fully discharges lead-acid battery (copyright pending)

As seen in the figure above the two plates are the same being lead sulfate and the electrolyte becomes primarily water because the electrolyte becomes dilutes while discharging. Now will be the fully charged state of a lead-acid battery.

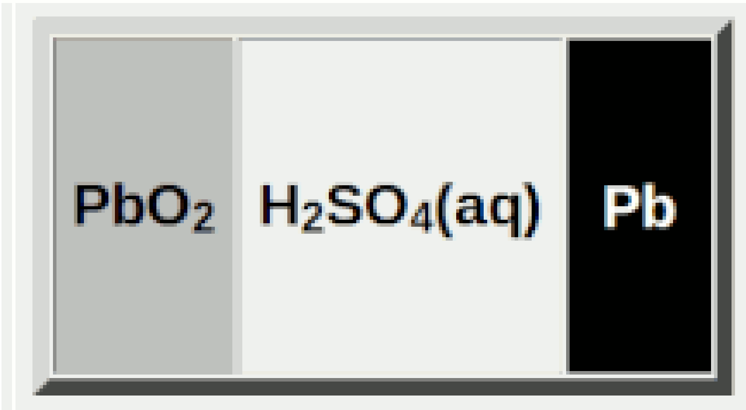

Figure 8: Fully charged lead-acid battery (copyright pending)

In this case the negative plate is the lead while the positive is the lead dioxide and these plates are separated by a concentrated sulfuric acid.

3.1.1 Battery selection

Currently most options consist of Lithium-Ion batteries because of their low discharge rate, higher voltage output, and high energy density. Therefore, the market is saturated with lithium-ion batteries and the only other option would be a set of nickel-metal hydride cells in series to get the needed voltage and has a higher rate of discharge compared to a Lead-acid which has fewer charge cycles which would not work for this application. This would require more work than is in the scope of this project so will not be done and will instead use the readily available lithium-ion batteries. Below will be a comparison of the two battery chemistry types that will be used and the other option that could be used if there was a more market availability.

Seeing as the motor (in 3.6) has a power maximum of 700 watts, but is expected to be 250-350 watts, a 36-volt battery with a limit of 9.5 amps would produce 342 watts to attempt to stay under the max nominal power. If the battery outputs at 24 volts we would keep the amperage at around 14 amps for a wattage of 336 which would be below the 350 watts. We limit the amps to ensure the e-bike does not go above the speed that would make the e-bike a class three.

	Lithium Ion	**NiMH**
Market availability	High	Zero
Discharge rate	Very low	Has a range from very low to low
Voltage	Around 3.6	Around 1.2
Cost	Very high	Medium
Energy density	Very high	Medium

Table 9: Battery Options Comparison

From the table of differences above a lithium-ion battery pack is the only choice though would be the best choice given how this battery will be used. It will need to be reliable with a low discharge rate while idle and not in use. The energy density will also help with any distance issues that may have arisen if the NiMH chemistry had been used.

The battery we could use is the *36v 20Ah Unit Pack Power e-bike battery*. This would give us the voltage needed to power the motor and would be able to run the motor continuously for around two hours assuming we limit the amperage to the 9.5 amps that was calculated earlier. The battery will allow the motor to get the necessary power to not only run for two hours but will also have the capability to allow the motor to go up a slight incline while maintaining a higher speed.

Figure 9: Battery pack chosen (copyright pending)

The max current discharge for this item is 25A which is higher than the current we plan to take from the battery pack and also give us room to also ensure that the other components can be powered. The other components include the microcontroller which will allow us to run the Bluetooth module for the app.

The other option is much cheaper but if something breaks with the item it has a much longer return time of around two months. It is essentially the same thing. This could be positive because the two main constraints are time and money. In this case, however, the time is too high to be able to say that the lower cost is worth it, especially if there is a problem with the cheaper battery.

3.2 Microcontroller (MCU)

A microcontroller (MCU) is a small computer that is built into an integrated circuit (IC). MCUs are used in many applications from appliances, robotics, vehicles, and more. An MCU will contain a central processing unit (CPU), random-access memory (RAM), read-only memory (ROM), and inputs and outputs (I/O) all on a single IC. Not to be confused with a microprocessor, which is a single IC that contains a CPU with none of the peripherals a MCU will contain: RAM, ROM, or I/O. MCUs can be designed for large applications with high CPU word sizes, 64-bit or 32-bit, and high clock speeds, small 4-bit or 8-bit sizes, and low clock speeds; or any combination in-between. They also have many different built in serial interfaces such as: recommended standard 232 (RS-232), inter-integrated circuit (I2C), serial peripheral interface (SPI), and controller area network (CAN). These serial interfaces can allow a MCU to communicate with other MCUs or sensors that support those protocols. Along with serial interfaces there are many peripherals that can be added: analog-to-digital converters (ADC), digital-to-analog converters (DAC), timers, pulse-width modulation (PWM) generators, and digital signal processing (DSP). Taking all these MCU design options into consideration the e-bike will use a single MCU with enough I/O and peripherals to do the necessary control and communication for the project.

3.2.1 Microcontroller Investigation

These next sections will cover an investigation of different manufacturers of MCUs and their products. Also, the MCU programming environments for development and debugging will be explored. The MCUs in consideration are not meant to be a complete market comparison but just a small subset of what is known, available, and easy to use.

3.2.1.1 Texas Instruments - Mixed Signal Processing (MSP) 430

Texas Instruments (TI) makes multiple models of MCUs but the one focused on here is the MSP430. The MSP430 is a 16-bit MCU that focuses on low power usage. The top speed is 25 MHz and has multiple peripherals such as PWM modules, I2C, SPI, and ADCs. Programming the MSP430 is done in the C/C++ or assembly language using TI's Code Composer Studio (CCS).

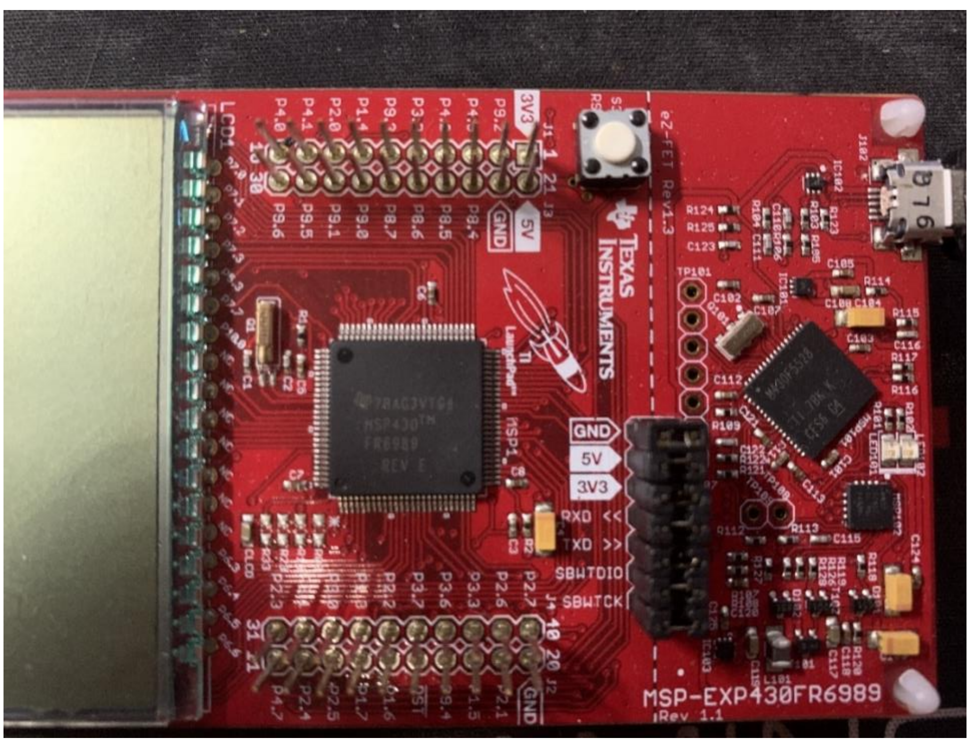

Figure 10: MSP430

3.2.1.2 Microchip - Peripheral Interface Controller (PIC)

Microchip makes 8-bit, 16-bit, 32-bit, and dsPIC MCUs. Some of the higher end MCUs can reach speeds of 120 MHz or have 49 to 78 I/O pins. Certain models have built-in peripherals such as integrated Motor Control PWM and a Motor Encoder Interface which would help directly with the motor control portion of the project. PIC microcontrollers can be programmed using Microchip's PICkit which is a hardware serial tool for programming and debugging. Microchip also provides an integrated development environment (IDE) called MPLAB X as a tool for writing code in C.

3.2.1.3 Atmel – AVR

Atmel AVR is a family of microcontrollers designed and manufactured by Atmel Corporation, now owned by Microchip. The AVR microcontrollers are usually 8-bit. AVR microcontrollers are typically programmed using C and Atmel Studio (IDE). The AVR microcontroller is also the microcontroller of choice for the Arduino platform. Arduino is both the hardware and software meant to make building and programming microcontrollers easily accessible. Programming in Arduino is done using the Arduino Programming Language which is like C and C++.

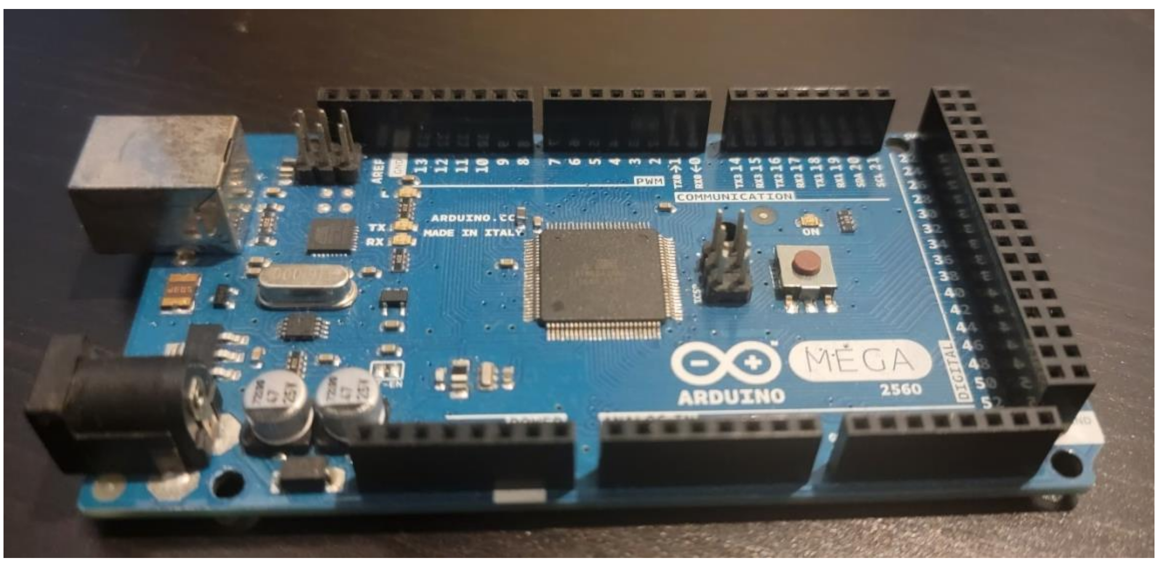

Figure 11: Arduino

3.2.1.4 Arm – Cortex-M

The company Arm designs and licenses a family of microcontrollers they also call ARM. The acronym used to stand for Advanced RISC Machine, with RISC being reduced instruction set computer. Arm does not manufacture any of their designs and instead licenses the architecture to other companies to build and add any peripherals. The main groups of Arm microcontrollers are Cortex M, R, and A. The groups have different applications with Cortex-A being application processes and would be used for operating systems like on Android or IOS phones and more. Cortex-R are for real-time applications and high performance, most are used in low level systems like hard drive controllers and airbags in vehicles. Finally, Cortex-M processors are more general purpose but still can be high performance and low power.

The focus for this project will be on the Cortex-M series. Because of the nature of the way Arm microcontrollers are manufactured it can be difficult to narrow down one way to program and use. There is one way to program Arm microcontrollers using Joint Test Action Group (JTAG) standard. JTAG is an industry standard for programming and connector layout. Also, Arm can be used with Arduino, Java, C, and many other languages.

3.2.2 Microcontroller Firmware

The code to be written to the MCU is called firmware, and there are two main structures: super-loop and real-time operating system (RTOS). The super-loop structure is very simple, after initial setup an infinite loop will loop through all of the tasks and functions the MCU needs to perform. This can be good for a few simple tasks but some steps in the loop can take longer than others causing slower than expected performance. On the other hand, RTOS has a scheduler that can allow tasks to be performed almost concurrently. Each task can be given a specific time to run and the scheduler will switch between them and make sure one does not block other tasks from running. For our e-bike application we will have a few main tasks always running: Bluetooth communication, polling sensor data, motor control, and other miscellaneous tasks such as lights and brakes. Because of this we will try to use an MCU that can support an RTOS due to all the functions we need to run. All of the MCUs previously discussed can support an RTOS with some being better than others.

3.2.3 Microcontroller Selection

There are many kinds of microcontrollers for different cases with diverse programming options. The expected MCU pins and peripheral requirements are listed in the table below. Speed will also need to be taken into consideration since controlling the motor and sending different communication protocols simultaneously could cause slowdown and unexpected behavior.

Function	Number of pins	Protocol or Peripheral
Motor driver	6	PWM
Motor sense/feedback	3	Interrupts
Bluetooth module	2	Serial (RS-232)
Throttle	1	Analog/Digital
Brakes (Front and Rear)	2	Analog/Digital
Battery monitor	1	Analog
IMU	2	I2C
Lights	5	GPIO

Table 10 : MCU Requirements

Based on the investigation the list of possible MCUs are shown in the table below. These were selected as options based on availability, programming interface, and use case.

MCU company	MCU	Programming Interface	Programming Language
TI	MSP 430	JTAG	C
Microchip	dsPIC	PICkit	C
Atmel	AVR	JTAG / Arduino	C / Arduino
ARM	Cortex-M	JTAG	C

Table 11 : MCU possibilities

STMicroelectronics makes a family of ARM MCUs called STM32 which look like a good choice and are available. The F4 series of STM32s is ARM Cortex-M4 with high speed and many peripherals that can work for our project, specifically an advanced timer feature that can output multiple PWM signals for motor control. The smallest STM32F4 comes in a low-profile quad flat package (LQFP) with 64 pins, specifically called *STM32F405RGT6* this will be our MCU for the e-bike. Also, Microchip's *dsPIC33EV* is a good choice since it has built in support for BLDC motors and multiple PWM channels. At the time of this writing some dsPICs are difficult to find in stock and will mainly be used as an alternative the STM32.

3.3 Spatial Sensing

As an added safety feature the bike will be equipped with a sensor to determine its position and orientation in space. Since the bike controller is all being designed and built, some sensing unit will be needed. This can be done in a multitude of ways from vision implementations to proximity detection. Using vision would require a processor with large computing power and unrealistic for the goal of this project. On the other hand, using a proximity detector can be useful but has poor tolerances and might require multiple sensors to detect all necessary angles. The goal will be to have an added sensor that could detect impacts, orientation and potentially road gradients, and other operational data.

3.3.1 Accelerometer

An accelerometer is a sensor that measures an object's acceleration. Accelerometers can be made in different ways from using a small, microscopic mass that moves inside the sensor called a micro-electromechanical system (MEMS). Another way is variable capacitive (VC) where a mass is between two plates and the change in capacitance is related to its acceleration. Also, piezoelectric which uses a piece of piezoelectric material to convert motion into an electrical signal. Depending on configuration, sensors can be setup for 1, 2, or 3 axes in space. Because of advances in manufacturing and parts becoming smaller, and more space effective, 3 axes sensors are common. The use of an accelerometer on the bike would be to sense sudden changes in acceleration or large spikes which would indicate an impact. In the event of an impact the bike could shut down or turn the motor off to prevent further damage.

3.3.2 Gyroscope

A gyroscope measures an object's angular velocity, which is its rotation about an axis. Like accelerometers, gyroscopes can also be MEMS or piezoelectric. The 3 axes in relation to an object's center of gravity have a name: pitch, roll, and yaw. Although these names are usually associated with aircraft the same names and applications can be applied to the bike. For use on the bike the absolute angle of the bike can be detected to see if the bike is falling over or is flipped. Depending on if it is possible, the gradient angle of the road the bike is on could possibly be detected which can be some helpful data to the rider.

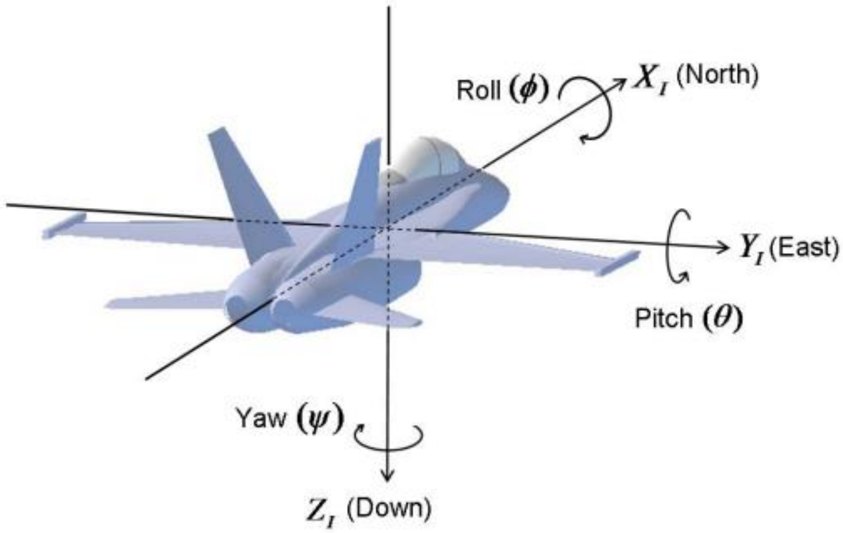

Figure 12: Pitch, Roll, and Yaw example (copyright pending)

3.3.3 Magnetometer

A magnetometer senses magnetic fields; a compass is a common application. Magnetometers have many applications in navigation, environment monitoring, and even medical devices. Our purpose in using a magnetometer would be for heading data. Giving the rider a readout of which cardinal direction they are moving is useful in urban and rural environments.

3.3.4 Inertial Measurement Unit (IMU)

All the above sensors can be packed into one sensor called an inertial measurement unit (IMU). The IMU can be used to measure a bodies' orientation and motion in space. Although an IMU is one unit it is usually made of individual sensors: accelerometer, gyroscope, and magnetometer. Not all are required; an accelerometer and gyroscope are used with a magnetometer being optional. We can use an IMU to sense all our criteria such as sudden acceleration spikes, angular data in case the bike were to fall, and use the magnetometer as a compass.

3.3.5 IMU Selection

There are many IMUs on the market. A 9 degree of freedom (DOF) IMU with 3 axes accelerometer, gyroscope, and magnetometer are available and widespread on the market. Another thing to consider is converting the data from the IMU into useable information. The process to do this is outside the scope of this project. Fortunately, there exist IMUs with built-in microcontrollers that can do the processing on-chip and offload that to the IMU instead of the main MCU. The main MCU can then get relevant data from the IMU via interrupts or polling. Our other option would be to use an IMU that outputs raw data where we can poll for large changes or spikes in values.

IMU	Features
BNO055	FusionLib software for absolute position
BNO085	SH-2 firmware with MotionEdge
ICM-20948	Digital Motion Processor
MPU-9250	Digital Motion Processor
AltIMU-10	Raw data from sensors

Table 12: Different IMU Chips (copyright pending)

Some of the listed IMUs are reaching end-of-life (EOL) and are no longer in production. Others using the integrated Digital Motion Processor (DMP) have difficult documentation. Originally, we wanted to use the *BNO085* which is built on the same hardware as the BNO055 but with an updated firmware for better performance. This became problematic since the BNO085 uses SH-2 for it’s communication protocol which was complex to decipher. In the end we chose to use the *AltIMU-10* for its simplicity and because we only needed basic data.

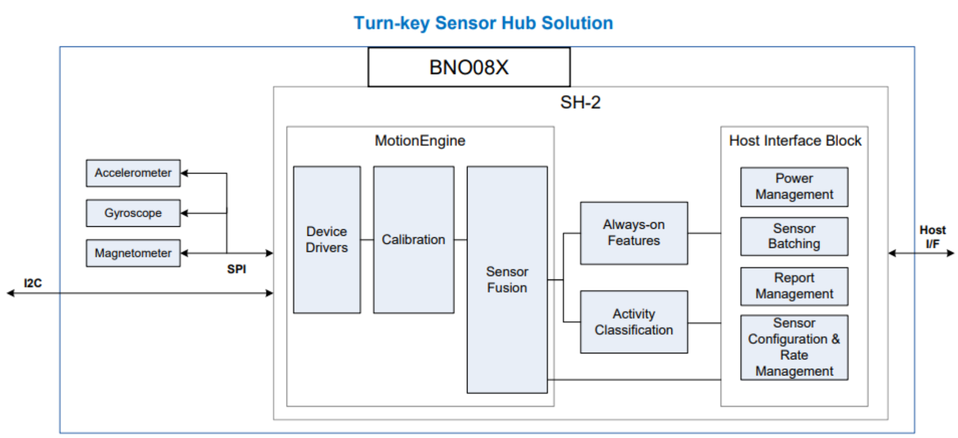

Figure 13: BNOO85 block diagram (copyright pending)

3.4 Bluetooth Modules

For this section we are looking into Bluetooth modules. Bluetooth is a technology that uses radio waves to transmit high-data information across short distances, as opposed to the small-data, long-distance nature of most radio waves used today.

We will discuss some of the features that we deemed necessary for our design specifications. The group is planning on utilizing Bluetooth technology by integrating a Bluetooth module into our main PCB, and then engineering it to transmit and receive information from a connected smartphone running our custom smartphone application.

3.4.1 - General Information

Bluetooth modules are exactly what the name may suggest: finished hardware components that, when integrated with other technology, can enable the use of Bluetooth within whatever the module is fitted into. Rather than having to come up with a way to implement this technology by making our own circuit, Bluetooth modules simplify this process. The group's goal is to achieve a Bluetooth connection with a smartphone and its respective application; *how* this goal is achieved is not being considered.

According to mokoblue.com, Bluetooth modules have a maximum communication range of around 100 meters (about the length of a football field). This range classifies Bluetooth connections as short-range communications. This is an ideal specification for the group's application, since a communication technology is required that can send high amounts of data over a short distance; it is assumed that the connected smartphone would be carried by the operator of the e-bike.

When selecting a Bluetooth module to use for our project, there are multiple things to consider. Included in these things are the Bluetooth module's application and transmission distance, which were already considered previously; both the application and transmission distance are compatible with almost all Bluetooth modules. Among other things to consider are power consumption, technical documentation/support, and, since we live in the wonderfully moist state of Florida, we must also consider the moisture sensitivity of these modules.

Figure 14: A Bluetooth module (copyright pending)

3.4.2 - Power Consumption/Output

Power consumption and power output are both incredibly important specifications to consider when selecting a Bluetooth module. Having the incorrect power consumption information could mean that your hardware will fail; parts could not function properly because they're not getting enough power, or they can become damaged after receiving too much power.

The group is considering both specifications for the project. When considering power consumption, having a smaller power consumption will be ideal. The e-bike is just that: an electric bicycle. It is dependent on the electricity that can be provided by the battery we select. Having components that use less power where they are able will be invaluable to the longevity of the e-bike, both in the short term and long.

The output (transmit) power of the module is measured in dBm (decibel meters). The units scale similarly to decibels, with increases in the units being exponential rather than linear. The modules listed after performing a quick search on Mouser Electronics reveal that modules commonly operate within the +10 to +20 dBm range. In our case, as previously mentioned, the range will not be very far; the operator of the e-bike will be within typical bicycle operating distance (which means either sitting on or driving it in some capacity). Opting for the lowest transmit power that achieves the functionality we require would be ideal; so far that seems to be the +10 dBm models.

3.4.3 – Communication Connection

When selecting a Bluetooth module to use, something to consider is how the module is going to communicate with other technologies. There are two main options for this: an internal, built-in antenna or using a radio frequency (RF) pin on the module. The RF pin can be used to pair with coaxial technology (like a coaxial TV cable) to communicate with an external antenna. The group has opted to find a Bluetooth module with a built-in antenna to not only avoid having to purchase a separate antenna, but also to make that part of the building/testing phase simpler.

3.4.4 Bluetooth Selection

When performing a search on Mouser Electronics, many different models of Bluetooth modules come up. As mentioned under the Bluetooth section (3.4) of the technology investigation, the group has opted to use a module with a lower transmit power (+10 dBm) and one with a built-in antenna. In addition, it is also compatible with the USART serial interface and meets the criteria for our design.

Specifically, the group has chosen the *DSD TECH HM-10 Bluetooth 4.0 BLE iBeacon UART Module with 4PIN*.

Specifications	**BGM220PC22HNA2**	**DSD TECH HM-10**
Protocol Stack	Bluetooth Low Energy 5.2 Bluetooth Mesh	Bluetooth Low Energy 4.0
Max TX Power	8 dBm	-59
Security	Vault Mid	Unsure what to put here
Antenna	Built-in	Built-in
Flash (kB)	512	64e6
RAM (kB)	32	32
GPIO	24	0
Temp Range	-40 to 105 Celsius	-20 to 95 c
Carrier	Reel	Plastic case

Table 13 : Bluetooth Module Comparison

There are a few factors to consider when using a self-contained Bluetooth module like the one selected *Silicon Labs 802.15.1 BGM220PC22HNA2*. Most modules are independent MCUs which require their own base firmware to operate. Factory modules will come blank with no initial firmware which can be confusing since development boards have firmware loaded onto them. During development we may find that a development board works, but an off the shelf Bluetooth module of the same part will not work. In case this happens we have also chosen a backup module; an off-the-shelf pre-built Bluetooth module, the *HM-10*. This module uses Bluetooth 4.0 which supports Bluetooth low energy and has only 4 pins: 3.3V VCC, GND, TX, and RX. This is designed to be a very simple UART connection between Bluetooth devices. The initial baud rate is 9600 but using attention (AT) commands settings can be changed. Once paired with a smartphone the module will transmit and receive data appropriately.

Figure 15-HM-10 Bluetooth Module

3.5 Voltage Regulator

For this part of the text, we are conducting research on the different types of voltage regulators. Looking at two types of voltage regulators and seeing how they work. Also discovering the purposes of each and why they would benefit our needs.

3.5.1 Linear Voltage Regulator with Low Dropout

Without the need of an inductor, linear regulators are a straightforward way to use an integrated circuit to regulate a higher voltage to a lower voltage. Due to their simplicity and low price, linear regulators can be widely used in today's electronics and have a wide range of specialized applications. The dropout voltage is the smallest voltage that must exist between the input and desired output voltage for the linear regulator to function properly. The linear regulator enters dropout mode, where it stops regulating the input voltage and the output voltage tracks the input voltage, if the voltage difference between the input and output is less than the specified dropout voltage.

The benefit of using linear regulators with lower dropout voltage is that they can regulate input voltages more closely to the desired output voltage. In other words, the closer the input voltage can be to the output voltage while preserving regulation, the lower the specified dropout voltage. LDO linear regulators regulate the input voltage from a source, such as a battery, into a lower output voltage that may be used by a device on the load side, typically a microcontroller. Voltage regulation is achieved by the two main LDO architectures, PMOS and NMOS. LDOs achieve voltage regulation regardless of architecture thanks to a feedback loop that enables the circuit to regulate the drain-to-source resistance.

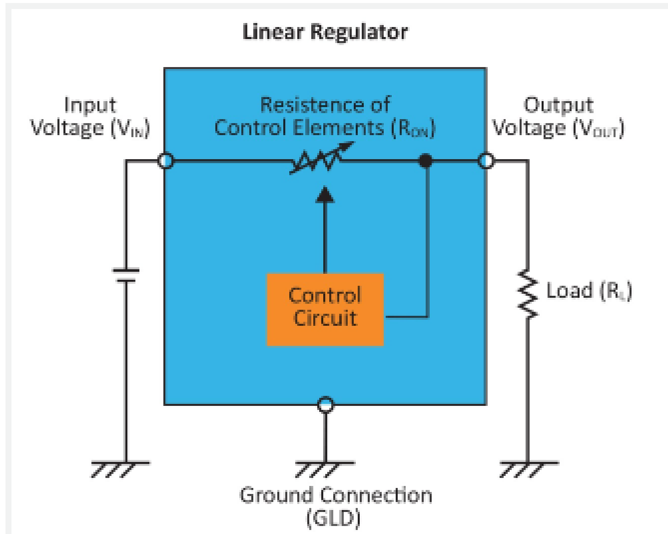

Figure 16: Linear voltage regulator with low dropout (copyright pending)

3.5.2 Switching Regulators

Due to their high efficiency, compact solution sizes, and versatility to step-down, step-up, or invert voltages, switching regulators are currently the most often used type of DC/DC power conversion. As implied by its name, switching regulators regulate the turning on and off, of a switch (transistor) to move energy from its input to its output. Some important additional elements required in a switch design include inductors and capacitors for energy storage and diodes for controlling the direction of currents. The magnetic field of the inductor stores and releases energy as the switch cycles on and off. Switching regulators can effectively transform power by controlling how much energy is kept or released by the inductor.

3.5.3 Voltage Regulator Selection

For the purposes of this project there may need to be multiple power supply regulators. The most common voltages being 12V, 5V, and 3.3V. Our supply voltage is from the battery with a nominal 36V DC. On the one hand linear regulators are very simple but have a low efficiency and switching regulators have higher efficiency but are more complex. We will choose a switching regulator for each of our main power voltages. These power circuits can be designed manually, but there are many free tools that can be used to design power circuits such as WEBENCH Power Designer by TI. Using these tools parameters can be used to choose high efficiency, low cost, small footprints, or a balance of all options. Once all components and their power requirements are determined we can use a tool to create our necessary circuit for the main control board.

3.6 Motors

In this section of our document, we will be going over and comparing the different motor types that we could possibly use for our design. This is just to give a scope of the type of options that we may have for our design. Also not only are we going to be comparing them to each other but we will also give some insight on which ones will and will not work for our project.

3.6.1 AC Motor

An electric motor known as an AC motor, or alternating current motor, transforms electrical current into mechanical power through the use of a stator and a coil that are both powered by alternating current. AC motors can be single- or three-phase, with three-phase motors typically being used for converting large amounts of electricity. Small power conversions employ single phase AC motors. Due to their adaptability, efficiency, and quiet operation, AC motors are a power source for a wide range of applications. They are utilized on pumps, water heaters, gardening tools, off-road vehicles, and many other products, equipment, and appliances. As they can readily fit into a variety of applications, they are an engaging and interesting instrument.

Types of AC Motors:
Single Phase AC Motor
Polyphase AC Motor
Synchronous AC Motor
Reluctance AC Motor

Hysteresis AC Motor
Repulsion Motor
Asynchronous Motor

Table 14: Types of AC Motors

The frequency and voltage of the circuit affect the speed and torque of an AC motor. With the help of variable frequency drives, you can control the speed and torque of your motor and maximize the efficiency of its operation. Variable frequency drives also known as VFD are seen in many industrial applications. Unfortunately, due to our own design constraints we could not go in the direction of an AC motor mainly due to its size and design.

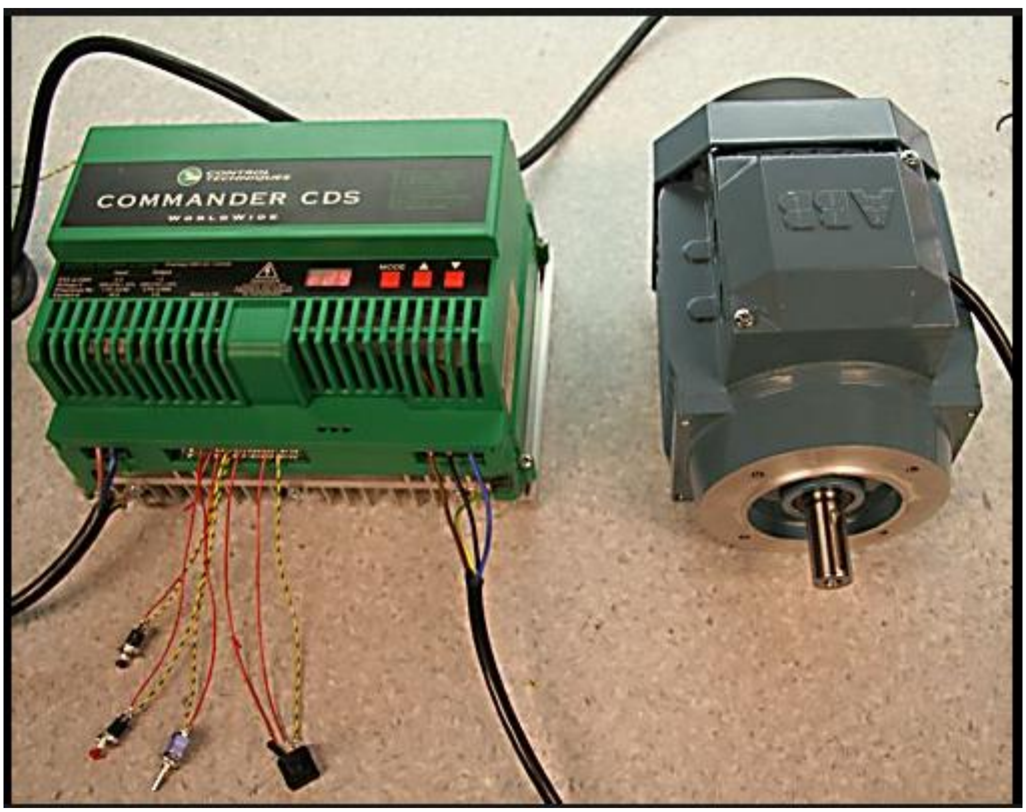

Figure 17: AC motor with VFD (copyright pending)

3.6.2 DC Motors

A DC motor, also known as a direct current motor, is a type of electrical device that uses direct current to generate a magnetic field that converts electrical energy into mechanical energy. A magnetic field is produced in the stator of a DC motor when it is energized. Magnets on the rotor are drawn to and drawn away by the field, which rotates the rotor. The commutator, which is connected to brushes and the power source, supplies current to the motor's wire windings in order to keep the rotor turning continuously. The capacity of DC motors to precisely control their speed, which is essential for industrial gear, is one of the reasons they are favored over other types of motors. The ability of DC motors to instantly start, stop, and reverse is crucial for managing the functioning of production machinery.

Understanding the different types of DC motors is crucial to appreciating their advantages. Before buying and using, it is important to consider the advantageous qualities that each type of DC motor has to offer. The ease of installation and low maintenance requirements of DC motors over AC motors are two of their key benefits. The connections between the field winding and the armature distinguish different types of DC motors. The field winding can be connected in a series or parallel to the armature. The connection can occasionally be both parallel and in series. Whether a DC motor's rotor is operated by a brush, or a brushless motor is another way to distinguish them. Brushes in brush DC motors transfer electricity to the rotor. The rotor of a brushless DC motor is permanently magnetized. There is a different type of DC motor to suit any application's requirements because they are ubiquitous and used for a wide range of purposes. In regard to our project design, we have come to a decision in going forward with a DC motor due to the design and functionality of it.

Types of DC Motors:
Brushed DC Motor
Separately Excited DC Motor
Permanent Magnet DC Motor
Self-Excited DC Motor
Shunt
Series

Compound
BLDC Motor
Servo DC Motor

Table 15 : Types of DC Motors

3.6.2.1 Brushed DC Motor

Wound wire coils are used in DC motors to produce a magnetic field. These coils, which make up the "rotor" of a brushed motor, are free to rotate in order to drive a shaft. The coils are typically wound around an iron core, however there are brushed motors that are "coreless," meaning the winding is supported by itself. The magnetic field of the rotor must be rotating continually in order to attract and repel the fixed magnetic field of the stator, which in turn produces the torque that causes the rotor to spin. A sliding electrical switch is utilized to rotate the field.

The brushes and commutator experience mechanical wear over the course of the motor because there is some mechanical friction between them and since, as an electrical contact, they are typically unable to be lubricated. The motor will eventually stop working as a result of this wear. Large, brushed motors often include replaceable brushes made of carbon that are intended to retain good contact over time. Periodic maintenance is required for these motors. Even with replaceable brushes, the commutator eventually wears out to the point where a new motor is required.

The mass of the rotor, the brushes, and the commutator, as well as other factors, can all affect the rotational speed of brushed motors. Brush arcing can become more pronounced and the brush to commutator contact can become irregular at very high speeds. The rotor of the majority of brushed motors also contains a laminated iron core, giving them a high rotational inertia. This restricts how quickly the motor may accelerate and decelerate. Strong rare earth magnets on the rotor can be used to create brushless motors, which reduce rotational inertia. Naturally, that raises the price. This essentially is what makes the brush DC motor not viable for our design due to its short lifetime and low efficiency.

3.6.2.2 Brushless DC Motor

In a Brushless DC motor, the rotor's wire is instead powered by transistors in the electronic control circuitry, which creates an alternating current from a DC supply to reverse the current every half-cycle to achieve continuous rotation. Brushless DC motors often have higher torque-to-power ratios, are smoother and more efficient than brushed motors, and can operate at higher speeds with more precise control. They require less maintenance and have a longer useful life because there is no wear on a brush or commutator. The cost of the brushless motor, as well as the more complicated drive circuitry that is required, is one of its primary disadvantages.

Changing the control voltage can be done digitally using a microprocessor, or analogy with analog components. In order to activate the right phase at the appropriate time, the control circuitry needs to be aware of the motor's relative angular position. The rotational angle can be determined without sensors by inferring it from the back EMF produced by the magnetic field, or it can be determined with sensors by employing an optical encoder or Hall effect sensor. In either scenario, an all-in-one motor driver that combines the necessary features onto a single chip is frequently employed. This is what we are leading towards our design in the project.

	Brushless DC Motor	**Brush DC Motor**
Efficiency	High	Low
Maintenance	Little	Frequent
Thermal Performance	High	Low
Output Power over Frame Size	High	Moderate
Speed/Torque Characteristic	Flat	Moderately flat
Dynamic Response	Fast	Slower
Speed Range	High	Low

Noise	Low	High
Lifespan	Long	Short

Table 16: Brush vs Brushless DC Motor

3.6.2.2.1 Hub Motors

The hub motor, which is sleek and unobtrusive, is slowly becoming the go-to drive system for many light electric vehicles, including scooters, solar cars, and e bikes. External mounting brackets and drive chains are not required with a hub motor conversion to support a motor and transmission. Instead, everything is housed inside the wheel, which attaches to your bike just like any other wheel. Hub motors can be divided into two fundamental groups: direct drive and geared.

3.6.2.2.2 Geared Hub Motors

A geared hub motor has gearing inside of it to lower the high speed of a powerful and effective motor to the low speed of the wheel. A geared hub motor is typically wider and has a lower radius than a straight drive hub. They can have a variety of various internal configurations, but most typically they contain an outrunner motor that powers a planetary gear set attached to the rotor. The hub motor is straightforward to install and has the weight advantages of a transmission drive packaged into it thanks to the geared hub idea. They often weigh approximately 50% less than a direct drive machine of comparable power, and they frequently produce more torque. A conventional direct drive machine can only generate about 35 Newton-meters (N-m) of torque, however the German-made Heinzmann can generate up to 80 N-m.

A freewheel is within almost all the available geared hub motors. This eliminates the possibility of regenerative braking but implies that there is very little rolling friction to spin the wheel when the motor is off. Weighing these benefits of the geared hubs against their drawbacks is necessary. In general, geared hubs are more expensive, have several moving parts that are prone to wear, and make audible noise. Whereas geared hub motors with various speeds are theoretically possible, every device on the market right now only has a single speed. Some people fervently contend that e-bikes ought to feature a variety of transmission ratios so that the same 500-watt motor may both propel you slowly up a 15-degree hill and propel you quickly along a flat surface while always functioning in the vicinity of its peak efficiency zone. Yet if the market is any guide, the additional prices and difficulties from multi-speed drives surpass their advantages.

Figure 18: Geared hub motor (copyright pending)

3.6.2.2.3 Direct Drive

The direct drive hub motor is as straightforward as they get. Think of taking an electric motor and holding the axle while allowing the motor's body to rotate. The only moving component is the wheel itself when you hook a bicycle rim to this rotating motor body. These machines are typically radial-flux Brushless DC (BLDC) units with a grid of permanent magnets on the hub's interior surface. Alternating currents passing through the stator windings, which are connected to the axle, cause the hub to rotate. In a DC hub motor, the windings are spinning on the interior of the hub while the magnets are on the axle.

Through a commutator plate, a carbon brush delivers energy to the rotating windings. The direct drive system's one drawback is that, considering their power output, they are typically big and heavy. The wheel speed is only about 200 rpm, which is the cause of this. Since the speed between the magnets and the winding is directly related to the power density available from an electric machine, a large motor is required to produce enough power and torque. A considerably smaller motor can produce the same amount of power in a geared transmission since the motor is frequently rotating at speeds over 3000 rpm.

Direct drive hub motors all have the feature of always being mechanically engaged. Something has both good and bad aspects. It is advantageous because it enables regenerative braking. Whether they are designed to do so or not, all direct drive hub motors can be engineered to do regeneration. In contrast to the majority of Chinese kits like Crystalyte, Golden Island, and Wilderness Energy, the BionX and TidalForce both contain sophisticated controllers that enable varying amounts of braking energy to be stored back in the battery. Constantly engaged is undesirable since it implies that even when the motor is not being used, you are constantly overcoming its rolling resistance. This additional drag torque can range from being invisible to making you feel like you're always riding with a flat, depending on the symmetry and quality of the motor.

Figure 19: Direct Drive Motor (copyright pending)

3.6.2.2.4 Direct Drive vs. Geared Hub Motors

In the table provided below is a general comparison of the Direct Drive and Geared Hub Motors. With the data that was collected and researched we came to terms with the categories in which we deemed to be important for our design specifications.

	Geared Hub	**Direct Drive**
Energy Efficiency	High Efficiency	Lower Efficiency
Torque	Higher in torque	Low in torque
Weight	Lighter in weight	Heavy in weight
Performance	Low performance	Higher performance
Price	Cheaper	Higher in price

Table 17 : Comparison of Motor types

3.6.2.2.5 Hall Sensors

A BLDC motor's commutation is electronically controlled in contrast to a brushed DC motor. The stator windings should be activated sequentially to turn the BLDC motor. Understanding which winding will be powered after the energizing sequence depends on knowing the position of the rotor. Hall effect sensors built within the stator are used to determine the position of the rotor. On the non-driving end of the motor, the stator of most BLDC motors has three hall sensors included into it. The hall sensors provide a high or low signal whenever the magnetic poles of the rotor get close to them, indicating whether the north or south pole is doing so. The precise commutation order can be calculated using the combination of these three hall sensor signals.

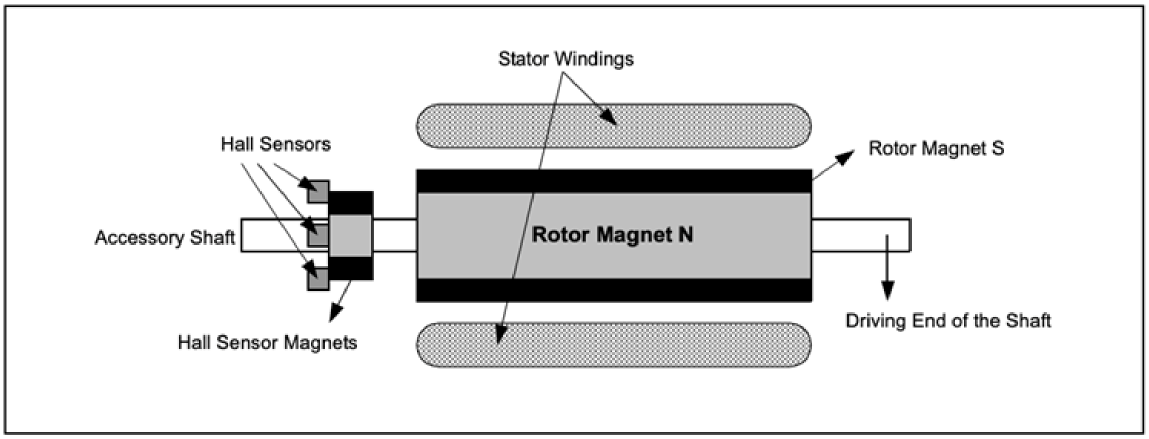

Figure 20: Tranverse Section of a BLDC Motor (copyright pending)

A BLDC motor with an alternate north and south permanent magnet rotor is shown in transverse in the following figure above. The stationary portion of the motor contains Hall sensors. Because any misalignment of the Hall sensors with respect to the rotor magnets will result in a mistake in determining the rotor position, embedding the Hall sensors into the stator is a difficult task. Certain motors may include the Hall sensor magnets on the rotor in addition to the primary rotor magnets, making it easier to place the hall sensors onto the stator. They are a miniature duplicate of the rotor.

As a result, the hall sensor magnets have the same impact as the primary magnets whenever the rotor turns. Typically, the hall sensors are fastened to the enclosure cap on the non-driving end by being installed on a PC board. This gives customers the ability to align the entire hall sensor assembly with the rotor magnets for the greatest performance. There are two different forms of output depending on where the hall sensors are physically located. The phase shift between the Hall sensors can be 60 degrees or 120 degrees. Based on this, the manufacturer of the motor specifies the commutation sequence that must be used while operating the motor. Especially these being a major component in regards to our specifications within our design.

3.6.3 Motor Selection

Figure 21: Bafang G310 Motor Schematic (copyright pending)

Specifications	**G310 Motor**
Actual weight (kg)	2.55
Power Range (Watts)	250 - 500
Motor KV (RPM/V)	8.5
Phase Resistance (Ohm)	0.124
Disk Brake Compatible	Yes
Connectors	Z910
Thermistor style	10K NTC, B – 3450 with 10K pullup Multiplexed with Speedo
Axle Length	138
Axle Threads	3/8” x 26 tpi

Spoke holes	36 Hole
Motor type	Geared
Flange Spoke Diameter (mm)	123
Magnetic Pole Pairs (Phaserunner)	88
Hysteresis Losses (N-m)	0.666
Eddie Losses (N-m / rad/sec)	0.015
Motor Inductance (H)	0.00021

Table 18: Motor Design Specifications

With a lot of the research that was conducted in terms of the motor we came to the decision of choosing a geared rear hub motor. The one in which we are looking at is the *Bafang G310 Standard Wind (8.5 rpm/V) Geared Rear Hub Motor*. The *Bafang G310* motor is a 2.5-kilogram rear hub that is compatible with small disks and has a nominal 250–350-watt power rating. It works well for covert systems that don't need much assistance for steep slopes or heavy loads. In order to use a CA-DP device and avoid the need for an external speedo pickup, this motor has an integrated 6-pole speedometer sensor. It has spiral gears on a second stage planetary gear's first reduction and is remarkably quiet for a geared motor. uses a side cable exit and a cassette freehub mechanism rather than a screw-on freewheel. This hub has a typical RPM/V winding that is appropriate for 26" and 700c wheels. The motors price point before tax is at about $240 USD. In addition, with the purchase of the motor we are able to have the company assemble the motor to the rim for a certain fee.

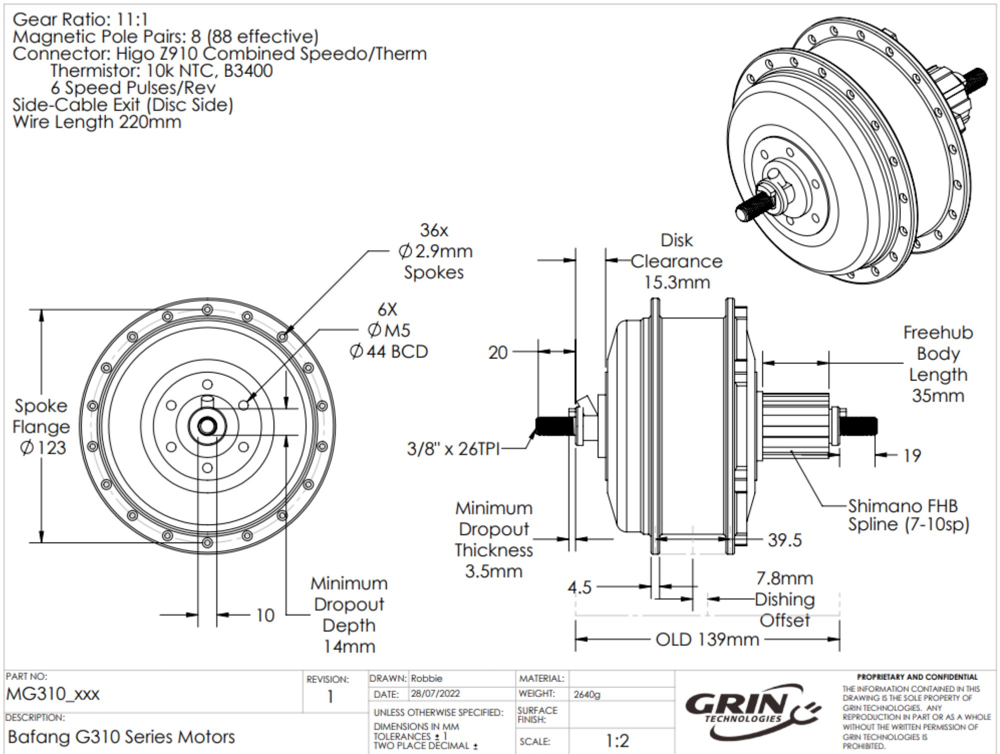

Figure 22: Bafang G310 Motor Schematic (copyright pending)

3.7 Throttle Controls

The following section is an evaluation of the different types of throttle controls that an operator can interact with to propel an e-bike. There are a few different kinds of throttles; each has their pros and cons, and they should be weighed against each other to determine which is best for the project.

3.7.1 - What is an E-Bike Throttle?

According to the Oxford dictionary, the technical definition of a throttle is a device that controls the flow of fuel/power to an engine. Typically, the operator of a vehicle is the one that has access to the throttle. Although an e-bike does not have nor requires fuel to operate, it does require power and a device to control how much power is provided. This device is what the operator uses to propel the e-bike and control the power flow to the motor. This specification is the reason why this device is classified as a throttle.

3.7.2 - Popular Throttle Types

If one has ever ridden a motorcycle or an electric scooter or anything of the like, they'll know that the throttle on these vehicles is located at the handlebars. An e-bike's throttle is typically located at the handlebars as well for ease of access for the operator. Since it is typically located at the handlebars, the weight/size of this device is something to consider. If this device is too heavy, it can negatively impact the operator's intent on directional control. If this device is too big, then it will negatively impact the operator's comfort when operating the e-bike. To minimize the weight/area this device takes up, some compact designs for throttles have become popular. Of these compact designs, there are two to make note of:

1) Twist Throttle

The twist throttle is a throttle device located in the handle of the e-bike itself. This is identical to the throttle a motorcycle uses. The handle can be twisted; the more it's twisted, the more power is output, thus more acceleration is experienced. There are two types of twist throttles:

a) Half-twist throttle

Only the inner half of the handle can twist to control speed

Figure 23: Picture of a half-twist throttle on an e-bike (copyright pending)

b) Full-twist throttle

The entire handle can twist

Figure 24: Picture of a full-twist throttle on an e-bike (copyright pending)

2) Thumb Throttle

The thumb throttle is located at the edge of either the left or right handle, fashioned in such a way that the thumb can comfortably rest on it. To engage the throttle, the operator presses their thumb down on the throttle's tab.

Figure 25: Picture of a thumb throttle on an e-bike (copyright pending)

3.7.3 - Consideration for Project

Both throttle designs are popular in the e-bike community. Although, there appears to be a debate surrounding which design can be considered “better” than the other. This, one can assume, comes down to personal preference. When it comes down to which will be used in this project, that will be the result of the same debate: which is the design the group prefers. There are numerous things to consider when making such a decision, as either throttle choice has its pros and cons. Below is a quick list of these pros and cons for each of the throttle types.

Twist throttle pros and cons:

- Pros
    - Allows for the operator’s entire hand to maintain contact with the handle
    - Can be more comfortable over long distances over the thumb throttle, as long as frequent speed changed are not required
- Cons
    - If frequent speed changes are required, then constant adjustments must be made; this can introduce wrist fatigue
    - May give uneven handle feeling, as the throttle handle may be much bigger than the other handle
    - May be difficult to implement, as an entire handle must be replaced
- Half twist vs full twist
    - Half twist
        - Pros
            - Smaller than full twist, more compact
            - Less accidental engagements
        - Cons
            - May be the most difficult to attach
            - May not be as comfortable as full twist
    - Full twist
        - Pros
            - Larger than full twist; better control
            - May be easier to attach to e-bike
            - May be more comfortable for hand
        - Cons
            - Prone to accidental engagements when running to wall/object

Thumb throttle pros and cons:

- Pros
    - Will not introduce wrist fatigue with constant speed adjustments

    - Easier to implement; a thumb throttle attaches to the handlebar as is
    - Prevents uneven handle feeling as both handles will be the same size
- Cons
    - May be less comfortable over long distances, as the thumb is doing all the work to maintain speed
    - Operating e-bike over difficult, bumpy terrain can impact operator's reliability to maintain speeds
        - If the bike goes up and down, so does the thumb

If the group decides they want to focus on giving the e-bike a sleek and minimal look/feel or are more comfortable with twisting their wrist, then the twist throttle should be considered.

If the group decides they prefer a static hand grip and are uncomfortable with twisting their hand, then the thumb throttle should be considered.

Throttle Type	**Comfort**	**Implementation Ease**	**Grip Rigidity**
Thumb	Low	High	Low
Half-Twist	High	Lower than Thumb	High
Full-Twist	High	Lower than Thumb	Higher than half-twist

Table 19 : Types of Throttle

3.7.4 Throttle Selection

For the throttle on the e-bike, the group has decided to go with a twist throttle; the full-twist, specifically. For this selection, the *Keenso Full-Twist* throttle will be used. The image of the exact throttles will be provided below.

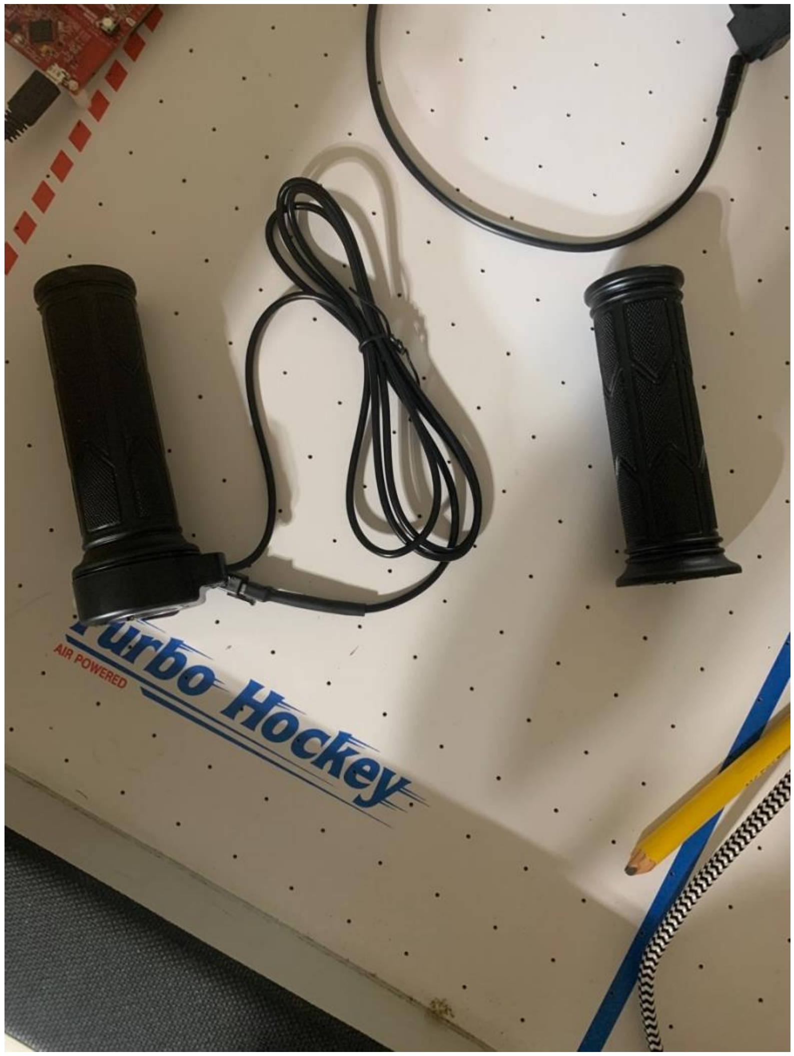

Figure 26: throttles we will use

3.8 Torque arm

Torque arms are used in hub motors in order to prevent axle rotation. Essentially with the torque being generated one way there is an equal amount of it in the opposite direction. Most electric bike hub motors have flats carved into the axle that fit into the dropout slot and provide some rotational stability. Nonetheless, in high power systems that generate a lot of torque or in designs with weak dropouts, the forces present may be greater than the material strength and push the dropout open. The axle will spin freely in that scenario, possibly causing the wheel to fly off the bike and wrapping and severing the motor cables. When looking for a viable option for the torque arm there are some factors in which to take in account. The material in which you are looking for is stainless steel for how strong and durable it is. Next being the thickness of the torque arm needs to be looked at when selecting it. In most cases thicker is better so that it is able to have more of an effect in the design. An estimated measurement that is recommended is about a quarter inch or 0.635 cm. Lastly is the length in which the axle is from the point where the torque arm is mounted to the frame. Greater the distance equates the amount of force that is being resisted on the torque arm.

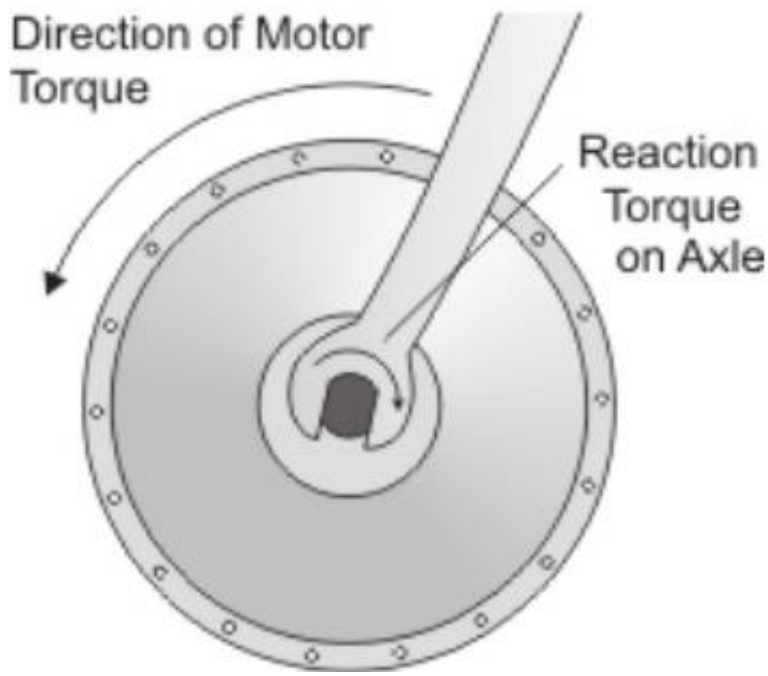

Figure 27: Torque Arm (Copyright pending)

3.9 Brakes

Brakes are used as a way to slow down objects that are in motion that need to stop. There are many different types of brakes used on bicycles, however, only two are widely used which are rim brakes and disc brakes. There are three more brakes used by bicycles, however, they are not as common as the previous two so the information given for those brakes will not be as in depth compared to the first two. We may need to replace ours with due to the fact that the brakes currently on our bicycle was not designed to stop someone using a motor to propel themselves.

The company REI and the website bikeradar.com state that the most common type of brakes on bicycles today are rim brakes or disc brakes. They each have different opinions on which brakes are used on bicycles other than the two stated above. REI states that coaster and drum brakes are also used on bicycles while bikeradar states that V-brakes are used. The safety standards for all of the brake types will be found in 4.0 related standards and design constraints.

3.9.1 Rim Brakes

These brakes use the rim of the wheel in order to apply a force to slow down and to stop. They are simple using only a wire on the bicycle we currently plan to use for the project. Currently rim brakes can cost anywhere from $10 up to $544. This wide range of price is most likely because of materials used or if a company has a sort of “elite” status like Apple™ has.

These brakes have upsides and downsides as with any item with an upside being a lower cost compared to other brakes. The downsides consist of poor performance when it rains, which is common occurrence in Florida depending on the season, however, this can be overcome if a proper pad is used. There are also issues of mud and snow with snow being a non-factor in Florida. There are different types of rim brakes that can be used and the main difference between them is the way the force is applied. For example, a rod-actuated brake uses rods and pivots to apply force on the rim when the brake is squeezed to apply pressure. The one on the bike that is being planned to be used are called side-pull caliper brakes but those also have a subdivision with one being a single-pivot side-pull caliper brakes and the other being a dual-pivot side-pull caliper brakes. Below is the type of brake currently on the bike.

Figure 28: Single pivot-side-pull caliper brake

These are brakes that are generally used on narrow tires because with a wider tire the force applied decreases.

On figure 23 there are labels that show the major parts of the caliper brake. At A and B is what the person moves when pulling on the brake. The two parts move closer together causing C to pivot causing D and E to get closer to the rim until one is touching then as more force is applied the other touches. When not being pulled the spring inside C makes the pads D and E stay away from the rim of the tire.

3.9.2 Disc Brakes

These brakes are attached at the wheel hub of the tire that rotates along with the tire. While rotating a force can be applied to the disc to slow or stop the bicycle. A disadvantage is heat dissipation because it is possible to have a small surface area if you need a small disc brake because of weight reasons compared to rim brakes. An example of a dick brake is below.

Figure 29: Hydraulic disc brake

The reason for the holes in the disc is to help reduce weight and with heat dissipation. While rims are inherently part of the design on a bicycle (the wheels) a disc brake is an addition to the bicycle that must be added to after the fact. This would not be best when we are trying to make the e-bike have a higher speed and have a longer lasting battery.

On figure 24 there are labels that will be used to explain more about the disc brake. A is a line where force is used to push on B the caliper. A could be hydraulic or cable actuated. C is the rotor that the caliper applies force to in order to slow down or stop the bicycle.

3.9.3 Coaster Brakes

A coaster brake is engaged when the pedal is rotated in the opposite direction, an amount that can be changed depending on the distance between the clutch and the driver side expander. During normal operation (moving forward) the clutch grabs onto the casing so the wheel and casing are moving in sync. When back pedaling the clutch goes along a thread to push against the driver side expander. This expander goes inward pushing internal plates up against the casing which causes friction against the container slowing you or stopping you.

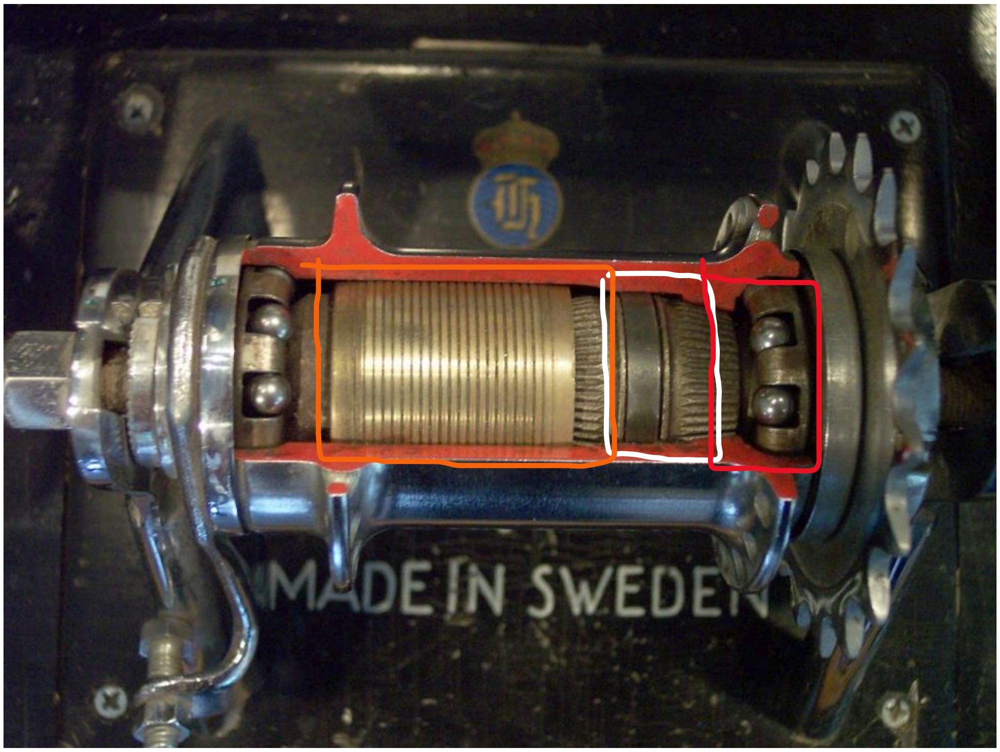

Figure 30: Inside a coaster brake

In the picture above the red square is the clutch which has teeth that grabs onto the white box (driver side expander) which then pushes the orange box (brake pads). The brake pads have a slope on the underside of them so when the clutch is not engaged, they rest on top of the expander and apply no force to the container. To disengage the brakes going forward with the pedals release the brake pads.

3.9.4 Drum Brakes

The drum brakes can also commonly be seen on cars and use a wheel cylinder to push the brake pads (also called shoes) against the drum to apply the friction required to slow or stop. This is similar to the coaster brake in that they both push metal on metal to stop or slow down. The drum brakes, however, use springs in order to disengage the brake pads from the drum. In a vehicle hydraulics are used to engage the brakes but, in a bicycle, cables are used instead.

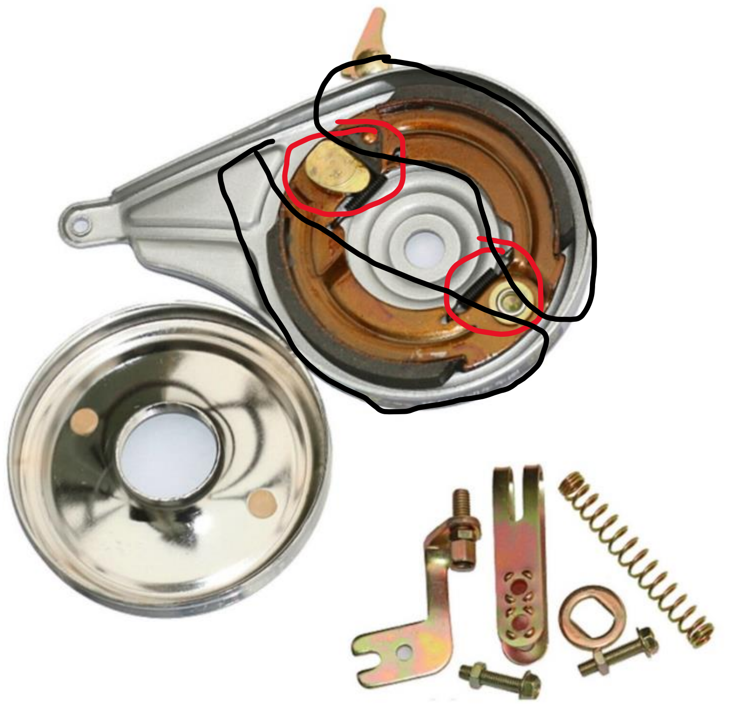

Figure 31: Internal of a drum brake

In the picture above the black are the brake pads while the red is the wheel cylinder. When a force is applied the brake pads are pushed into the sides of the drum to apply friction.

3.9.1 Brake selection

There are a few different types of brakes we could use, however, we will need to determine what would be the best option for the e-bike we will use based off the requirements we have. The two most common are the rim brakes and the disc brakes. Each has their own positives and negatives that will need to be taken into account when deciding which one to utilize.

The speed of the e-bike will be going fairly fast with a max of 19.5 miles per hour so as not to go beyond the class two e-bike. We can look at automobiles for some inspiration for the type of brakes we should use. Automobiles use disc brakes because of the stronger stopping power over that of other types in the front, however, recent information states that disc brakes are now used on all wheels. This is an option for an e-bike that we could consider because of the added momentum and weight that is added because of all the parts added to the e-bike.

	Rim brake	**Disc brake**
Market availability	High	High
Cost	High	Medium
Heat dissipation	Low to high Depends on size of disc	High
Weight	High	Low
Reliability	High	Medium
Ease of replacement	Low	High

Table 20 : Comparing Types of Brakes

From the table we can see these differences between the two most common types of brakes along with their positives and negatives. We will need to decide if we want the disc brakes but because of the nature of the project rim and the use case for the bicycle, rim brakes should be fine. They should also allow easier replacement and are cheaper if something does go wrong or brakes while testing or fixing the e-bike.

The rim brake type we would use is a dual-pivot side-pull rim brake. The reason for this is the better braking compared to other rim brake types. Below is an example of that type of rim brake.

Figure 32: Dual pivot side-pull caliper brake

The current choice is *SUNLIGHT Dual Pivot Brake Caliper, 39-49mm Reach*. This is because the cost is low and we will also need to purchase a cable set so it is not too much extra work and would allow us to have a better understanding of how these rim brakes work.

3.9.2 E brake

Figure 33: e brake we will use

While this is not the actual brake that will be used it will be used to slow down the bike it will tell the MCU that the brake has been engaged and will tell the motor to no longer drive forward.

3.10 - App

The group has decided upon building a multi-function smartphone companion application for the e-bike. This application will connect to the e-bike via Bluetooth and send & receive information to and from the e-bike. This will be made possible by the aforementioned Bluetooth module in section 3.4. The application will mainly behave as a heads-up-display (HUD) for the operator of the e-bike. The e-bike will feature a phone clamp on the handlebars and, along with the application receiving information, can be used as a visual display for certain statistics associated with the e-bike's function: current speed, remaining battery charge, and current battery usage, amongst other things.

There are different options for developing an application. The group can either opt to:

- Develop a web application that a smartphone can access via a web browser, or
- Develop an executable application specifically for the smartphone

Both options have their advantages, and both must be considered to find the best possible option for the group.

3.10.1 – Browser Web Application

With an application comes a technology stack that is used to build the application. A technology stack is a collection of technologies used to build an application. A stack is typically identified by either an abbreviation or a short phrase. For example: a LAMP stack. LAMP is an abbreviation used to identify the technologies: Linux, Apache, MySQL, and PHP. The abbreviation does not always identify the same scope for each letter.

When it comes to experience, however, there are two technology stacks the CpE group members are familiar with: LAMP and MERN stacks. When it comes to a certain stack that the group has mentioned may work the best for the project

When considering which one will function the way the group requires, any of the choices we come across are feasible. We are given freedom to use whatever technology we desire, though this technology must be something that can we are capable of using properly to achieve our goals. That being said, the main factor to consider when deciding which technology stack to use would be whatever the group is most comfortable with using.

3.10.1.1 – LAMP Stack

The LAMP stack, as previously mentioned, consists of Linux, Apache, MySQL, and PHP. Linux is used as the operating system (OS), Apache is web server software and is used to host the application, MySQL is used for databasing and storing information, and PHP is used for API calls between the UI and the database, sending and receiving information.

According to Amazon, the following five concepts are why a LAMP stack is an important contender for consideration when building an application:

- Cost
  - Cost is an extremely important factor when considering technology to use. Fortunately, all of the components of a LAMP stack are free. A developer is capable of downloading all the technologies and using them without spending a dime.
- Efficiency
  - LAMP is a tried-and-true tech stack that has been used numerous times. This can prove to be efficient, as the technologies are familiar to most developers, saving time to learn and set up.
- Maintenance
  - The maintenance of the technology involved with LAMP very good, as software experts from around the world contribute towards its development and maintenance.

    - There will still be maintenance on the developer's side for app-specific issues, but issues with the stack are well-maintained.

- Support
  - The number of developers around the world using LAMP is, to exaggerate, astronomical. The amount of documentation provided by its IT community can prove to be extremely useful, as experience is the best form of troubleshooting.
- Flexibility
  - The LAMP stack is flexible as certain technologies within the stack can be replaces by others. For example: Linux can be replaced by either Windows or MacOS to result in a WAMP or MAMP stack, respectively.

A LAMP stack would be a great option for app development, after considering the above reasons. However, to consider everything about it, not everyone in the group is familiar with Linux, since although it's a popular operating system, it is not as broadly used as Windows and MacOS are.

LAMP stacks are also better used for backend applications, ones that are hidden from end users. While DOM technologies can be incorporated into a LAMP stack to produce user interfaces (HTML, CSS), they are not explicitly included.

3.10.1.2 – MERN Stack

A MERN stack, similarly to a LAMP stack, is an abbreviation for the technologies used within its development. A MERN stack consists of the following technologies: MongoDB, Express.js, React.js, and Node.js. MongoDB is the database technology, Express.js a JavaScript web server framework, React.js is a front-end JavaScript library, and Node.js is a runtime environment.

Using similar factors to Amazon's evaluation of the LAMP stack:

- Cost
  - A MERN stack is not completely free, as MongoDB alone has a bit of cost to use, based on how much storage one requires. The other technologies are free frameworks/libraries open to use for everyone.
- Efficiency
  - The efficiency of a MERN stack is very high, as the web server, frontend, and runtime environment are all written in JavaScript.
- Maintenance
  - Maintaining a MERN stack application can prove to be quite simple since everything is written in one language. The learning curve begins with using MongoDB, as it is a document-based, no-SQL database. If one is primarily used to SQL databasing, this may prove difficult.
- Support

    - MongoDB is a company with a good support platform. They have entire websites dedicated to assistance for their customers. The JavaScript technologies in the stack would also prove to have good support as JavaScript is a very popular scripting language; the global IT community is very present in providing support for certain issues.
- Flexibility
    - The MERN stack is very flexible. Instead of using React.js for the frontend framework, once can opt to use Angular.js or even Vue.js, resulting in a MEAN stack or a MEVN stack.

3.10.2 – Smartphone Application

There are many things to consider when developing an application directly for a smartphone, with the most important factor being: which platform should the application be developed for: iOS, Android, or something else? For some, this is a simple answer: whichever they prefer the most. However, in this group's case, both iOS and Android platforms are an option as the group holds access to both.

3.10.2.1 - Platform

The platform that the application is developed for is incredibly important for development. Popular platforms that are available for development are iOS and Android; these two options not only dominate the smartphone market, but also are the only two platforms that the group holds access to. Either one of these would be a great option for development, since these operating systems are developed by incredibly large corporations that provide fantastic tools to assist with development.

- iOS
    - iOS applications are developed with Apple's own programming language: Swift. Applications also rely on Apple's XCode tool for development, which, apparently, is easy to work with. The unfortunate thing here is that XCode does not run natively on Windows computers, so either a Mac would be required for development, or a virtual machine running macOS.
- Android
    - Androids depend on Android Studio for development, which was introduced by Google in 2013 and has continued to be a frontrunner for Android app development for almost that entire decade. Android studio uses the Kotlin programming language to develop applications. An advantage of Android Studio is that it, while utilizing a plugin, can develop cross-platform applications for both iOS and Android devices.

3.10.2.2 – Flutter and a BaaS

One thing that the previously mentioned technology stacks have in common is their incredible ability to develop web applications. One thing they specifically lack, however, is the ability to port the web applications into mobile smartphone applications. Not only is this portability an issue, but also being able to port the web application to multiple different platforms.

This is one of the main advantages of Flutter. It is a UI framework developed by Google that uses its own Dart programming language, which is also developed by Google. Flutter is capable of compiling developed applications for multiple different instruction architectures; in the scope of this project, using Flutter would mean developing an application for iOS and Android devices at the same time. This would prove extremely useful since the group does not all share the same operating system on their smartphones, so using a technology that compiles into both operating systems enables the application to be more inclusive with testing.

Alongside Flutter would be a Backend-as-a-Service (BaaS). A BaaS is a simple way for developers to automate certain backend development processes and provide a cloud infrastructure management that is simple to work with. There are many choices for a BaaS to pair with Flutter. There are a few that are popularly used and are worth considering:

- Back4App
- Parse
- Firebase
- Backendless
- AWS Amplify

The above options provide similar functions to each other, while all having similar price points. These options are not free but are worth seriously considering, as streamlining app development for a negligible price tag seems more than worth it.

Approaching Flutter and BaaS similarly to the other tech stacks:

- Cost
    - Will cost more than the other two listed options.
- Efficiency
    - May be less efficient in learning to use the technology, since both Flutter and Baas are new concepts to the group. However, once enough learning has been done to develop, development efficiency would be unparalleled, considering the automation capabilities of a BaaS as well Flutter’s compilation into multiple architectures.
- Maintenance
    - Depending on which BaaS would be selected for the pairing with Flutter, there would be incredible maintenance for these technologies. For example: if Firebase were selected. Both Flutter and Firebase are developed and maintained by Google. AWS Amplify is developed and maintained by Amazon.

- Support
    - Google invested a lot of its resources into Flutter; it can be assumed having a great support team is part of this investment. The same applies to the BaaS options as well, regardless of which one it is.
- Flexibility
    - Contrary to the other great evaluations, the flexibility of Flutter is very low, if there's any at all. Dart is a programming language unique to Flutter, so applications developed on Flutter would have to stay on Flutter.

Using a ranking system of 1, 2, and 3 to rank the solutions on a comparison table (1 being the most optimal):

Technology	**Cost**	**Efficiency**	**Maintenance**	**Support**	**Flexibility**
LAMP	1	3	2	2	1
MERN	2	2	3	3	2
Flutter, BaaS	3	1	1	1	3

Table 21 : Comparison Table of Browser Web Apps

3.10.3 App Design

When considering many different factors including, but not limited to testing inclusion, learnability, and progress efficiency, the group has decided to use *Flutter* as the UI framework for the smartphone companion app, and, if a database were implemented, *Firebase* would be selected as the Baas. It can be expected that the documentation and support for it is of good quality/availability.

After exposure to Flutter and meeting promising milestones in developing the app, the group can safely say that Flutter was the correct decision for app development. Flutter itself is an incredibly forgiving and easy to set up technology. The group had no prior experience with Flutter, but we still managed to make such long strides with developing the app, even in Senior Design 1. There is still plenty of work to be done on the application as of writing this, but it should be noted that Flutter was an incredible decision.

The downside to using Flutter, however, is its native Dart programming language. On the surface it appears to be similar to Java in the sense that it is very object-oriented, it is not like Java in the slightest. The syntax in completely different for a great number of things, and that can be very confusing, especially to the group since we do not have prior Dart experience. The learning curve was incredibly steep, but it is possible to get around. As mentioned previously, the group has made significant strides in application development, and this is only possible due to the efforts put in to properly learn programming in Dart.

While Flutter was an incredible decision and still is, the decision of using Firebase as a Backend-as-a-Service (Baas) was ultimately deemed unnecessary. As of writing this, the application has been completed with the features the group set as requirements. Now that it is complete, the decision to not utilize a backend/database was a great decision for the scope of this project. Since there was no data collection occurring, the need for a backend/database was nonexistent; there is no communication between the application and a database. The features the group planned for the app did not require one, so implementing it would be redundant and take up precious development time.

3.10.4 Flutter Libraries

With Flutter comes a gigantic number of libraries that can assist development. Libraries are either developer-developed or even user-developed and include functions and objects that can assist/streamline development. The application uses some of these libraries, a few that are used are listed below:

- Flutter_reactive_ble (v5.0.3)
  - Provides a framework for the application to communicate via Bluetooth Low Energy
- App_settings (v4.2.0)
  - Includes a pathway to link user directly to their device's settings
- Location_permissions (v4.0.0)
  - Allows developer to ask the user for necessary device permissions (not just location)

3.11 – Lights

On most vehicles today exist lights. Whether they are lights on the front of the vehicle to illuminate the forward surroundings of the vehicle operator for clarity, or whether they are indication lights found elsewhere on the vehicle to indicate to others the direction the vehicle is going.

The group has decided placing not only front-facing illuminating lights but also indication lights on the e-bike is ideal. Not only is this ideal for operating a vehicle, it is also required by Florida state law, which states that bicycles must exhibit a white light on the front and a red light & reflector on the back. While operating an e-bike can be considered fun and entertaining, it must be safe not only to the operator, but to the operator's surroundings as well. Lights to illuminate the road ahead of the operator and to indicate deceleration (brake lights) are going to be installed. Lights to indicate change of direction/turning (turn signals) can be implemented if time allows.

There are two common technologies used to produce light: incandescent light bulbs or light emitting diodes (LEDs).

3.11.1 Incandescent Bulbs

Incandescent bulbs are the bulbs that are found most in vehicles as well as households. These bulbs consist of a glass bulb with a filament inside hooked up to a circuit that applies a current to the filament. The filament then heats up to a significant temperature (around 3500° – 4500°F) and radiation is emitted from the filament. This radiation is in the form of visible light. The term “incandescence” means to obtain light from heat.

Because of the functionality of incandescent bulbs, their total lifespan can be relatively short compared to other lighting technologies (around 1,200 hours). Not only is their lifespan very short, but they are also notoriously power-inefficient. This group’s e-bike is dependent on a single battery to operate everything electricity-based, so power efficiency is and incredibly important factor to consider.

Figure 34: A pair of incandescent bulbs (Copyright pending)

3.11.2 Light Emitting Diodes (LEDs)

Light emitting diodes (LEDs) are indeed a common lighting solution, however a smidge more expensive. LEDs consist of a semiconductor diode that produces light when a current is passed through it. Heat is generated in this scenario as well, but most LEDs implement heat sinks to pull heat away from the diode itself. LEDs capability to manage its thermal output is a significant factor in how long their lifespan is, which, compared to incandescent bulbs, is roughly 20-150 times longer (based on an average lifespan of 25,000 to 200,000 hours).

Not only do LEDs provide a longer lifespan, but they are also more power efficient. As previously mentioned, power efficiency is incredibly important to this project, as everything is being powered by a single battery. Since LEDs not only last longer but also are more efficient than incandescent bulbs, they would be wiser to select these when implementing lighting systems onto the e-bike. Even though they do cost more than incandescent bulbs, the money will be well spent.

Lighting Solution	**Cost**	**Efficiency**	**Lifespan**
Incandescent Bulbs	Low	Low	Low
LEDs	Relatively higher	Vastly higher	Vastly higher

Table 22: Lightning Comparison

Figure 35: A single, standalone LED (Copyright pending)

Figure 36: An LED strip (Copyright pending)

3.10 Lights Selection

Considering the options available for lighting, LEDs seem to be the most reasonable. Although they do cost a little more than incandescent bulbs, not only are they more reliable and dependent, but they are also much more power efficient. So the group has decided to go with LEDs for lighting on the e-bike.

There are many options of LEDs to choose from, but the group has opted to go with a strip of LEDs; specifically the *Adafruit RGB LED Weatherproof Flexi-strip 1m* strip of LEDs. This gives us the flexibility to cut and reuse sections of LEDs, with plenty of leeway in case any of the other pieces get damaged/become broken. This selection of type of light aids us creating some new and diverse for the project. With this light we have the possibility of having the app have the ability to change the colors of the strips in certain areas of the bike. Also we can have a diverse ability to customize our design with this type of LED since it can adhere to any place on the frame of the bike.

3.11 Motor Controller Selection

It is impossible to overstate the significance of an electric bike controller. They are necessary for connecting the electrical components of an electric bike. Various brands and models of ebike controllers are available. It's interesting how fierce the competition is amongst different businesses, making a decision tough. The effectiveness of your electric bike depends on the model you choose. Controllers for electric bikes are among the various parts that make up the bike's electrical system. They serve as connectors, joining parts like the batteries, the throttle, the pedal-assist, the brakes, the motor, etc. E-bike controllers are sometimes referred to as the heart and brain of the vehicle. The majority of an electric bike's functions are handled by controllers. Some of the duties that a motor controller is tasked with are over-voltage protection, brake protection, over-temperature protection, and over-current protection. For our project we were in the market for a BLDC controller for a motor containing hall sensors. On the rotor, they have a permanent magnet, and on the stator, an electromagnetic field. The hall sensor regulates how the circuit rotates and aids in scheduling when electricity needs to be given. While using hall sensors, this form of controller is comparable to BLDC motors. They have a potential to be driven electrically by using a control system due to the hall sensors.

The motor controller needs to be able to handle the current requirement of the motor going from the battery to the motor. This could be up to 30 amps at one time so one capable of handling that amperage is required. Our group was planning on making our own motor controller but toward the end of June we determined that the idea of doing that is not feasible with the time constraints. The one we decided to buy was a Flipsky mini VESC4 motor controller that can handle up to 50 amps continuous and has PWM input that we can use to control the motor using our MCU. The voltage tolerance is also acceptable at 60v max, while our battery will max out at 42 volts when fully charged. We had others that could only handle 10 amps which is not enough for our motor and battery, but we just used it to test that the motor could move and be controlled.

4.0 Related Standards and Design Constraints

There are many different governing bodies that regulate in three different ways. These governing bodies can regulate how something is done, i.e. our electrical grid is 240 volts at 60 Hz three-phase. They may require a specific technology to be used, i.e. in Europe recently small electronic connectors are required to be USB-C. The last way is to let the market figure it out which means that the people will vote with their wallets to get what works best for themselves.

In the United States the national standards body is called American National Standards Institute (ANSI). There are however industry standards that communicate with the national body. The one we most commonly hear about is IEEE, which is a Standards Developing Organization (SDO).

In our case the governing body for some parts of the bicycle is governed by ANSI, however, the original standards come from the International Organization for Standardization (ISO). ISO has numerous technical committees labeled ISO/TC “x”. The one for cycles is committee 149. They are also the committee responsible for developing the international standards that over 170 countries use.

The cycle safety standards from ISO created International Classification for Standards (ICS) which is a classification of the standard. For example, in our case our division would be 43 which is for road vehicles engineering. Then to further classify, the next number would be 150 for bicycles which would include their components and systems.

The standard is then called ISO “xxxx-x:YYYY” which is underneath the ICS classification. The standard for bicycles is ISO 4210-1:2023, 4210 meaning cycles, the 1 meaning part 1 and 2023 meaning the year the standard was published.

There are other standards such as Underwriters Laboratories Solutions (UL). For this project we won’t need to but if we wish to have certification will need to abide by standards set by UL. For e-bikes the certification is UL 2849.

Both UL and ISO create standards for the same thing so a product may have multiple certifications, however, while they may have the certifications the design must comply with the government which all manufactures must comply with.

Design constraints are set by the governmental body of Consumer Product Safety Commission (CPSC). The CPSC put a label on what a bicycle is defined as and different requirements for a bicycle that importers and manufacturers must abide by. For bicycles, including e-bikes, manufacturers must ensure they follow 16 Code of Federal Regulations (CFR) Part 1512.

4.1 Battery Standards/Regulations

There is a difference between standards and regulations in the sense that standards are open to interpretation and may change depending on which regulatory body you go with. Regulations are set by the federal government and must always be followed.

4.1.1 Battery Standards

Current lithium-ion battery standards can be recommended by a few government bodies including CPSC but also United States Agency for International Development (USAID). For CPSC, the staff is currently participating in these standards:

- ANSI/CAN/UL 2272 – Electrical Systems for Personal E-Mobility Devices
- ANSI/NEMA C18 – Safety Standards for Primary, Secondary and Lithium Batteries

- ASTM F2951 – Standard Consumer Safety Specification for Baby Monitors
- ASTM F963 – Standard Consumer Safety Specification for Toy Safety
- IEEE 1625 – Standard for Rechargeable Batteries for Multi-Cell Housing
- IEEE 1725 – Standard for Rechargeable Batteries for Mobile Telephone
- UL 1642 – Standard for Safety for Lithium Batteries
- UL 2054 – Standard for Household and Commercial Batteries
- UL 2056 – Outline of Investigation for Safety of Power Banks
- UL 2595 – Standard for Safety for General Requirements for Battery-Powered Appliances
- UL 4200A – Standard for Safety for Products that Incorporate Button or Coin Cell Batteries Using Lithium Technologies
- UL 60065 – Standard for Audio, Video, and Similar Electronic Apparatus-Safety Requirements

The USAID have a set of standards that should be references as well, some of them are different and will be posted below:

- IEC 61960 – Secondary cells and batteries containing alkaline or other non-acid electrolytes – Secondary lithium cells and batteries for portable applications – Part 3: Prismatic and cylindrical lithium secondary cells and batteries made from them
- IEC 62133-2:2017 – Safety cells and batteries containing alkaline or other electrolytes – Safety requirements for portable sealed secondary lithium cells, and for batteries made from them, for use in portable applications – Part 2: Lithium systems
- IEC 61959:2004 – Secondary cells and batteries containing alkaline or other non-acid electrolytes – Mechanical tests for sealed tests for sealed portable secondary cells and batteries
- UL 9540, 2nd edition – ANSI/CAN/UL Standard for Energy Systems and Equipment
- UL 9540, 4th edition – ANSI/CAN/UL Standard for Test method for Evaluating Thermal Runaway Fire Propagation in Battery Energy Storage Systems
- UL 1973, 2nd edition – ANSI/CAN/UL Standard for Use in Stationary, Vehicle Auxiliary Power and Light Electric Rail (LER) Applications
- UL 1974, 1st edition – ANSI/CAN/UL Standard for Evaluation for Repurposing Batteries
- JIS C 8715-2 – Secondary Lithium Cells and Batteries for Use in Industrial Applications – Part 2: Tests and requirements of safety

4.1.2 Battery Regulations

Next will be the federal regulations for the transportation of lithium-ion batteries can be viewed in Title 49, subtitle B, chapter 1, subchapter C, part 173, 173.185. These are regulations that apply to nearly all aspects of the battery including: classification, packaging, exceptions, air transportation, batteries sent for disposal or recycling, low production runs and prototypes, damaged batteries, exceptions to air travel and approval.

All the batteries must meet the standard set in the UN Manual of Tests and Criteria. Included in the regulations is what the manufacture must do such as maintain a record of the test report as long as they are transporting it plus one year and need to be available to all forms of government.

There appears to be little no information about how to make the battery and there are minimum safety regulations about how to use batteries and is limited to medical devices. There are also smaller references to batteries across a few different titles in CFR. In title 30, chapter 1, Subchapter O, Part 74, Subchapter C, 74.7 which is about mineral resources there are regulations about the length of time the battery must last (12 hours) and the charger which must operate from a 110 (VAC) (nominal), 60 Hz power line. There is also a reference to if the battery is rechargeable or not. Stating "If the CPDM (Continuous Personal Dust Monitor) uses a rechargeable battery, the CPDM shall have a feature to indicate to the user that the device is sufficiently charges to operate and provide accurate measurements for an entire shift of 12 hours under normal conditions of use".

4.1.3 Motor Regulations

In terms of the motor there are not many regulations regarding it but in many states, they are restricted to stay under 750 watts. This fits our state law since we are located in the state of Florida and we can be considered owners of a Class 2 Electric Bicycle.

4.2 Bicycle Standards/Regulations

As stated above the standards for bicycles are maintained by ISO which is what ANSI uses. For the safety of cycles, there are many different parts of the standard. Most are the methods used when testing components.

4.2.1 International Standards for Bicycles

4.2.1.1 Lighting standards

- ISO 6742:2015 – Lighting and retro-reflective devices
- ISO/DIS 6942-1 – Lighting and retro-reflective devices

4.2.1.2 Safety requirements for electrically power assisted cycles.

- ISO 4210 – Safety requirements for bicycles

4.2.2 Federal Regulations for Bicycles

- U.S. Consumer Product Safety Commission's Code of Federal Regulations (CFR), Title 16, Part 1512 – Requirements for Bicycles

4.2.3 State Regulations for Bicycles

- Florida Statutes, Title XXIII, Chapter 316.003 (4) – Bicycle Definition
- Florida Statues, Title XXIII, Chapter 316.2065 – Bicycle Regulations
    - Chapter 316.2065 (7) – Requirements for lighting
- Florida Statues, Title XXIII, Chapter 316.20655 – Electric Bicycle Regulations

Class of Electric Bicycle	**State Regulations (Fl)**
Class 1	A bicycle with a motor that only assists when the rider is pedaling. Then when reaching approximately 20 miles per hour will stop.
Class 2	A system in which the motor contains a throttle-actuator and stops helping the rider once the speed of about 20 miles per hour is met.
Class 3	A bicycle with a motor that only assists when the rider is pedaling. Then when reaching approximately 28 miles per hour will stop.

Table 23: State of Florida Ebike Class Regulations

4.3 Interface regulation

4.3.1 SWD

Known as serial wire debug is a 2-pin interface that is an alternative to JTAG. SWD uses an ARM CPU standard bi-directional wire protocol. This protocol is also designed for debugging so for that purpose this would be a better choice.

4.3.2 JTAG

Joint Test Action Group who codified it is a 4-pin interface. Generally, uses a daisy chain for its data which means that its slowest chip is where the limit is. There is a 2-pin JTAG, however, it is not widely used today. JTAG is designed for chip and board testing to check for faulty chips and/or boards. JTAG is also able to be used on non-ARM architectures meaning it will be more widely supported as opposed to SWD. JTAG does this by connecting to a test access pin that allows access to a set of test registers that show chip logic levels and device capabilities.

4.4 Communication interfaces

4.4.1 UART

Is a device that allows asynchronous serial communication. It accomplishes this by sending the data bits one by one through a line where there is a start and stop bit to know what information is being sent. For this to work a clock generator send signals to two shift registers, each going in an opposite direction. The leaving register is labeled as TX and receiving is labeled as RX. There is a transmit register that goes to the TX shift register and a receiver register that gets information from the RX shift register. The communication has three different modes, simplex, half duplex and full duplex. Simplex means that data is only sent in one direction. Meaning it can be sent and not received or vice versa. Half duplex means that both devices can send and receive data but only one at a time. Full duplex means that both devices can transmit data simultaneously. The use of UART has been decreasing recently in favor of other interfaces such as I^2C and SPI while between computers and peripheral Ethernet and USB is being used. The places it is used is in low speed and low throughput applications. This is beneficial for us because we are not sending large amounts of data to the Bluetooth module. Another positive is the low cost and ease of implementation. The UART frame format is it has start and stop bit, a data bit and an optional parity bit. When being read a high voltage is a one while low is seen as a zero. When idle the line is held at a one. To indicate the beginning of data transmission the line goes to low from idle then immediately afterwards is the data. To indicate data is done being sent the line will stay and or return to idle state. Sometimes an additional stop bit is implemented but it appears to be uncommon in practice. Data is sent in 5 to 9 bits, however, usually it is seven or eight bits and will be sent with the least significant bit first. The parity bit is used as error detection. This is not foolproof because it can only detect a single flipped bit.

4.4.2 I^{2}C

I2C is a synchronous protocol which has a master that initiates communication with a slave device. The master provides the clock which is the data transfer rate or clock frequency. The bus is bidirectional unless the speed is at 5 MHz in which case it becomes unidirectional. This means (when bidirectional) the master can write to the slave and read from the slave. The bus is also in serial so the data is shifted bit by bit. The two buses is SCL for serial clock and SDL for serial data. As mentioned earlier about the speed the most common speeds are 100 KHz, 400 KHz, and 1.0 MHz. The ones that need special consideration is the above 5MHz and 3.4 MHz. The 3.4MHz speed is still bidirectional like the 100, 400 and 1000 KHz frequencies. A benefit of I2C as opposed to UART is there can be multiple slaves to a master as opposed to UART where there is only two connections between devices. The start condition is similar to UART where the SDA goes low, however, SCL remains high. Immediately after follows the slave address which can be seven to 10 bits long, however the most common is seven bits. After the address is the read or write bit. All the previous bits make up the control byte. Once the information is sent every ninth clock cycle is the acknowledge bit. The device receiving data is the device that generated this bit. After this is the data byte which is equivalent to how UART does this. Then there is the stop condition where both signals go to high.

4.5 Testing Certifications

These are items that require a certification to work with.

4.5.1 ANSI/ESD S20.20-2021: Protection of Electrical and Electronic Parts

According to ESD ADV1.0-2017 electrostatic discharge is "the rapid, spontaneous transfer of electrostatic charge induced by a high electrostatic field. Note: Usually, the charge flows through a spark between two conductive bodies at different electrostatic potentials as they approach one another".

This certification outlines the development of electrostatic discharge (ESD) programs to protect components that may be sensitive or at risk of damage to ESD. One item that is sensitive to ESD includes microcontrollers, which is something we will be working on for our project. Other items that are sensitive to ESD include integrated circuits and printed circuit boards.

The reason for bringing this up is because the human body is a source of ESD and should be taken into account when working with these sensitive devices, especially when that are not connected to ground. To endure no damage as a result of ESD a device that discharges the body to ground should be used as well as having materials such as rubber matting in working areas to minimize the risk to the above-mentioned components.

4.6 Economic Constraints and Time Constraints

The cost of this system is a constraint that we would like to take into account to ensure that what we do buy, if it breaks we can replace and so if other decide to copy what we do it is cheaper than an average conversion kit that can be bought online. The most expensive components of the bicycle (other than the bike itself which is already owned) will be the battery and the motor. Their higher cost somewhat allows them to stay in stock because less people will want to buy them because of the cost. The global shortage of chips does no appear to still be in effect as of right now but time will tell as we get further into the project if this ends up being a problem.

Time constraints may be a problem with this project, the main battery that is chosen will take only a few days to ship, however, if we go with a much cheaper option that changes to two months to ship. In the end the group will have to decide which one would be preferred. Seeing that the time this project takes will only be a few months as opposed to other project at companies which may take up to a year and a half we must work efficiently and quickly for this project in order to meet the deadline for Senior Design 2. Us getting the components early will allow the group to begin the testing process for the project earlier which lets us figure out any problems and trouble shoot early.

4.7 Environmental and Health Constraints

Currently there is a large push for everything to go electric and to decrease the amount of fossil fuel use. This e-bike will be purely electric and if the user chooses to do so can ignore the motor part of the e-bike and pedal like normal. The highest power draw will be the motor at a maximum of 350 watts, but this can be reduced if the stretch goal of assist mode we have planned is implemented. The most damaging component to the environment of the e-bike is most likely the lithium-ion battery pack because of the waste products they come out as a result of the production of the cells.

Safety is also very important to everyone here and we will attempt to follow safety procedures whenever possible. For example, following ANSI/ESD S20 and only buying from companies where safety is a high priority such as the battery pack that has a battery management system to help prevent short circuits and to make sure the battery temperature is regulated.

5.0 Hardware and Software Design

5.1 Initial Design Architecture

PCB

This is the WIP block diagram for the PCB. Each block is its own small system that will be connected to each other. The arrows represent a connection that will be made to another part on the board. You can see there is quite a few connections between the components on the PCB. In the images following the block diagram are the schematics that are being created in EasyEDA. There are also connectors used to interface with the board. The block diagram was created in OneNote because of its ease of use and the ability to quickly move around the blocks and draw arrows to represent the flow of information or power between components. The PCB will have six inputs/outputs which will be a 36v (nominal) lithium-ion battery, a motor, a throttle, an LED strip, e-brakes, and a switch that has four buttons on it to have an alternate input to the PCB if for some reason the phone runs out of battery or some other unforeseen issue arises. The battery will need two inputs onto the PCB. The motor will have nine pins that will need to be connected to the PCB for phase (x3), three hall signals, hall ground, hall 5v and a speedometer that is built into the motor. The switch will have four buttons but have 5 pins. Each button can do whatever we wish it to do while the last pin will be a ground pin. The throttle will have three pins, one being the throttle, one being 5v and the other being ground. The e-brakes have two connectors both of which are switches.

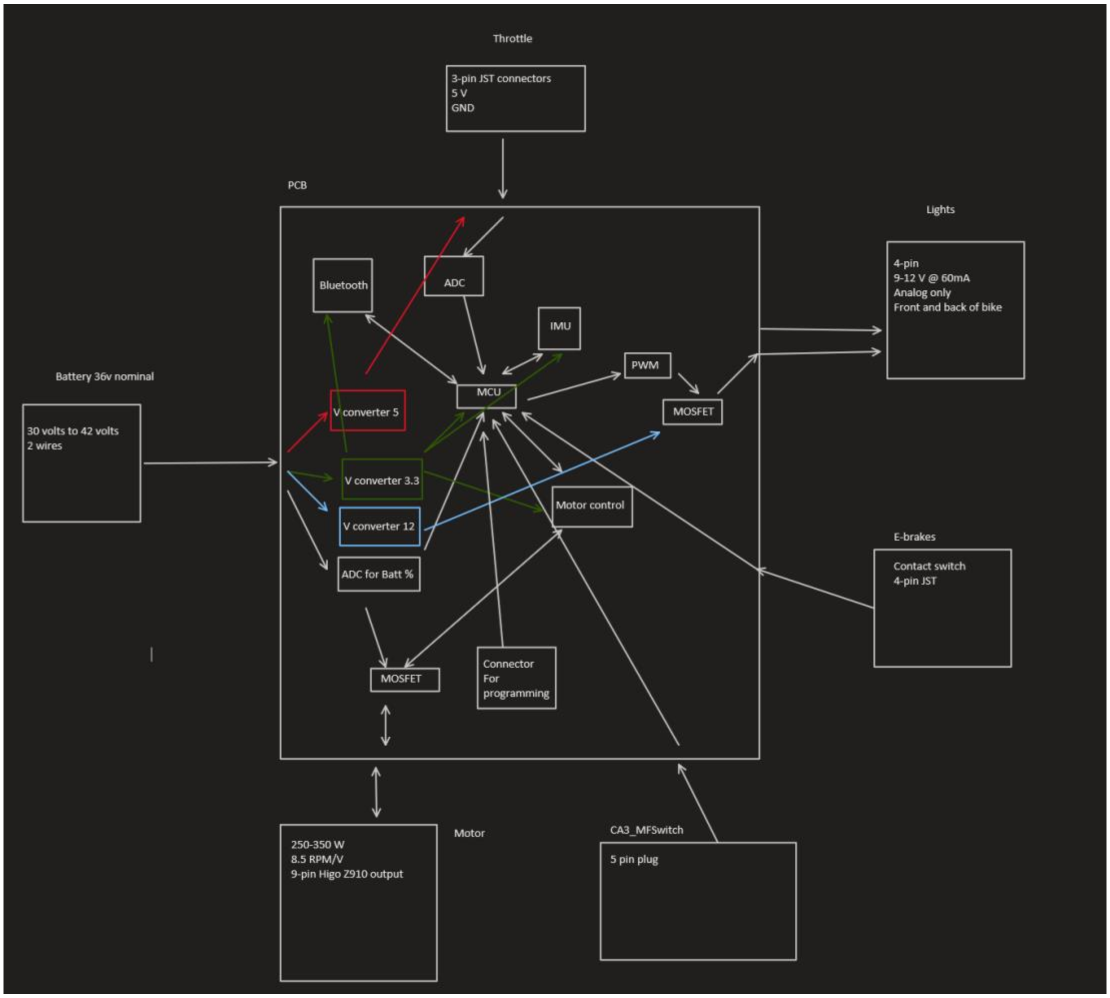

Figure 37: Block diagram of schematic

5.2 Schematics

Below are the schematics that will show the logic that the board will have between components. These will continuously be updated as the project progresses through iterations in senior design 2 due to failed parts or components that may not be able to be used for our purpose. So while the idea for the project is here this is in no way the actual final product that may end up being wildly different than what is here on the paper now. Schematics are being created using EasyEDA again and because they already have their own library so there is no reason to have to search for the part and import it as we had to do with Fusion 360 Eagle from Autodesk in junior design.

External components

There are also other inputs onto the board that will all need to be connected to the MCU. One such item is the switch and below will be an example of some external parts such as the throttle connecting to the PCB and MCU.

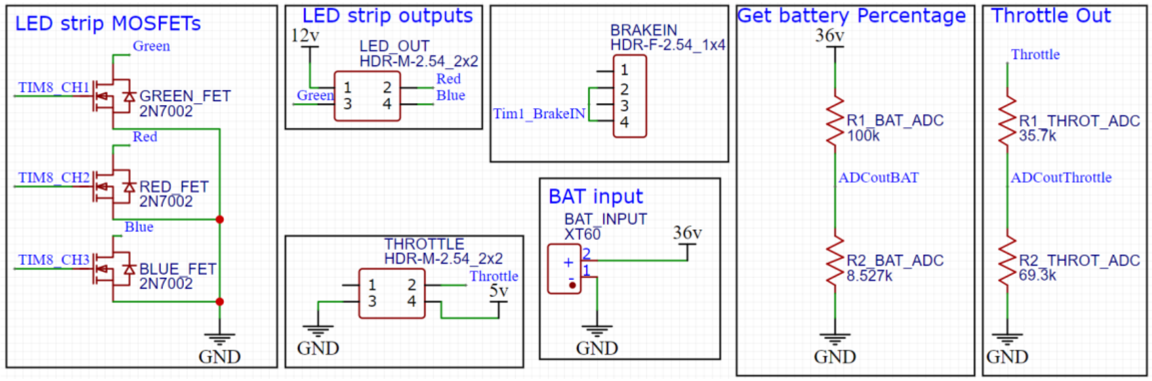

Figure 38: Example of peripherals

These are still subject to change as we continue to make progress in the project and as always is subject to change at any time. Hopefully there will be minimum changes to the future design, however, everything is subject to change.

The peripherals not shown in the picture above include the switch and a probe for ground incase it is needed. There is also a port for the speedometer to go directly to the MCU.

MCU

We will utilize the MCU to speak to and control the other components on the board. We will primarily be using the USART protocol as communication for the Bluetooth and the IMU. The motor controller is still being worked on as of the time of this writing.

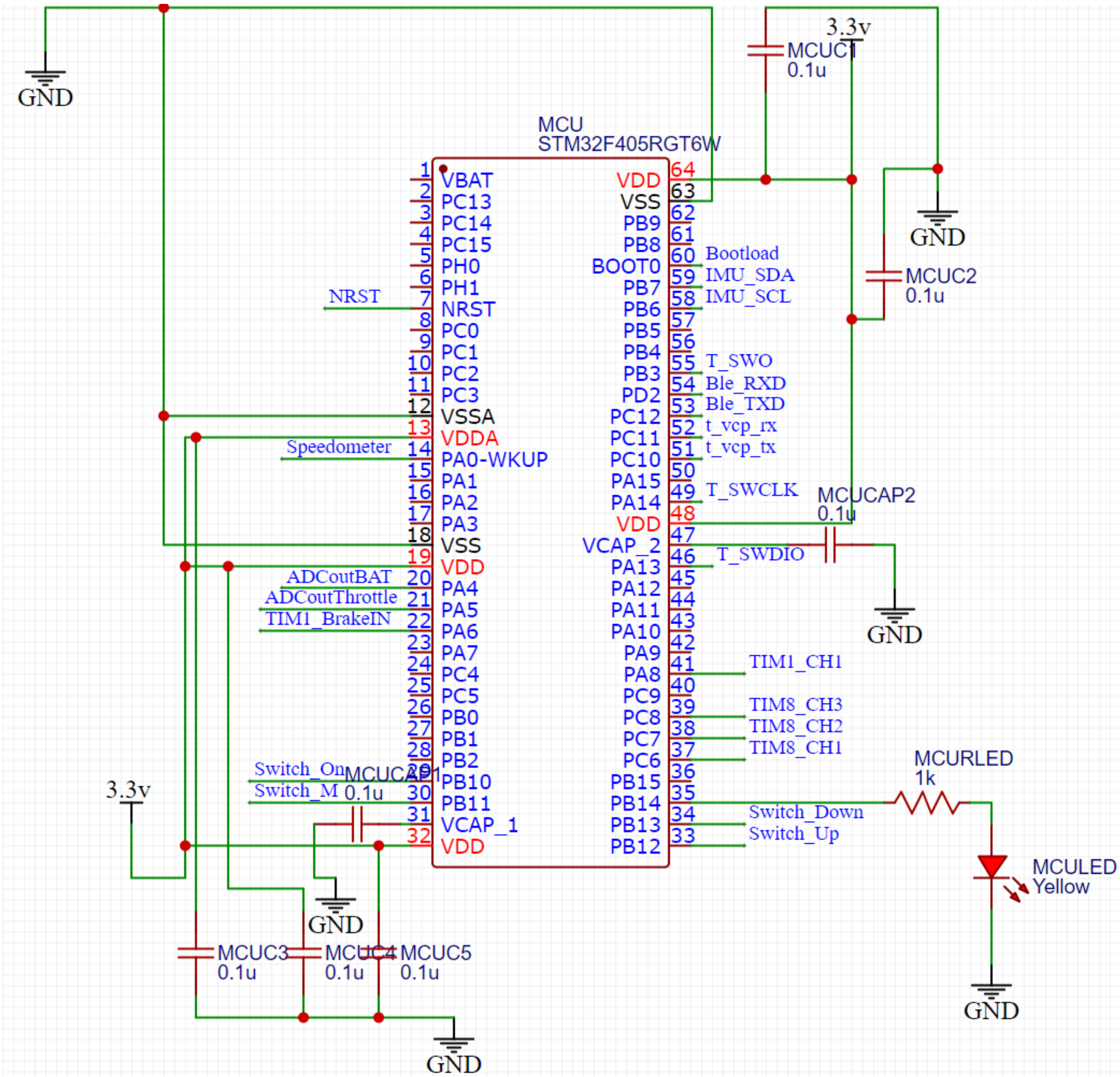

Figure 39: MCU connections

Above is the schematic of the MCU which will be the brains of the operation on the PCB. It has an input voltage of 3.3 which we well get from the buck converters.

Bluetooth module

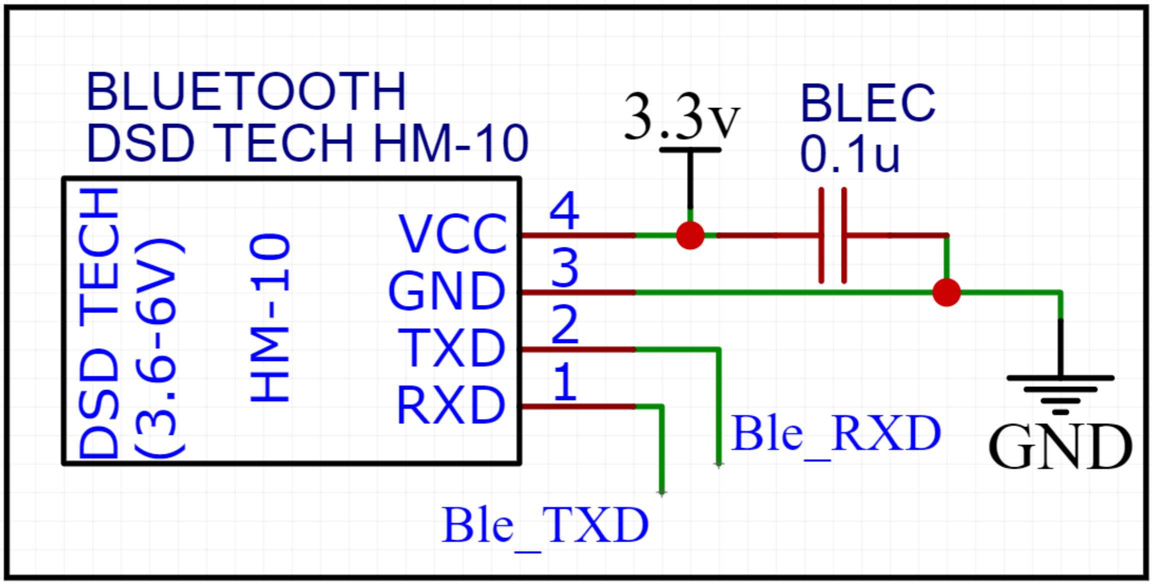

Figure 40: Bluetooth connections

Here is the Bluetooth module which will use the UART protocol to speak between the Bluetooth and the MCU. The purpose of using UART is to enable a much larger amount of information to be transferred instead of using the pins as GPIO which can only have two commands per pin.

This Bluetooth module has changed since the last iteration and this one is used instead which will be written about in the challenges section.

Driver

The motor driver will be used to control the motor while using the MOSFETs because of the continuous current that they are capable of transferring.

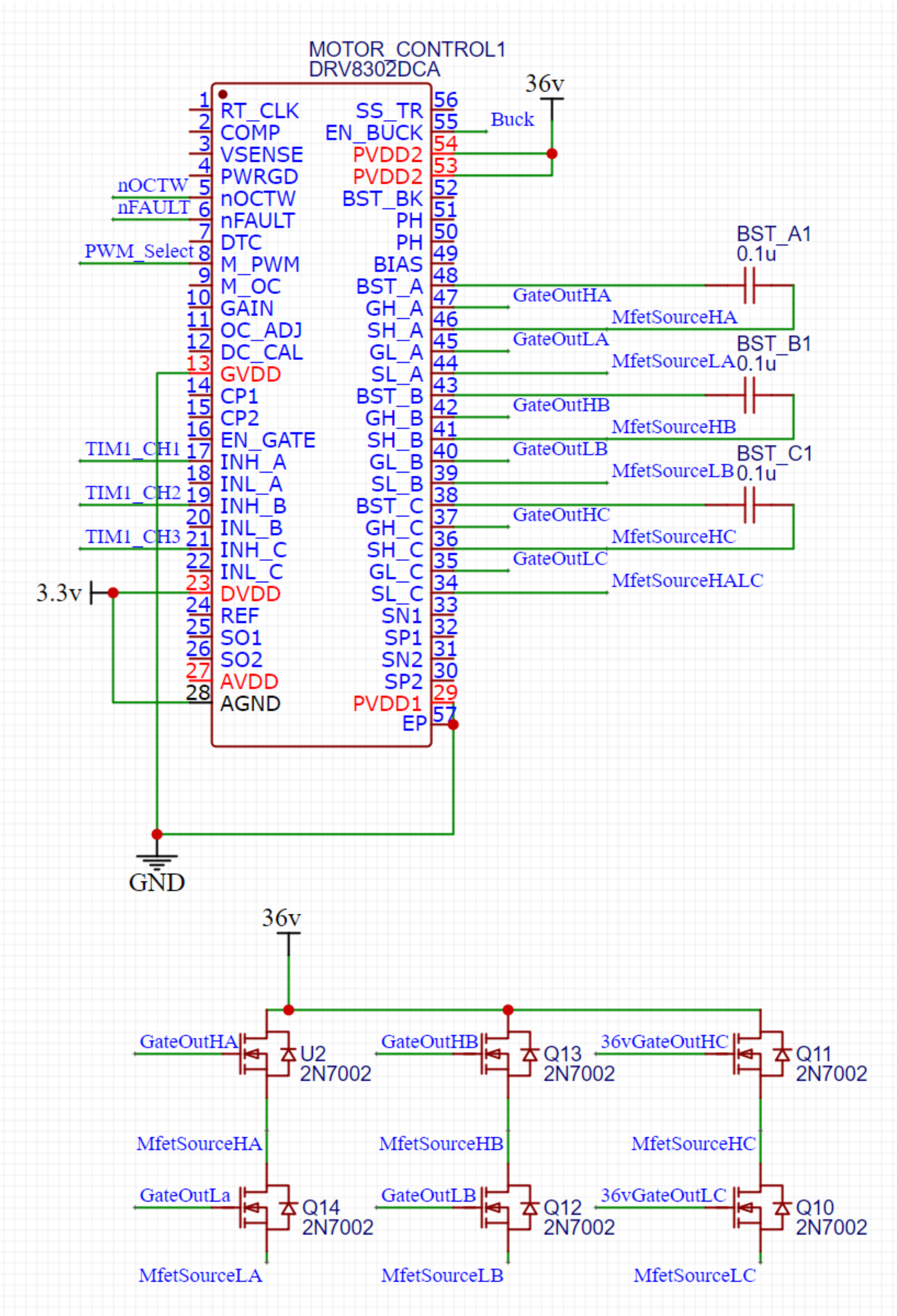

Figure 41: Motor control connections

IMU

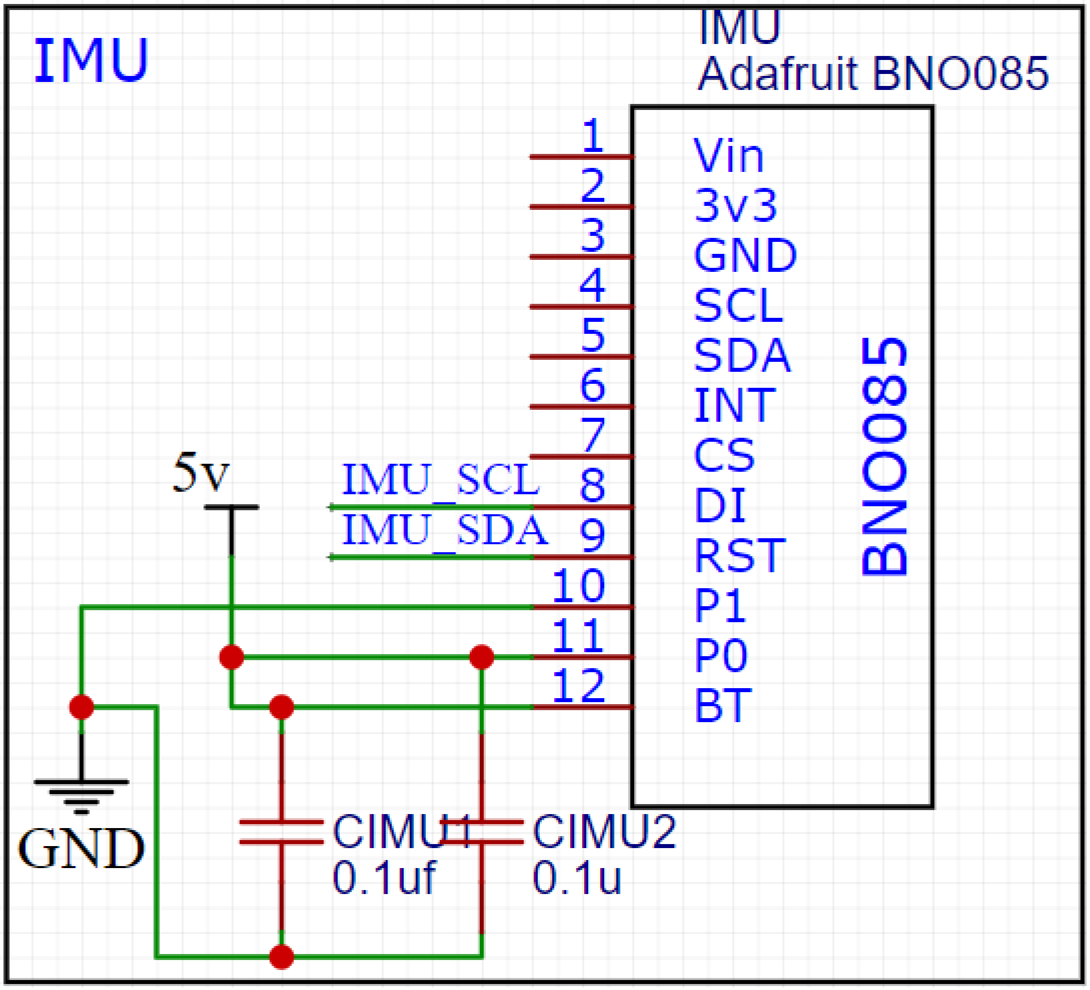

Figure 42: IMU connections

The IMU will be able to give us information about the orientation of the bicycle which will hopefully be used to force stop the motor in the event that a rider falls off the bicycle.

Flash device that will be used

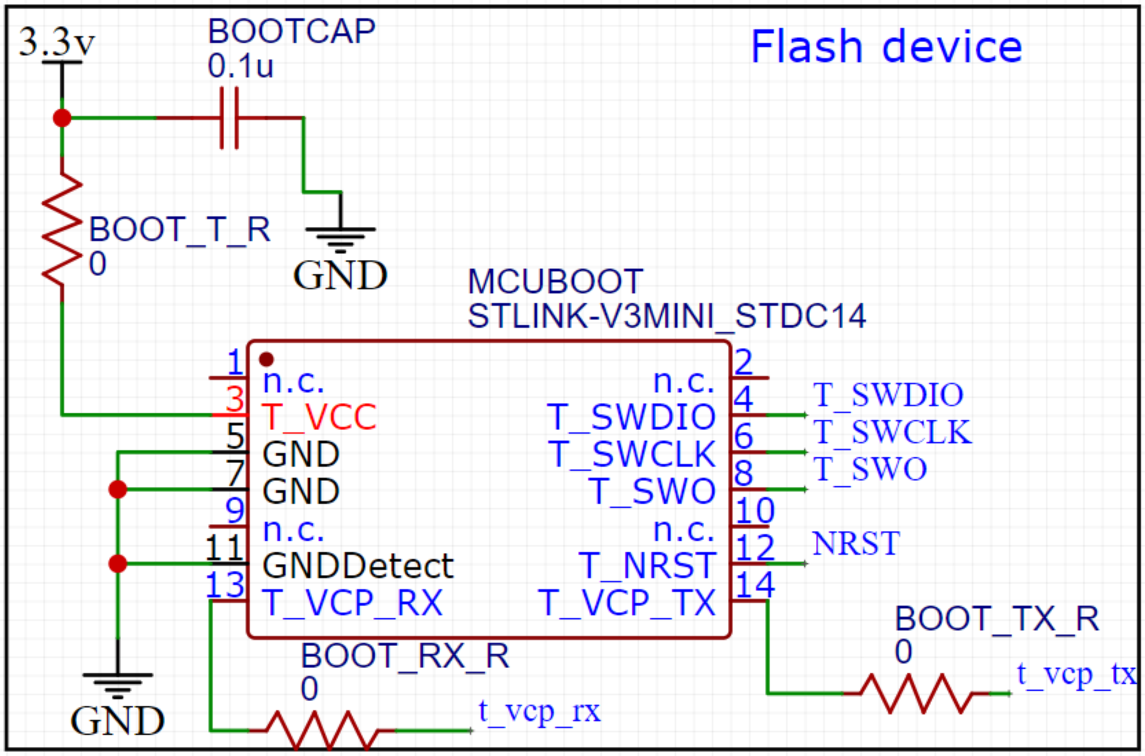

Figure 43: Boot schematic

The connector is an STDC14-pinm connector that is used to program STM electronics. This will be the connector used to code the MCU while it is soldered on the board so if there are any updates that are needed it won't need to be taken off the PCB the resoldered

DC/DC 12 volts @2A max volts for the lights

Option 1:

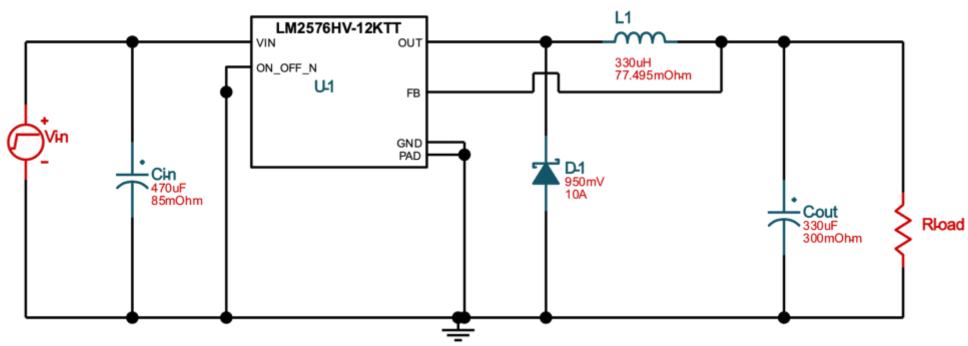

Figure 44: Option 1 12v

Above will make use of the LM2576HV-12 which has the lowest BOM count at 5 but has a higher cost of $11.63. The efficiency is fairly high as well at 89.8% but has a large footprint at 1577 mm squared. This device can handle an input of 4v to 60v and output at a max of 3 amps. This will be useful for the lights as they will need a large amount of current in order to get to max brightness.

Option 2:

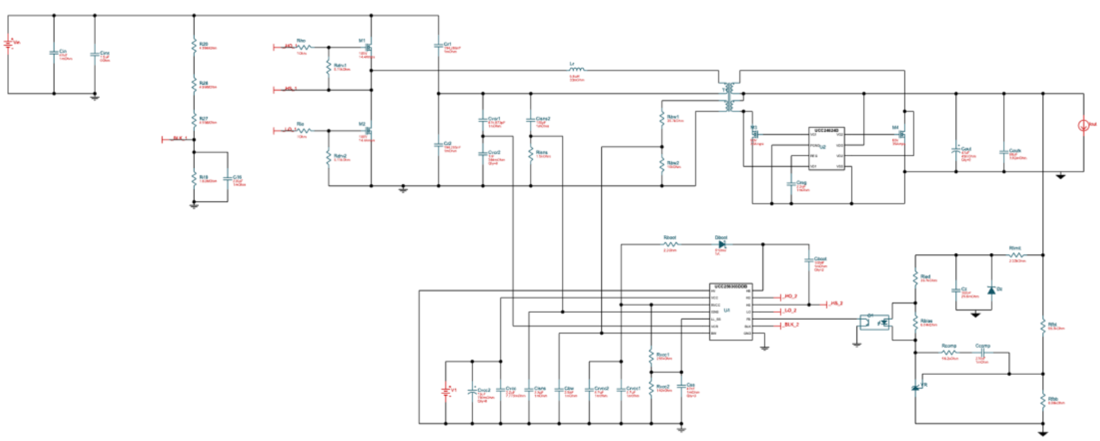

Figure 45: Option 2 12v

Above will make use of the UCC256303-UCC24624 to achieve an efficiency rating of 96.8%. The circuit is so large that the BOM cost and footprint are NA. The BOM count is at 71. This one will most likely not be used due to the large amount of time that would have to go into soldering and ensuring that all parts are indeed working. This would only be used in applications where energy saving are paramount to the operation. This is not the case here because while we do want the battery to last for as long as possible it is possible to still ride the bike in a normal fashion.

Option 3:

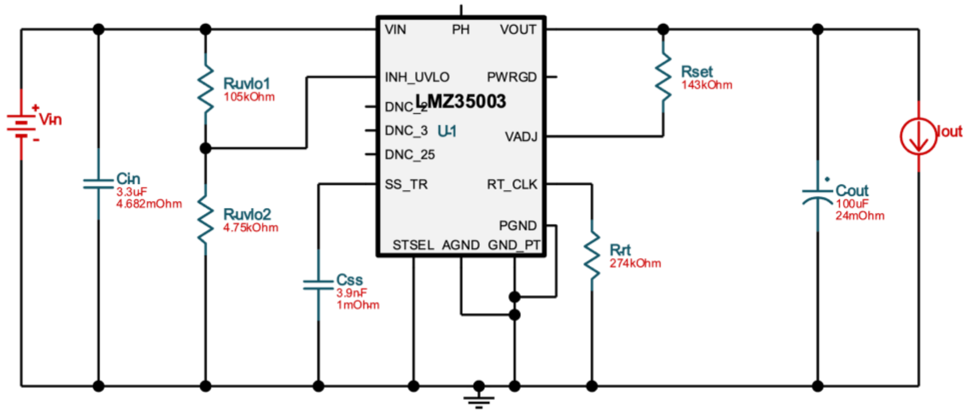

Figure 46: Option 3 12v

This design makes use of the LMZ35003 to go for the smallest footprint at 280 mm squared. The BOM count sits at a respectable 8 while the cost is at $8.67. The efficiency is not that bad either being at 88.7% nominal.

	Option 1	Option 2	Option 3
Nominal efficiency	89.8%	96.8%	88.7%
BOM cost	$11.63	NA	$8.67
Footprint in mm^2	1577	NA	280
BOM count	5	71	8

Table 24: 12v DC/DC

There are three designs that we could pick from but option 2 has too many components for us to be comfortable to use on the project because of the parts that could be wrong or a missing connection that could cause a failure in the circuit. That leaves us with option 1 or 3. Each has their own positives and negatives. Option 1 has the lowest BOM count, however, it does have a fairly higher price but fewer parts to break or mess up. This one is also similar to a design seen further in the paper so it would most likely be the best choice because of familiarity with specific components.

The lights will require MOSFET to power and an optional PWM for color changing. This will allow us to have different colors for the front and back. Like brake lights and a headlight for the front. This will also enable us to have the app flash the LEDs to ensure that the lights are working properly before you rise.

DC/DC 5 volts for throttle

Option 1:

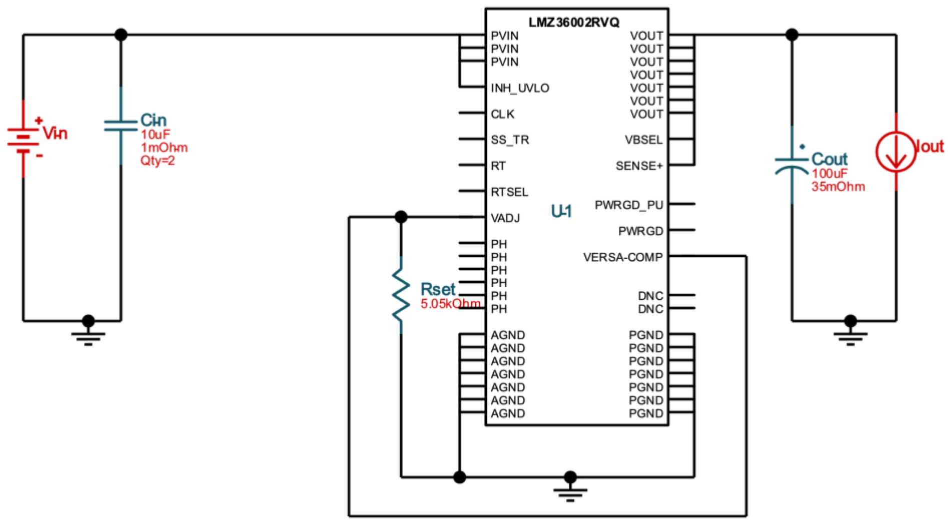

Figure 47: Option 1 5v

This DC/DC design uses the LMZ36002, this one has the lowest BOM count at 5 with a cost at around $5.26. The footprint sits at 197 mm squared and has a nominal efficiency of 77.8%.

Option 2:

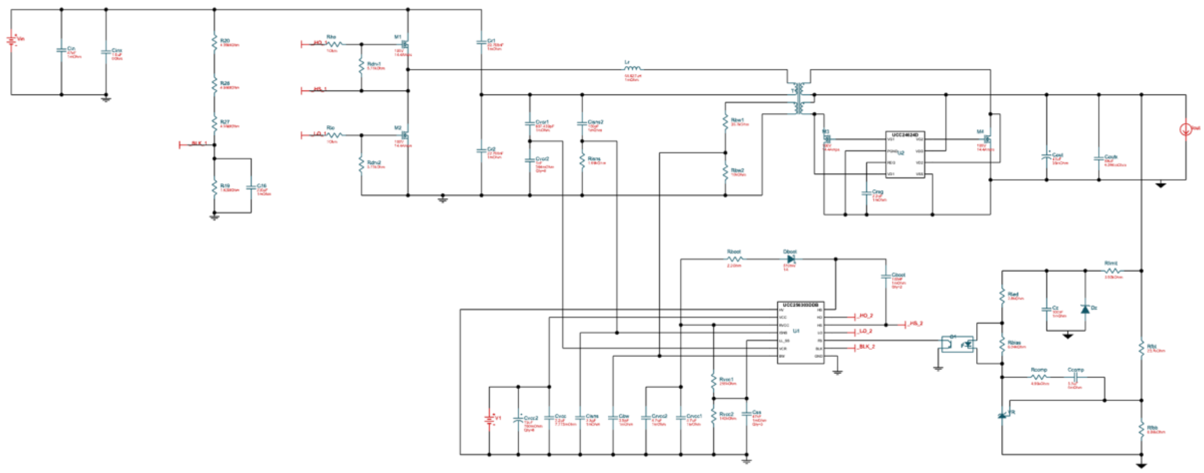

Figure 48: Option 2 5v

This DC/DC design uses the UCC256303-UCC2464. This design has the highest efficiency at 92.5%. It does, however, have a very large BOM count at 70 pieces which is far too many to work with and would increase the soldering time drastically. The BOM cost and footprint are unavailable because of the variability of this circuit. This design is similar to what was found in the 12v section for option 2 and there is probably some sort of correlation that could be investigated at some point in the future because it is curious as to why they look the same but output at different voltages.

Option 3:

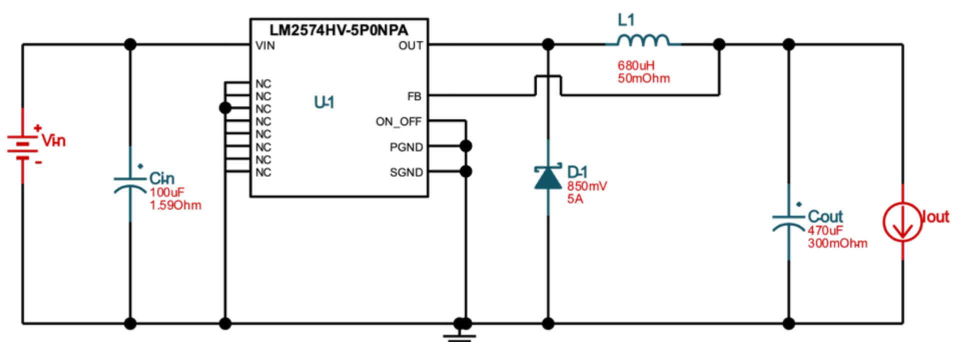

Figure 49: Option 3 5v

Here is the TPSM365R6V5. This has the smallest footprint at 76 mm^2 which is the lowest at the specs that we want. The nominal efficiency of the circuit is 69.9% and has a BOM count at 8 with a cost of $3.02.

	Option 1	Option 2	Option 3
Nominal efficiency	77.8%	92.5%	69.9%
BOM cost	$5.26	NA	$3.02
Footprint in mm^2	197	NA	76
BOM count	5	70	8

Table 25: 5v DC/DC

In this case option 2 is obviously out because of the large BOM count at 70 parts. That leaves our options for either one or two. In this case believe that option 3 is our best choice because of not having a singular large component when instead there could be a few more capacitors that are much smaller and will have a cheaper cost that we may end up breaking at some point.

DC/DC 3.3 volts @ 500mA max for MCU, Bluetooth, IMU, and Motor control

Option 1:

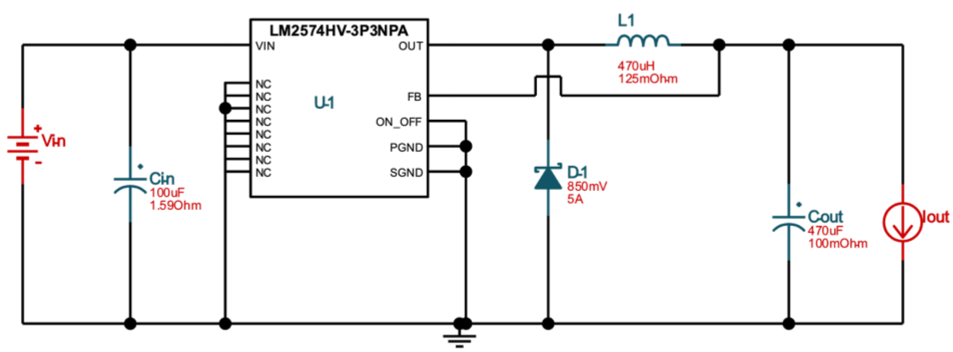

Figure 50: Option 1 3.3v

This DC/DC design uses the LM2574HV-3.3, this one is chosen primarily because of the low BOM count at 5 which will have a generally lower cost at 5.42 dollars. The footprint for this circuit is higher than the others at 1,566 mm^2 but also decrease the amount of parts they we may break or lose during the prototyping process. The efficiency of this circuit is at 67.1%.

Option 2:

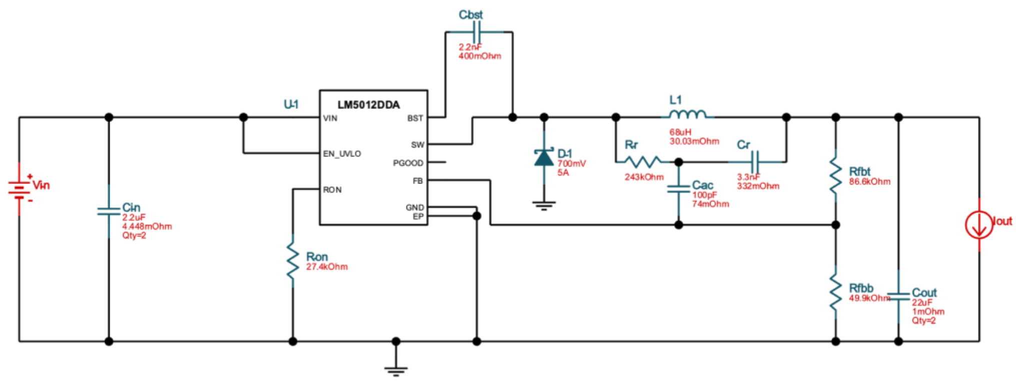

Figure 51: Option 2 3.3v

This DC/DC design uses the LM5012DDA, the reason for this option is because of the high efficiency at 81.3% nominal but has a negative of costing more at 6.63 dollars and a larger BOM count of 14. The footprint for this circuit is much smaller than the lowest BOM count one at 795 mm^2.

Option 3:

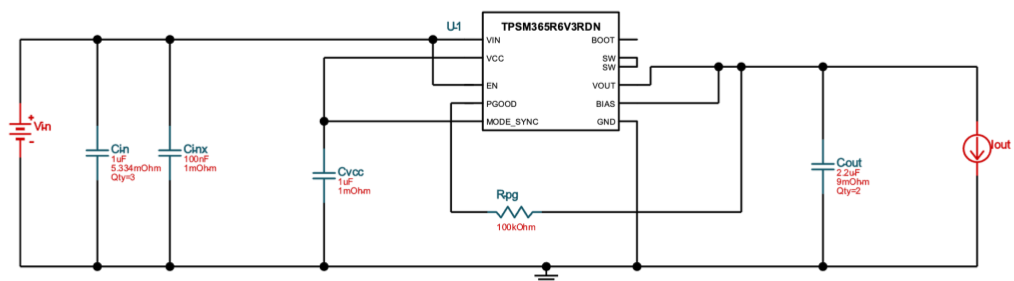

Figure 52: Option 3 3.3v

This DC/DC design uses the TPSM365R6V3, the reason for the consideration is the lower footprint at only 81 mm^2 which is the lowest of all the 3.3-volt options. The BOM count is in between the two previous ones at 9 but has the lowest cost at 3.03 dollar. The efficiency is also in between at 78.2%.

	Option 1	Option 2	Option 3
Nominal efficiency	67.1%	81.3%	78.2%
BOM cost	$5.42	$6.63	$3.03
Footprint in mm^2	1,566	795	81

BOM count	5	14	9

Table 26: 3.3v DC/DC

The group decided to go with option 1 because of the lower BOM count. We have heard that soldering can take quite a long time and by having a lower count of parts to solder this will save us time to work on other parts of the project, such as doing testing and debugging of the project, which we believe will be most of senior design two. It also has similar components as stated earlier in the paper with the circuit that

5.3 Software

The software required for this project to function can be found in two main places: the microcontroller unit (MCU) and the smartphone application being developed. The software in the MCU will handle the PCB operations: regulating speed, Bluetooth connection, etc. The software found in the smartphone application will be dedicated to just that: the app. The application and the PCB will communicate via Bluetooth and will be capable of sending/receiving information to and from the e-bike.

5.3.1 MCU Software Design

To develop and write the firmware for the MCU, STMicroelectronics (STM) provides an IDE called STM32CubeIDE. It is free software that integrates everything needed for development and deployment of code. The IDE provides code generation, hardware configuration, integrating libraries, and debugging features. Using the IDE peripherals, pins, and internal timers can be configured and skeleton code is generated based on these selections.

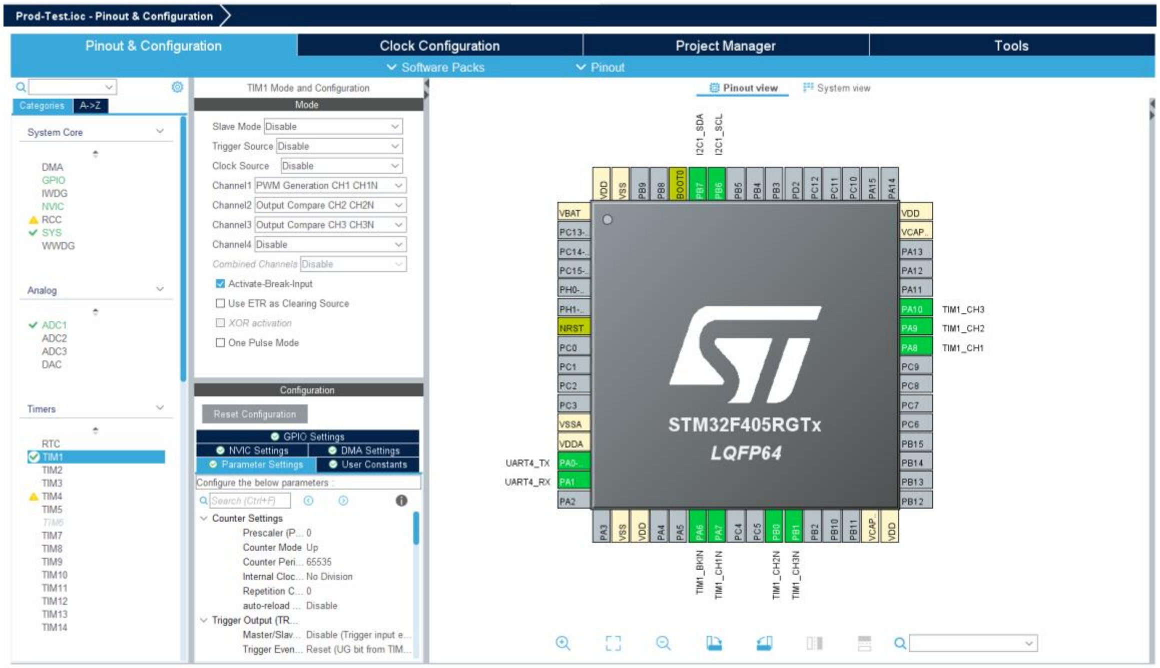

Figure 53-STM32CubeIDE Hardware Configuration (copyright pending)

For the E-bike the overall software architecture will be based on an RTOS instead of super-loop. A version of FREERTOS is provided by the IDE using the Cortex Microcontroller Software Interface Standard (CMSIS) v2 API. The CMSIS library can be used for all things related to the kernel, thread management, semaphores, mutexes, timers, and event flags. The following image shows the structure of how the ARM CPU interfaces with external APIs.

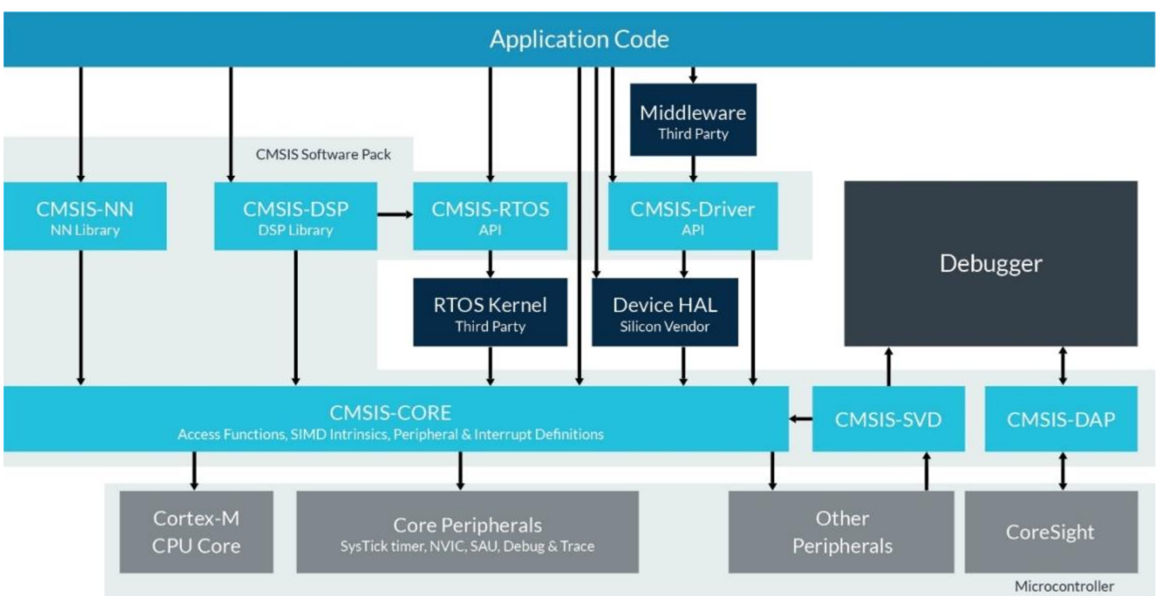

Figure 54- ARM CMSIS Interface (copyright pending)

Based on the image, the code we will write will be the application code that will use the CMSIS-RTOS API to communicate with FREERTOS a 3rd party RTOS Kernel. Along with the CMSIS API we will be using the STM32 Hardware Abstraction Layer (HAL) libraries. The HAL libraries are provided by STM to simplify development by abstracting low-level hardware interfaces and providing a common functionality that is portable across multiple STM32 MCUs. This will allow us to use the *STM32F407G-DISC1* development board. Although this board features a *STM32F407* which is different than our chosen MCU the *STM32F405*. Because of the abstraction provided by the HAL libraries and CMSIS via RTOS we can develop on this board and have it be portable to our final chosen MCU.

There are many features of FREERTOS that are currently not in the plan for use such as mutexes and queues but it is possible they will be added later during development. The library will be used to setup the multiple tasks the MCU will be asked to do and FREERTOS will handle switching between each of the tasks. There are many task priorities the list from highest to lowest priority are: osPriorityRealtime, osPriorityHigh, osPriorityAboveNormal, osPriorityNormal, osPriorityBelowNormal, and osPriorityLow. The task priorities do not include interrupt service routines (ISR)s which will always have the highest priority overall. The planned list of tasks and their expected priorities are listed in the table below.

Task Name	Function	Priority
Motor Control	GPIO & PWM	osPriorityRealtime
Throttle Control	GPIO	osPriorityRealtime
IMU Communication	I2C	ISR or osPriorityAboveNormal
Brake Sensing	GPIO	ISR
Bluetooth Communication	UART	osPriorityHigh
Light Control	GPIO	osPriorityBelowNormal

Table 27 - RTOS Task List

Each task while run sequentially and individually the scheduler of the RTOS will allow them to run as if they were in parallel. The main two focuses are the ability for the MCU to control the motor whether controlled via the throttle or in a cruise control mode. At the same time the MCU will be collecting data from the IMU, motor, battery, and etc. to send data to the app. Also, the app will be sending control flags back to the MCU to determine the mode the bike should be in.

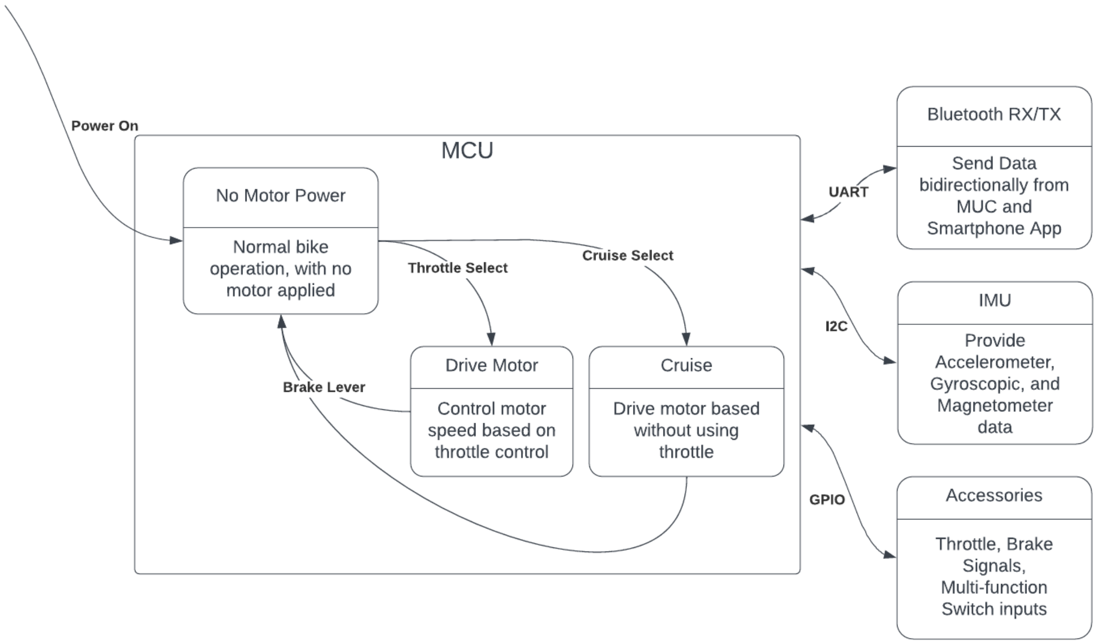

Figure 55- MCU software diagram

5.3.1.1 Proportional Integral Derivative (PID) Control

To implement the motor control using a throttle or a set point speed we will use a Proportional Integral Derivative (PID) controller. PID control is a control method that is widely used in industrial automation and control systems to regulate the behavior of a system. The proportional term (P) is proportional to the difference between the desired setpoint and the actual process variable. The P term is multiplied by a constant gain (Kp) that determines the magnitude of the output signal. The integral term (I) sums the error over time and multiplies it by a constant gain (Ki). The derivative term (D) is proportional to the rate of change of the process variable and multiplies it by a constant gain (Kd). The three terms are combined to form the final output signal using the following equation:

Output signal = Kp * P + Ki * I + Kd * D

The values of the gains (Kp, Ki, Kd) must be properly tuned to achieve the desired control response. An improperly tuned PID controller can lead to instability, oscillations, or overshoots in the controlled system.

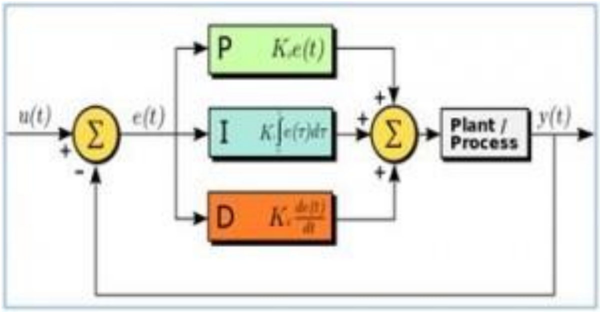

Figure 56-PID Controller (copyright pending)

Although the bike will likely use a PID controller for motor control there are other types of controllers that could be used. The individual components of the PID controller could be used for example using just a Proportional controller or only a Derivative controller. Also, some simpler combinations can be used such as a PI or PD. Ideally the simplest controller would be used that meets the goal of smooth motor control. Designing a full PID controller will give the most flexibility and tuning the system gains can allow the PID controller to act as any simpler subset of controller. PID controllers and their subsets are not the only controllers possible either, lead or lag controllers are just some of the other types. Those controllers are not planned to be used or tested. This is mostly due to the prevalence of PID controllers and their use in industry.

5.3.2 Application Software Design

This section will go into detail about the structure of the smartphone application and its functions. The current build of the application works as intended on both platforms: iOS and Android. Screenshots are provided as a visual aid. Even though the application works on both platforms, the provided screenshots are currently only from the iOS simulator to reduce image clutter.

5.3.2.1 Use Warning

The working design for the application is currently as follows: when running the application on the operator's smartphone, after it is finished loading, it will open to a warning, prompting the user to confirm they are not riding the bike currently to ensure their safety. This warning screen communicates the following message:

"WARNING:

Using this application while operating the e-bike is strongly disadvised.

While it is not illegal, it puts not only yourself in danger, but also those around you.

If currently operating e-bike, please come to a complete stop before continuing.

Thank you for your cooperation."

This exact message is used and is worded precisely as it is above. As mentioned in the message: it is not currently illegal to operate an e-bike and use a smartphone. However, it is strongly advised against due to the smartphone's nature to distract the e-bike operator. The group believes this message is entirely necessary to take steps to prevent operator harm or danger. At the bottom of this message is a button for the user to interact with that confirms that they are, in fact, not in motion. Only after the user confirms there are not in motion via the button on this screen will they be able to access the application and be brought to the Bluetooth screen. A screenshot of the warning message screen is provided below.

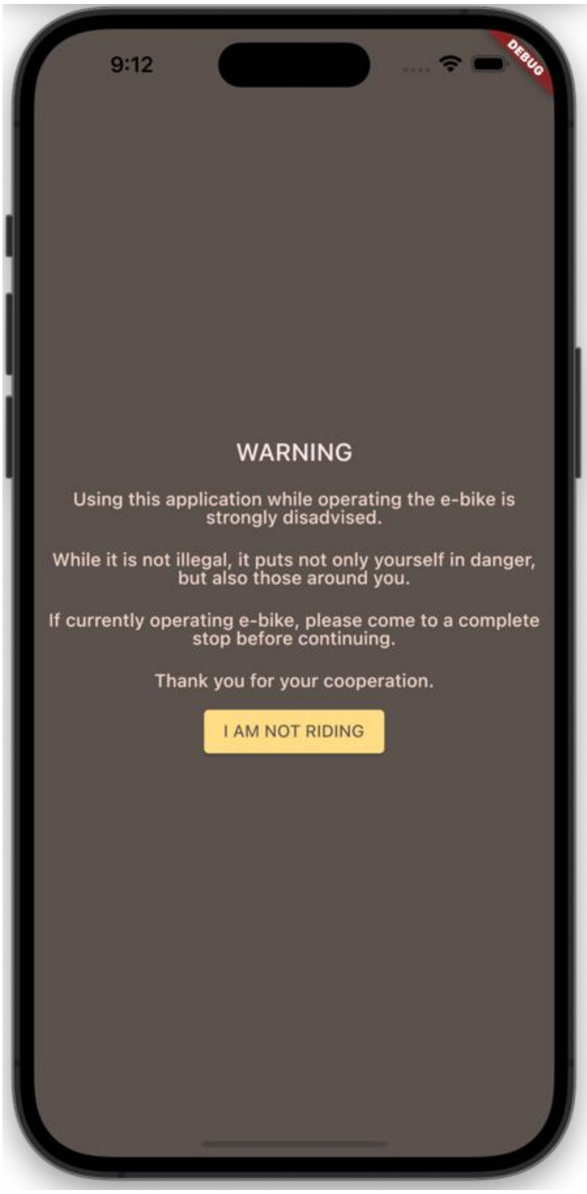

Figure 57: The warning page the app opens to

5.3.2.2 Bluetooth Connectivity Page

The Bluetooth connectivity page will be the first page open after the warning is confirmed. It will contain features to check on, maintain, and test the Bluetooth connection between the smartphone and the e-bike. This page contains a window that will inform the user of the status of the current Bluetooth connection with the e-bike, as well as instructions on how to connect to it. This page also contains a button that, when pressed, directs the user to the smartphone's Bluetooth settings. There were plans to add a feature that would test the connection; however, the connection would be tested on the next screen: the Heads-Up-Display page.

Screenshots of the completed Bluetooth connectivity page are provided below:

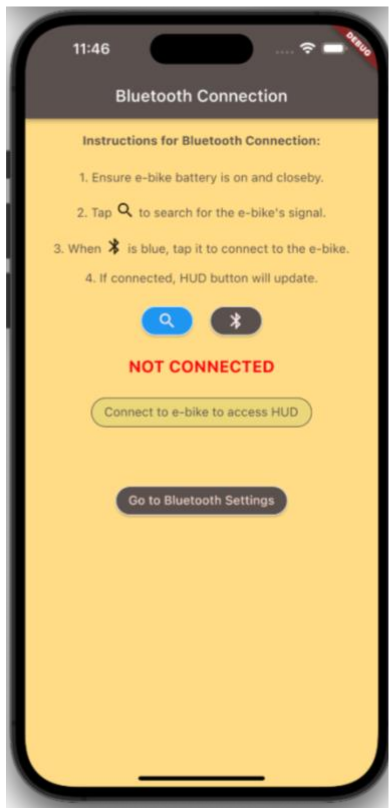

Figure 58: Bluetooth page (not connected)

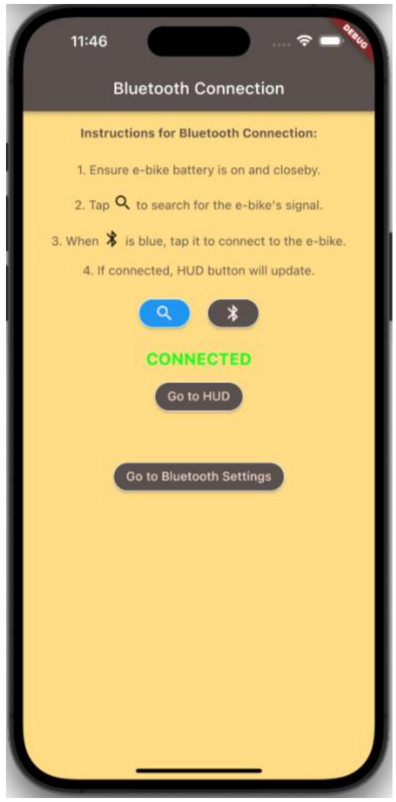

Figure 59: Bluetooth page (connected)

Currently, the "*Go to Bluetooth Settings*" button works correctly on both platforms and takes the user to their platform-respective Bluetooth settings page. Please note that the GIF provided below is old and shows the WIP, non-functioning Bluetooth page.

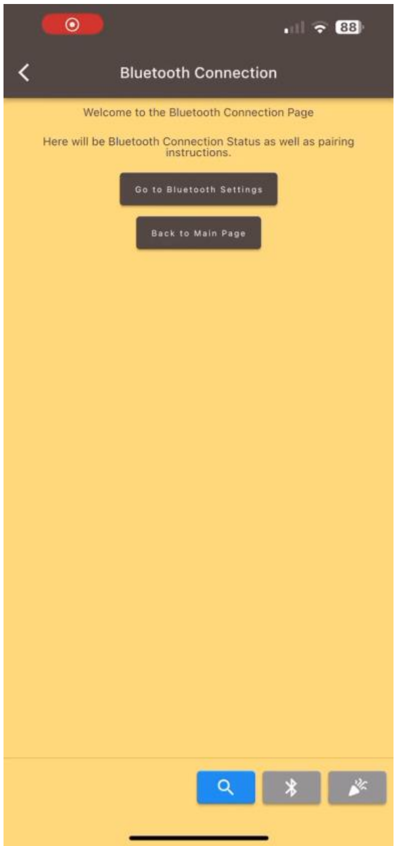

Figure 60: The "Go to Bluetooth Settings" button in action

5.3.2.3 Heads-Up-Display Page

The second option on the home/landing page is the Heads-Up-Display (HUD) page. This page will be exactly what it is titled to be: a page that holds several widgets containing useful information to report to the operator of the e-bike. This page is now complete and functions as intended. This page is built to function primarily with the device in landscape orientation, to provide a larger canvas for the widgets to fit in.

The widgets currently on the HUD page are a bike throttle mode toggle (to turn the throttle on), a speedometer, (to track the speed of the e-bike), a charge meter (to track the charge of the e-bike's battery), and a light control prompt (to control light color). These are a great number of widgets to have on the screen so as to not distract the user with too many. The values for the speedometer and charge meter are tracked by the e-bike's PCB and are sent to the application via Bluetooth at a preset interval and are updated on the application as they are received.

A screenshot of the complete HUD page is provided below.

Figure 61: The WIP HUD page

5.3.2.4 – Light Control

The e-bike will have LED lights mounted on the front of it, acting as a front-facing head lamp. A head lamp for bicycles, in the state of Florida, is required and mandatory if the bicycle is to be operated between sunset and sunrise, as per Florida Statues Title XXIII Chapter 316.2065 (7). Light control was planned to be a separate page with color selections; however, it made more sense to add it to the HUD page in the form of a pop-up. The color under "Light Color" on the HUD page is a button which opens a color selection wheel. The light control page is able to not only flip the state of the head-mounted LED lights (turn them off and on), but also change the color if desired. This is done via a color wheel to change the headlights' color if the operator so desires.

5.7 Summary of Design

5.7.1 – Hardware Summary

For the design of the project, we will have the motor assembled on to the rear tire rim and then attach it to the frame of the bike. This will then be wired directly towards the PCB that will contain the microcontroller, Bluetooth module, and IMU that we will be using. In terms of the PCB, we will try to create or acquire a waterproof enclosure that can be mounted directly on to the frame of the bike. Once this has been done then we will go forth with adding a rear attachment to the bike in which the power supply will be resting. This is due to the PCB, and we believe that the rider would feel more comfortable with that orientation. Next would make sure that all the wiring is secured and connected, so that all the components of the bike are integrated. The throttle requires 2 pins for access on the PCB one is a switch and the other is ground. The MFSwitch has 5 pins which will need to be connected to the PCB, there are four digital outputs and a ground pin. All wires will be secured to the bike frame where possible by using zip ties when all bundled together.

5.7.2 – Software Summary

The application was designed with simplicity in mind. The group understands that not everyone who rides bicycles (or e-bikes for that matter) are technologically savvy and know their way around a smartphone. It was desired to keep the application simple, along with contrasting but pleasing colors of beige and gray to assist in differentiating content.

The home screen was phased out and a linear experience was opted for. While the home screen would have been a great app design feature, it was deemed unnecessary for the features we added; the features that *would* have gotten their own page were just merged with the HUD Page. The group believes that, while the home screen was desired to maintain simplicity, the finished version of the application is very simple and easy to use, without the home screen. Buttons are large, text is legible, and icons are used to help with button differentiation.

6.0 Project Prototype Construction

For this section of the project, we will be introducing some of the features of the design that are going to be added to the bike. If possible, we will give step by step instructions on how to assemble and mount the component. The prototype will also contain all the problems we may come to face as we go further along with the process. Most of these problems have yet to be seen, however, because the construction of this project will be done over the course of Senior Design 2. The code for the project will also be spoken about here as the app is a vital part of the goals we have set earlier in the course. We expect the first iterations of the prototype to not work and/or be faulty to which we will have to create more prototypes.

6.1 Battery mounting

The battery pack came in two parts. The mounting bracket and the battery itself. The bracket will be mounted in front of the knees along the rim of the bike. Here are the included parts:

- Battery fix plate and part
- Battery and key
- Charger and AC wire
- Manual and qualified certificate

To install onto the bike, we will need to drill two holes into the frame of the bike then use the screws to secure the battery mount to the bike. The battery pack itself will be removable from the mount and has a key that is required to lock the battery pack onto the mount. As seen below the bottom is the mount that will need to be secured to the bike.

In the end, however, we may decide to not mount the battery pack near the knees and instead get a rear cage to hold the battery and instead put the PCB between the legs as that could be the best place to put the Bluetooth module and ensure the best possible connection.

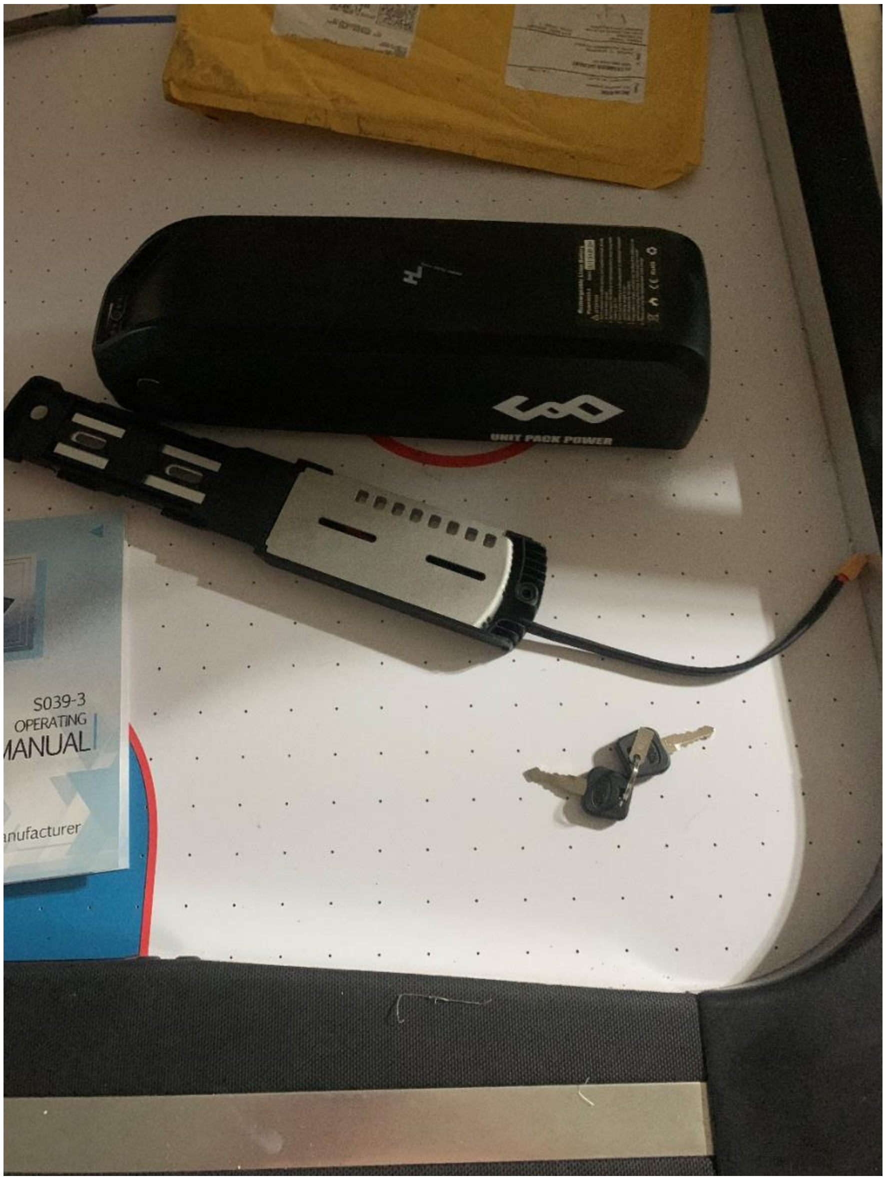

Figure 62: Our battery pack

6.2 Motor Mounting

Figure 63: Our wheel and motor

Parts included:

- G310 Hub Motor
- Axle Spacer Wash
- Tabbed Washers (2)
- M10 Washers (2)

- Acorn Nuts (2)

1. If the bike has disc brakes, install the disc rotor; in this case, make careful to remove the disc rotor spacer.
2. On the cable side, the hardware is arranged as follows: tabbed washer, dropout, M10 washer, and acorn nut. Make that the motor and tabbed washer are positioned such that the cable emerges downward. Water intrusion may occur if the wire is oriented upward.
3. The hardware is installed in the following order on the non-cable side: axle spacer washer, tabbed washer, dropout, M10 washer, and acorn nut.
4. Once the hardware is in place and the engine is mounted in the dropouts, tighten the axle nuts to 40Nm (30 ft-lb).

The Grin wheel builder will have dished the wheel presuming the addition of the axle spacer washer on the cassette side if the motor was purchased from Grin pre-laced. Otherwise, if we were doing the wheel build ourselves, we would need to be aware that adding this axle spacer (as needed for wider 9-11 speed cassettes) will have a modest impact on the correct dishing of the wheel, comparable to half the width of the spacer (0.75mm or 1/32").

6.3 RGB LED Strip Wiring/Mounting

1. It's quite simple to connect to the strip; you only need to solder four wires to the copper tabs. Red, green, and blue wires will be used for the matching LED colors, with white used for +12V.
2. Trim off the waterproof over molding at the strip's end. There is no right or wrong end to utilize because the strips are symmetrical.
3. To reveal the copper pads, scrape off the rubber. To tin the pads and burn off any remaining rubber, melt some solder onto them.
4. Glue the four wires together. Wire that is more flexible and likely a better option than solid core is the kind we utilized.
5. Use heat shrink to shield the wires and keep some waterproofness.

We can easily utilize any microcontroller with these LED strips because they are so straightforward. To regulate the strip, we advise using PWM dimming methods. Power transistors are needed since each "LED" pin can end up needing an amp or more to get to ground. Avoid trying to connect the pins directly to the microcontroller; we are at risk of having them burn out or malfunction. We can use any power NPN or N-Channel MOSFET, but make sure it is rated to carry as much current required. As we typically draw 0.2 Amps per channel per meter, for instance, a 5-meter strip would require passing up to 1 Ampere each transistor. For the mounting of the bike, we were going to use a strong adhesive to make the strip adhere to the bike. We want it to run across the frame of the bike and light up according to which action is taking place. For example, when we brake on the bike, we would want the lead strip towards the back light up the color red.

6.4 PCB Vendor and Assembly

For the construction of the PCB we need to put up an order with other businesses such as PCBWay or JLCPCB. The more popular of the two being JLCPCB. They appear to have the same deliverables with respect to time of around 5 business days if set to rush order, however, there are options to have the PCB built in 1-2 days then delivered in 2-4 business days. JLCPCB has a cheaper price, assuming a PCB size of 250x250 mm the price from PCBWay is approximately $156.04 while the cost from JLCPCB is $71.36. However, there are a lot more options to pick from at PCBWay such as the FR4-TG where the TG stands for Glass Transition Temperature. The higher this is the better temperature resistance of the material. Below is a table of comparisons of the different PCB venders that we may be going with.

Criteria	**Advanced Circuits/4pcb.com**	**Sunstone Circuits**	**Bay Area Circuits**	**allpcb.com**	**jlcpcb.com**	**pcbgogo.com**	**pcbway.com**	**Camptech**
Controlled impedance	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Minimum hole size	0.1 mm	0.15 mm	0.08 mm	0.2 mm	0.2 mm	0.15 mm	0.1 mm	0.05 mm
Trace width/Trace Spacing	3 mil	3 mil	3 mil/ 3 mil	4 mil	5 mil / 5 mil	3 mil / 3 mil	3 mil / 3 mil	3 mil / 3 mil
Aspect ratio	12:1		15:1			12:1	20:1	12:1
Routing tolerance	±0.15 mm	± 0.08 mm		±0.15 mm	±0.2 mm	±0.2 mm		±0.15 mm
Min. SMT Pitch					0.25 mm	0.1 mm	0.4 mm	0.15 mm
Blind & Buried vias	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes

Via in pad	Yes				No	Yes	Yes	Yes
Stacked micro-vias	Yes	Yes		Yes	No	Yes	Yes	Yes
Flexible PCB	No	No	No	Yes	No	Yes	Yes	Yes
Location	U.S	U.S	U.S	China	China	China	China	Canada

Table 28: PCB vendors

There are still many other PCB vendors such as Sierra Express, however, we could not possibly put all of them on this paper.

In the end we decided to go with PCBway and also bought stencils to aid with the soldering to the PCB. The average delivery time of the PCB is around 3-4 business days. The reason to buy the stencil is because the pins on the MCU are approximately one cm apart from each other so soldering by hand would not be feasible.

6.5 PCB design

There are a few different options to use when designing a schematic and the PCB. In junior design we learned using Autodesk's Eagle, however, the library research needed for these higher end programs take up much of the time when designing what you need. Sometimes half of the time of designing goes into the library. Instead, we are using EasyEDA to design our board because there is already a library made and because downloading and using the software is much easier compared to Eagle where a license is required for use. Another reason for the use of the program is the ability for use offline instead of only being accessible via cloud.

There will also be considerations if the MCU or other part on the board needs to be adjusted once soldered on. This will be accomplished by having a pin on the board to connect then using something like an Atmel-ICE unit what can use a UCB cable and an IDC flat cable to send commands to the MCU even after being flashed then soldered onto a board. There will also be LEDs on the PCB to ensure that different components are working. One example is there will be an LED to show that the DC/DC regulator is working properly.

Below is a WIP of the PCB without connecting the lines as the schematic is getting close to the finish line but not quite there yet. There are common does and don'ts for PCB designing such as not making 90-degree corners and we will need to be mindful to ensure the Bluetooth module does not have any interference at the antenna. We will also have to orient the board in such a way that the Bluetooth module is facing toward the sky because that is the most likely place for the phone to be and would provide the best coverage and have the best connection. Doing this would ensure a reliable connection and have the least chance of being dropped by the module or being unable to find the phone.

There are still many things to accomplish but during the break between senior design one and senior design two is when most of this work will be done because there will be no other classes to worry about during that time. Once it is finished the time to get the PCB will take around 2 weeks to get and we should be able to test it and determine if it is working within a week which will then need to be applied to the schematic then updated on the PCB to then be reordered.

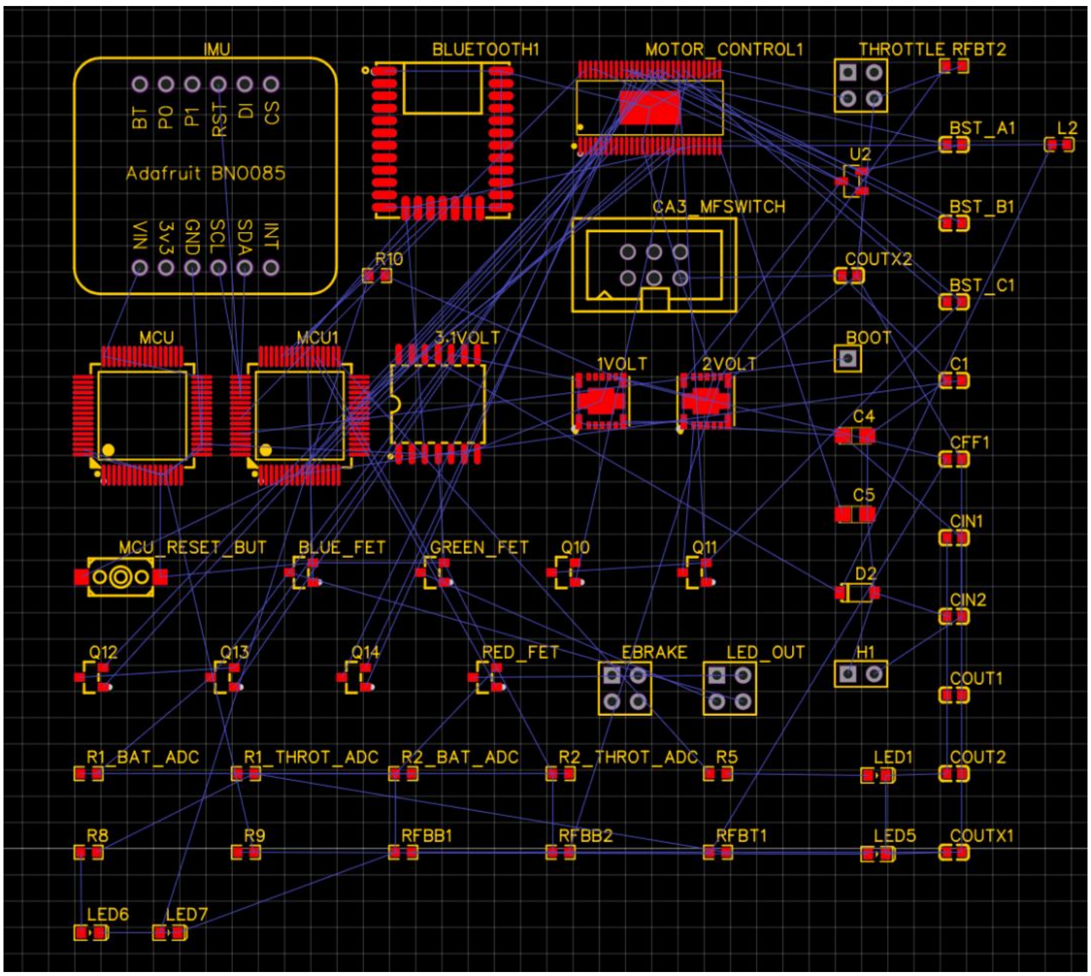

Figure 64: Initial PCB rats nest

While above was the initial iteration the following is the final design of the PCB after going through three iterations and testing. This PCB will allow communication with the app and be able to handle the input of 36 v and drop it down to the appropriate levels that will be utilized by the components.

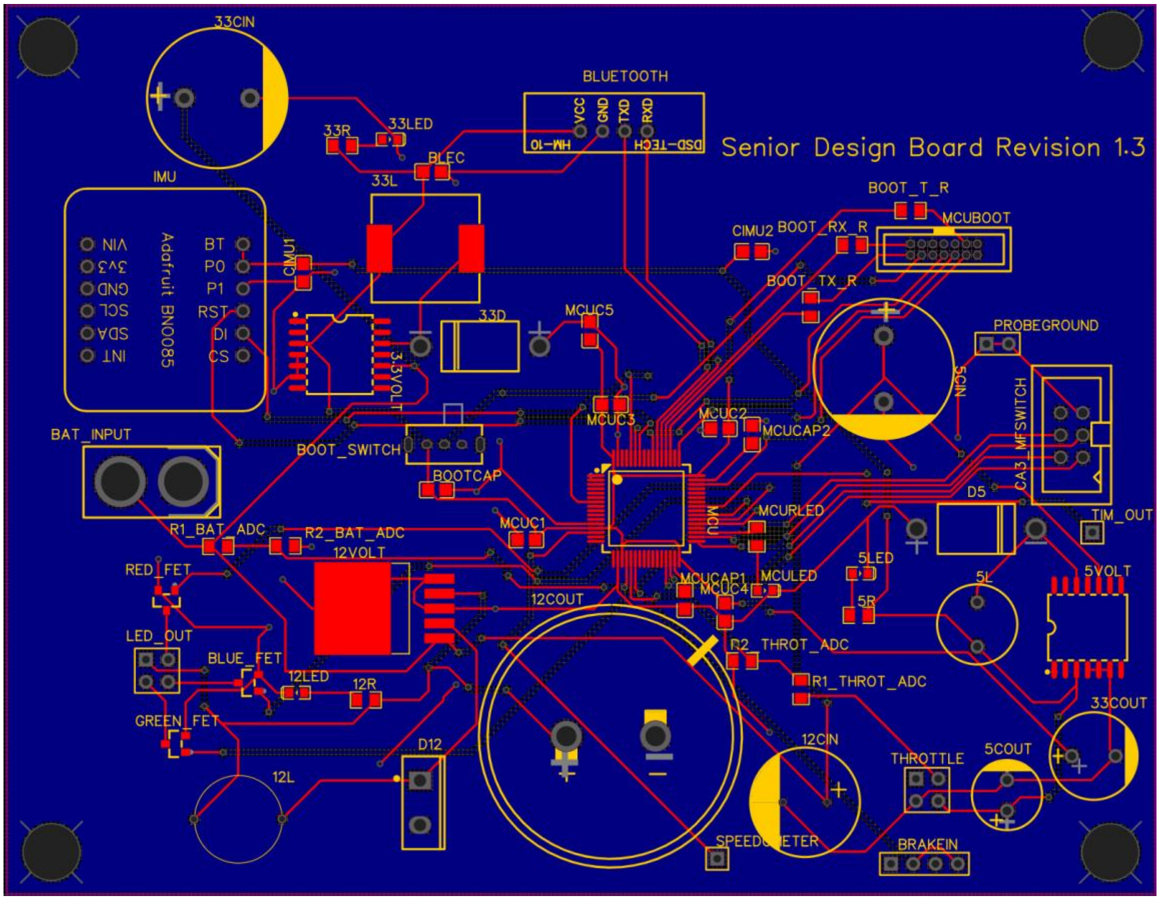

Figure 65: Last edition of PCB

6.6 PCB Waterproof Enclosure

Once the PCB has been assembled our goal is to get a case that allows it to be on the bike and that it is also waterproof. We will get the dimensions of the board and the orientation of all the wiring that is going into it. Afterwards we will plan to find or create a case for it. Essentially our design will have it rest on to the upper part of the frame or possibly on an attachment that could onto the back portion of the bike. Nonetheless, we are going to make sure that enclosure will be at least IP65 to IP68 in waterproof rating. Below are possible designs in which we will go with:

Figure 66: Clear case PCB enclosure

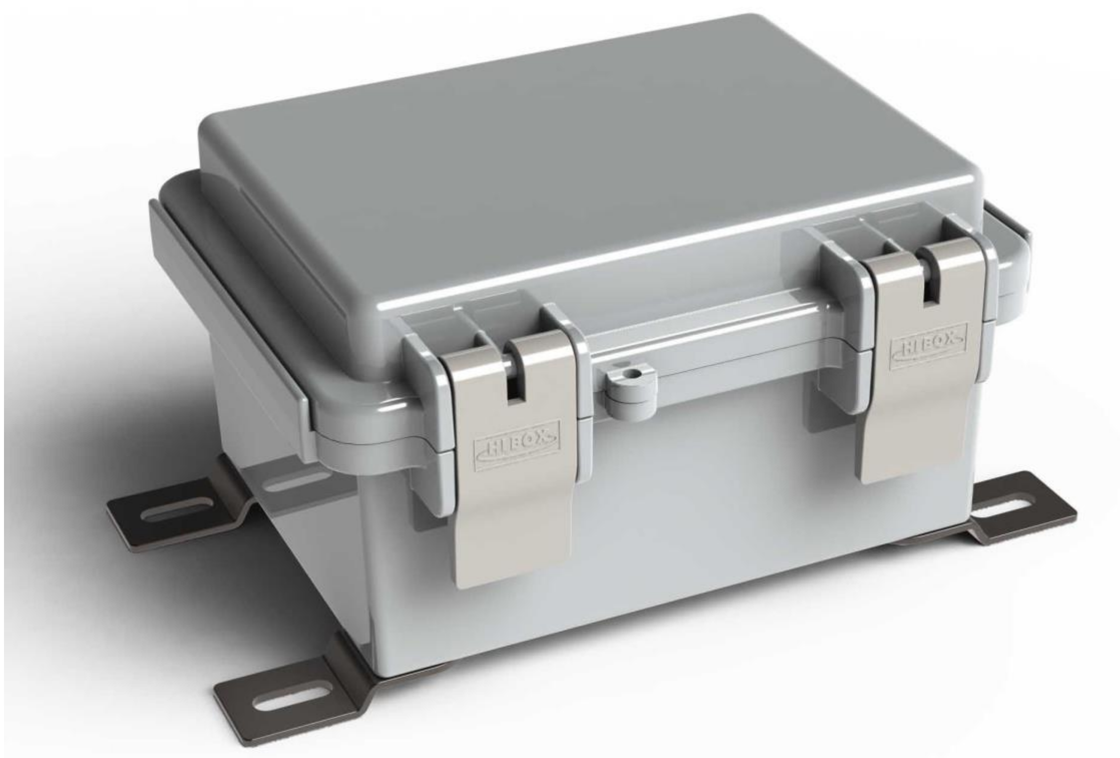

Figure 67: Hinged PCB waterproof enclosure

7.0 Project Prototype Testing Plan

In this section of the report, we will be explaining the different ways in which we will be conducting the testing phase. The goal of this section is for us to test the system and make sure that all parts are communicating due to proper integration. Also, with aiding us in making sure we can locate any errors and make the appropriate fixes that need to be made. The testing phase will subject our design to different tests in both hardware and software. For example, we would want to app to display the speed and battery life of the bike. So, we would need to make sure that the microcontroller and Bluetooth module is communicating appropriately. Then the Bluetooth module would then be able to communicate with any device that is connected to it. Essentially, any tests that are conducted to any of the physical parts on the PCB will be considered hardware and anything else such as the application that we are creating is considered software.

7.1 Hardware

Tools for Testing
Hantek 2D42
Digilent Ultra Analog Discovery 2 Bundle
Windows PC
Bluetooth enabled device
BK precision 2860
Kungber DC power supply

Table 29: Tools used for testing

Every engineering project needs to include a stage for testing operational hardware. Regardless of where a product comes from, it is essentially meaningless without comprehensive testing. Such rigorous testing and quality control techniques are used by the most reputable and regarded businesses, producers, and university inventors. The challenging testing and endorsement that those experts consider in their decision-making processes are influenced by a number of elements, but this project was designed and created by us senior level engineering students. Nonetheless the proper testing procedures must be taken into count in order to have the product with minimal to no problems.

7.1.1 Test Environment

The design will need to be able to use a power source that runs between 30 to 42 volts and have the ability to recharge. With this we would need to pick a battery that can pass any environmental or safety issues. Having a gadget undergo automated discharge cycles will help maintain battery life since people are less likely to totally deplete rechargeable batteries before recharging. More than enough power is produced by the rechargeable batteries to support the entire circuit. Delivering adequate electricity to every component of the system is the goal. To achieve optimal efficiency, the converter we are adopting is based on a pulse width-modulation (PWM) controller. The converter enters power-saving mode at low load currents to maintain high efficiency over a range of load current. As a result, the device's components are all appropriately powered.

In terms of the motor, it will be important to make sure that it can withstand any terrain. Since our design will be using a rear-hub motor it is important to make sure that the connecting wires are hooked up correctly and secured. This is especially important if the design is introduced to a rougher terrain. There is a possibility for the hardware to be jostled around and can ultimately add more weight to the frame of the bike, thus leading to the connecting wires getting damaged or weaken.

7.1.2 Durability

The most crucial testing for this project will take place in various weather scenarios. Given that the consumer will want to use this item almost everywhere (most frequently outside while cycling), the weather must be taken into account while considering this. The tool must be resilient to the majority of weather scenarios. We will test our powered tracking device's capacity to function even in the wettest of weather circumstances because if it can't, it won't be able to withstand those types of conditions for very long, which won't be good for us. For these kinds of circumstances, we must complete our design so that we won't be late on the day of the presentation. What kind of material will work best for our testing needs to be considered when doing this durability test.

This project's dependability and performance during our step-by-step construction process will demonstrate to us how well the project will hold up over time. We will be able to determine when our durability testing is complete whether or not our project is reliable enough to be used in the real world. The biggest concern at the moment is if our project's overall cost since we are funding it ourselves and for us to meet deadlines. As the quantity of money, we have will impact the types of materials we may buy, the time frame is another factor that will affect the project's durability.

To combat the durability, we will use items rated for use in weather. On spots with exposed wires that were soldered, we used liquid electrical tape. On wire-to-wire connections we used heat shrink tubing. This will prevent water from interfering with connections that used to be exposed.

7.2 Hardware Specific Testing

Regarding specific hardware testing, one of the components we will be testing are the LED light strips that are on the bike. We would make sure that when applying the brakes, the lights will then begin to go red. Also, when asked to be in "Night Mode" the lights will begin to flash at a rate in which we decide. Next, we would test the longevity of the bikes' power supply. Essentially what we will do is take the bike on a ride in different terrains to see how long the battery will last and how quickly it depletes. This also will lead us to test and make sure that everything on the PCB is integrated and communicates properly. When it comes to testing the motor, we have selected we will be conducting similar testing procedures such as the battery by taking the bike on different terrains to see how it holds up. In addition, we will conduct the simple testing of turning the motor on/off and having it run for a bit of time. This will not only test the motor but also test the motor controller as well, if the motor does not turn on or stutter then there is an error in the connection. This is important for the fact that we are using a rear-hub motor for our design.

Next, we can conduct the testing of the hall sensors that are inside of the hub motor and we will be using the multimeter to test them. We will use these multimeter prongs to test the voltage of the hall sensor by turning it on with our power source and placing the prongs on it to get a reading, which we will then compare to the recommended voltage listed in the manual provided. This then leads us on to testing the throttles that we selected from the project. In terms of testing the throttles are able to test them both on and off the bike. The ideal way of evaluating the throttle is the on-vehicle throttle test since it confirms that the controller is powering the throttle and that the throttle is sending a signal back to the controller. Instead of the power that the throttle would typically get from the controller, a 5 Volt DC power source may be used to test a throttle off of the bike. Turn the throttle slowly from zero to full power while doing the off-vehicle throttle test, and then check the voltage with the multimeter. As the throttle is turned, the voltage should respond appropriately to the location of the throttle. When the throttle is released, the voltage reading should be low, and when it is fully opened, the voltage measurement should be high.

7.3 Software

Tools for Testing Application
Windows PC
Macbook Air
Visual Studio Code
iOS & Android Emulators
Personal Smartphone Devices

Table 30: Tools for software testing

The software included in this project is somewhat straightforward to test, but it must be tested, nonetheless. As mentioned previously in this paper, Flutter and Firebase were selected as technologies to construct the custom smartphone application. Since this selection, the group has realized that, although we selected Firebase as a BaaS (Backend-as-a-Service), we are not planning on storing any kind of data yet. In the current build of the application, there is no stored data and figuring out data to store is not yet a focus of the application's development. As of now, the focus of the application's development is properly setting up Bluetooth and ensuring the connection between the application and the e-bike can be established and tested. This does not imply, however, that we are excluding having stored data entirely; there is just no data to store as of now. There is still an extensive amount of testing to be done on the application regardless of the status of stored data.

Tools for Testing MCU Software
STM32F407G DISC1 Development Board
STM32CubeIDE

Table 31: Tools to test MCU software

To write and test the MCU code we will use the *STM32F407G DISC1* development board using STM32CubeIDE. The development board uses an integrated programmer and debugger ST-LINK/V2 to use the board. For our final design we will use a standalone programmer which is the newest model by STM the STLINK-V3SET. A programming interface header will be on the PCB for iterative tests of the final software. Since the MCU firmware is mostly for motor control, data collection, and Bluetooth communication the interface of the MCU will be done through the smartphone application. The behavior of the bike and any signals, faults, or errors should all be sent to the app for the end user to see and manage.

7.4 Software Specific Testing

It should be noted that the purpose of the screenshots within this section is not to provide code for the application, but rather to visualize the features within the application's development environment that assist with testing.

7.4.1 – Application Environment (Virtual Studio Code)

Thanks to the fluidity of Flutter's compilation capabilities, running and testing the application in its current development environment is both relatively simple and developer friendly. The application is currently being developed in Virtual Studio Code (VS Code), a Microsoft-made source code editor with an alarmingly large number of extensions and libraries to support almost every programming language one can think of. Even though it is a Microsoft-made technology, it is usable on all three major operating systems: Windows, macOS, and Linux. Most, if not all, programmers nowadays have some familiarity with VS Code, and for good reason. The decision to use it was simple due to this flexibility it provides.

Within VS Code are multiple different features that streamline development/testing. In the context of this project, there are four features that stand out: syntax highlighting, Git integration, code detection/documentation, and emulation support. Syntax highlighting is not a new feature by any means, as most Integrated Development Environments used today have syntax highlighting, but it should be mentioned how useful it is to have it to help differentiate between variables, methods, classes, etc.

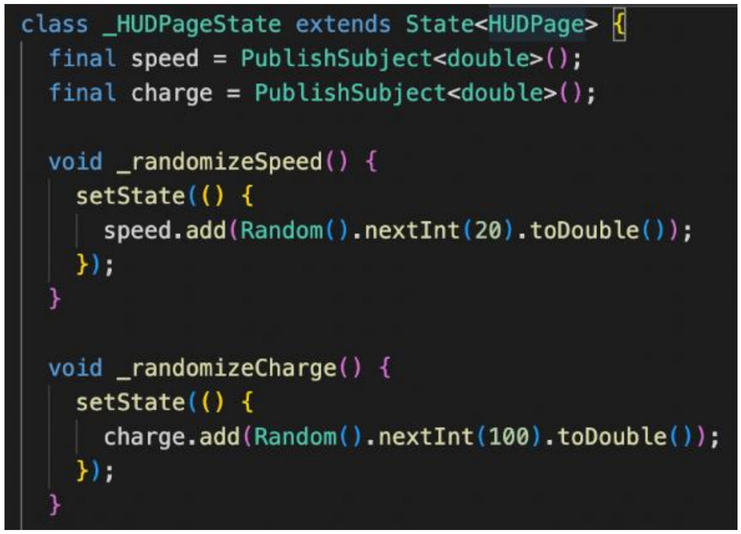

Figure 68: Example of syntax highlighting

Git integration is an incredible feature. Git is generally utilized via a command line/terminal with commands such as "git init", "git commit", or "git push". With VS Code, however, there is a separate icon/screen for adding, committing, and syncing (pushing/pulling) changes to and from a Git repository. This screen is named "Source Control" and provides a list of the files either added, removed, or edited since the last sync. Below is a screenshot of this screen and an example of how VS Code tracks changes made to files under the Working Tree and compares them to the branch's current file state.

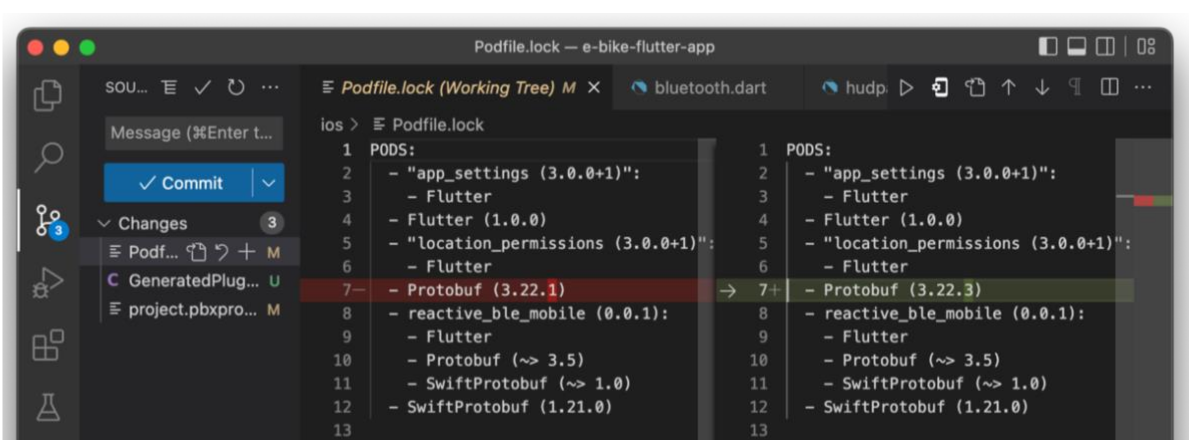

Figure 69: The "Source Control" screen in VS Code

Code detection/documentation isn't so self-explanatory. When writing code, VS Code will detect, based on the characters typed, and provide suggestions for what it believes you are trying to write. For example, in the screenshot below, it is desired to access a value 'title' within the object 'widget'. VS Code knows what values are held in 'widget' and will suggest them for you.

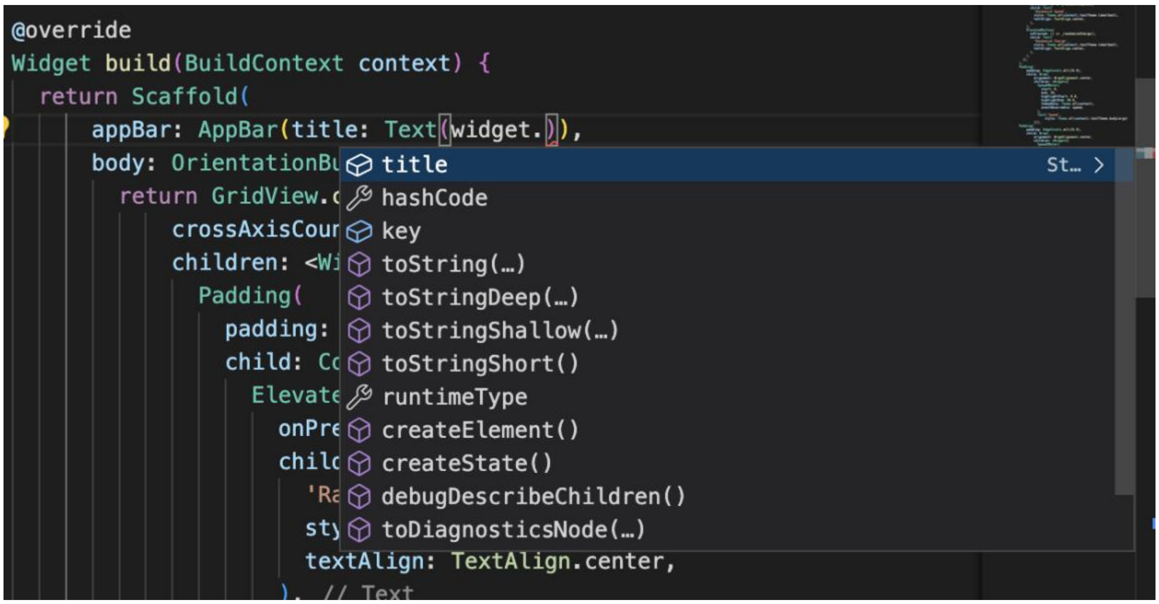

Figure 70: Code detection/suggestion

VS Code also supports running emulators. A developer can register different emulators within VS Code for it to use when debugging applications. This ties directly and seamlessly into Flutter's cross-platform-development capabilities. For Flutter to be used at all, one must download and install the technologies it requires; the technologies vary by operating system. The app is currently being developed on a macOS machine. Flutter required not only XCode (Apple's own iOS application environment), but also Android Studio. When setting up emulation, developer is required to go into both XCode and Android Studio and set up their respective emulators. Once the emulators are set up, VS Code can access and run the emulators.

The screenshots provided in section 5 of this paper were taken from the iPhone 14 Pro Max emulator running on the macOS machine. When running the application from within VS Code with a terminal command "flutter run", an emulator must be selected first. Only when an emulator is selected will the application run (if there is only one emulator, it will default to that one). More details on setting up the emulators cann be found below under section 7.4.2.

7.4.2 – Application Testing

As mentioned in section 7.4.1, there are multiple different features within VS Code that assist with development/testing. The most notable of these is the support for emulation, as most of the testing done for the application was done via an iOS emulator. With Flutter's cross-platform capabilities, an Android build has been constructed as well, but most of the testing is done on an iOS simulator due to a physical iOS device being the main device for testing outside of the development environment.

Before the developer can test their application, or even use Flutter rather, they must run a "flutter doctor" command. This command will have the already installed Flutter check for the dependencies it requires to function as expected. These dependencies include, among other technologies, Android Studio and XCode (for macOS users).

```
● christopherrumpf@Christophers-MacBook-Air e-bike-flutter-app % flutter doctor
  Doctor summary (to see all details, run flutter doctor -v):
  [✓] Flutter (Channel stable, 3.7.8, on macOS 12.6 21G115 darwin-arm64, locale en-US)
  [✓] Android toolchain - develop for Android devices (Android SDK version 33.0.2)
  [✓] Xcode - develop for iOS and macOS (Xcode 14.0)
  [✓] Chrome - develop for the web
  [✓] Android Studio (version 2022.1)
  [✓] VS Code (version 1.71.0)
  [✓] Connected device (3 available)
  [✓] HTTP Host Availability

  • No issues found!
```

Figure 71: The output of "flutter doctor"

Testing begins with a quick "flutter run" command in the terminal. This puts in the command to run the application and, when the emulators/devices are recognized, will select an emulator/device to build and run the app on (if multiple are detected, a choice will be presented). When a device is selected, it will continue compiling and building the app for that platform. There are differences in what Flutter must do to build and launch the app per platform:

- Android
  - Building for the Android platform is quite simple. Although the emulator set up was lengthier than the iOS emulator, the building process is simpler.
    - To set up a virtual Android device for emulation, the developer must have Android Studio installed. As mentioned previously, the developer can run "flutter doctor" to check for any missing dependencies. The most recent version of Android Studio (as of writing this document) is Android Studio Electric Eel, 2022.1.1 Patch 2.

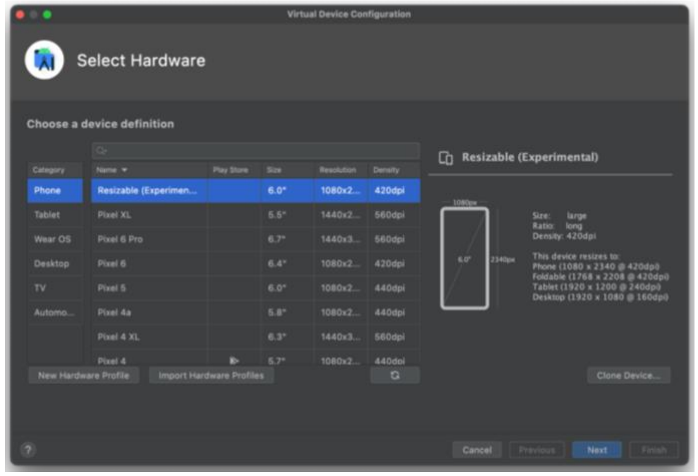

Figure 72: The device configuration in Android Studio

- Setting up Android emulators is done in the Android Studio application. When setting up the device, the developer is given options about which physical device they would prefer to emulate, as well as the image of the Android operating system they wish to emulate as well. After these choices are made, along with some other trivial options, the device is then set up. After this is set up, VS Code should be able to recognize it as an Android emulator and can then use it to launch the app, if so desired.

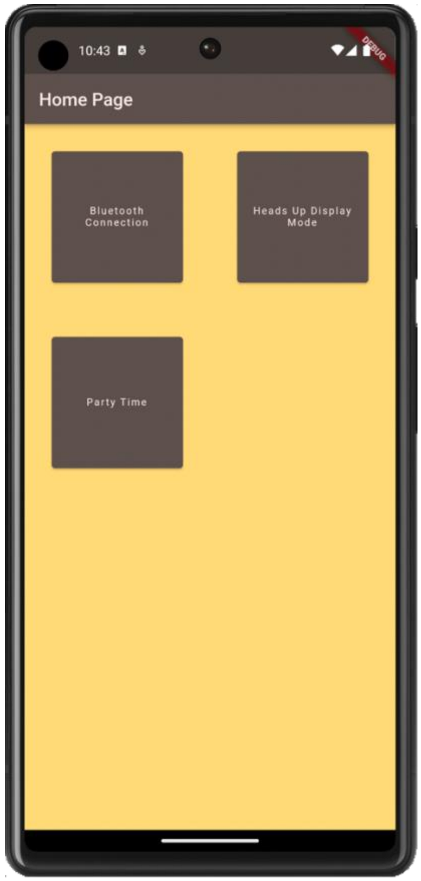

Figure 73: App running on Android emulator

- iOS
    - Building for the iOS platform was both simple and difficult. The iOS emulator was much easier to set up and use than an actual physical iOS device. When an iOS emulator was being selected for VS Code, there were already a lot of pre-loaded emulators; it currently being assumed that these emulators came loaded onto the MacBook Air the application is being developed on. Little set up was required to run this.

Figure 74: List of preloaded iOS emulators

  - The physical iOS device (a personal iPhone) was much more difficult to set up. To get Flutter to build an application on a physical iPhone, not only must the device's Developer Mode be turned on (which requires a device restart to take affect), a development team profile must also be set up. A development provisioning profile can be made in XCode, which is why Flutter requires XCode to build iOS applications. When the application is built and run, the development certificate associated with this provisioning profile will be signed onto on the physical iOS device. This certificate must also be trusted on the device for Flutter to continue.
    - This is all assuming the physical device is running a supported version of iOS. With the version of XCode used at the time of writing this, the latest physical iOS device version supported by XCode is 16.0. This is 4 versions behind the physical iPhone used by the group (it is currently running 16.4). Luckily, there are version support folders available on the Internet that can add support capabilities for versions that may not be available. The support folder for iOS 16.4 was downloaded and installed into XCode for the version support; it works perfectly as intended.

When the application is finished building, it will automatically launch on the emulator/device. After the application has launched, it is entirely usable (assuming there are no errors). A more in-depth explanation on how the application works and functions can be found in section 5 of this paper.

7.4.3 – Deploying to App Stores

When developing an application, regardless of platform, for consumers to access the application, it must be deployed to the platform's respective app store; for iOS: the App Store, and for Android: the Google Play Store. Both stores have their own process to deploy an application to it for it to become available for download.

- App Store
  - As mentioned in the previous section, a development provisioning profile and its associated development certificate must be created in XCode for the app to be loaded onto a physical iOS device. There is a similar concept when deploying an application to the App Store; a developer will need a *distribution* provisioning profile and its associated distribution certificate. This can be done in App Store Connect by either registering as an individual or creating an organization.
    - This registration will cost $99 on an annual basis.

    - After this is created, through App Store Connect, the developer will be able to upload, configure, and submit a new application for review to the store. App Store applications must be reviewed and verified by Apple to be made available on the store. The review period can take from one to three days. After the application is reviewed, it can either be approved or rejected. If it is rejected, changes/fixes must be made before it can be submitted again. The Resolution Center in App Store Connect can be used to communicate directly with Apple for any questions a developer may have. If the application is approved, it will be added to the App Store.
- Google Play Store
    - The process is quite similar to Apple's App Store, although much simpler. One must create a Google Developer Account, but also paying a registration fee of $25 dollars as opposed to an annual $99. After this account is created, the developer can add an app to their account, configure its descriptions/settings, upload the APK, and submit for review. The Google Play Store's review process takes significantly less time than Apple's App Store's review process does.
        - Although the Google Play Store has a higher approval rate with submitted applications since they allow for more creativity, a downside to this high approval rate is that the Google Play Store does not provide significantly helpful guidance if your application is rejected.

8.0 Administrative Content

In this section will be the discussion of what has been accomplished with respect to the timing of accomplishments such as when products were bought or designed. We will also be speaking on the cost of items needed to finish the prototypes and get to the final product. The timing of the deliverables is also important because we need to ensure that items are being delivered on time. Having the 3-minute video being required certainly accelerated the process of the group buying parts, however, we do not know for certain if these parts (such as the Bluetooth module) will end up working for the project and need to be replaced.

The electrical side of the project (Alex and Carlos) have already known each other from beforehand so we already knew how to work together. Victor and Chris both came into the fold very quickly and we all work well together. The team would meet at least once a week and sometimes more than that if there is going to be a deliverable soon (such as the 120 page). The group is very forgiving if something happens, and we all understand that life gets in the way and that we have other classes. Everyone works well together and has met outside of the weekly meeting not for the purpose of this class. In the very beginning all the group members were interested in the project so motivation is not an issue and all items that needed to be completed are completed on time.

8.1 BILL OF MATERIALS (BOM)

Part	Price	Quantity	Links
Motor: Bafang G310 Standard Wind (8.5 rpm/V) Geared Rear Hub Motor	240 USD	1	https://ebikes.ca/mg310-std.html
Torque Arm: GRIN Front Torque Arm	20 USD	1	https://ebikes.ca/torqarm-311.html
Rim: 26" DM30 Weinmann Double Wall Black/ Silver 30mm Wide Rim	26 USD	1	https://ebikes.ca/rim26_weinmann_r-rim26_dm30.html
Sapim Strong 13-14g Butted Spoke, Silver	1 USD	10	https://ebikes.ca/spcust14.html
Right Full Twist Throttle (140cm Cable)	15 USD	1	https://ebikes.ca/t-twist.html
Dual Ebrake Levers (50cm Cable)	24 USD	1	https://ebikes.ca/ebrakewuxd.html
Multifunction Switch for CA3-WP, with On/Off Power, Up/Down Digital Aux Buttons, and Screen Toggle. 45 cm cable length.	20 USD	1	https://ebikes.ca/ca3-mf-switch.html
60 cm Motor Phase Extension Cable with Waterproof Z910 Plugs.	15 USD	1	https://ebikes.ca/z910-ext-60-fmk.html
STM32F405RGT6 32 Bit Microcontroller	15.56 USD	2	https://www.newark.com/stmicroelectronics/stm32f405rgt6/32-bit-microcontroller-advanced/dp/71T9854
DRV8302DCA (Motor Controller)	7.35 USD	2	https://www.mouser.com/ProductDetail/Texas-Instruments/DRV8302DCA?qs=UMk2QrKu%2FwwgoG3aN39ZGg%3D%3D

BGM220PC22HNA2 (Bluetooth Module)	11.01 USD	2	https://www.digikey.com/en/products/detail/silicon-labs/BGM220PC22HNA2/12317141
Unit Pack Power Official 20Ah /15Ah / 13Ah Ebike Battery - 52V 48V 36V Bike Battery for 0-1500W Bafang Voilamart AW Ebikeling and Other Motor	328 USD	1	https://www.amazon.com/Unit-Pack-Power-Ebike-Battery/dp/B0B3T9M3G1/ref=sr_1_3?keywords=ebike%2Bbattery&qid=1679605785&sr=8-3&th=1
RGB LED weatherproof flexi-strip 30 LED/m - 3m	48 USD	1	https://www.adafruit.com/product/2485
Adafruit 9-DOF Orientation IMU Fusion Breakout - BNO085	24.95 USD	2	https://www.adafruit.com/product/4754

Table 32: BOM for main components

8.2 BOM for the PCB (WIP)

Device	Value	Quantity	Price	Total Cost
926-LM2576HVS-12NOPB	Voltage regulator	1	$8.97	$8.97
LM2574HVM-5.0/NOPB	Voltage regulator	1	$3.866	$3.87
926-LM2574HVM3.3NOPB	Voltage regulator	1	$4.69	$4.69
Diode	850mV 5A	2	$0.42	$0.84
Diode	950mV 10A	1	$	

Power MOSFET	800V 30A	6	$4.16	$24.96
2N7002	0.4A	12	$0.156	$1.872
Resistor	100k	5	$0.33	$1.65
Resistor	35.7k	5	$0.1	$0.5
Resistor	8.56k	5	$0.93	$4.65
Resistor	68.1k	5	$0.1	$0.5
Resistor	0, Jumper	10	$0.018	$0.18
Cap	0.1uF	25	$0.044	$1.10
Cap	470uF 25v	1	0.71	0.71
Cap	470uF 10v	1	0.37	0.37
Cap	330uF	1	11.43	11.43
Cap	470uF	1	1.5	1.5

Table 33: PCB BOM WIP

8.3 BOM for Development Boards

Device	Price	Link
MCU	21.17 USD	https://www.digikey.com/en/products/detail/stmicroelectronics/STM32F407G-DISC1/5824404
IMU	24.95 USD	https://www.adafruit.com/product/4754
Bluetooth	16.78 USD	https://www.digikey.com/en/products/detail/silicon-labs/BGM220-EK4314A/13278045
Bluetooth backup	10.99 USD	https://www.amazon.com/dp/B06WGZB2N4?psc=1&ref=ppx_yo2ov_dt_b_product_details
MCU Programmer	34.30 USD	https://estore.st.com/en/stlink-v3set-cpn.html

Table 34: Dev board BOM

8.4 Milestone Discussion

As the project design continues, we begin to make notice of us getting to and completing our milestones. Not only the ones set to us by the course but ones we have set for ourselves. We were able to make our deadlines in terms of writing and editing our projects document within each of them. This led us to start looking towards finalizing our part selection and ordering them. This process was helpful in both creating a technology demonstration video we needed for the course and for the PCB design. As of right now we have made tremendous progress in our PCB design and are currently looking for the best priced vendor to place an order. Many of our major components have been ordered and arrived at the predetermined location. We have been able to disassemble the bike that we intend to use and start mapping out where most components and wiring that is going to be on the bike. In terms of the hardware, we have been very happy with the progress we have been making but still have some more work in terms of PCB design and assembly. None the less we are more than capable of having a fully functional design.

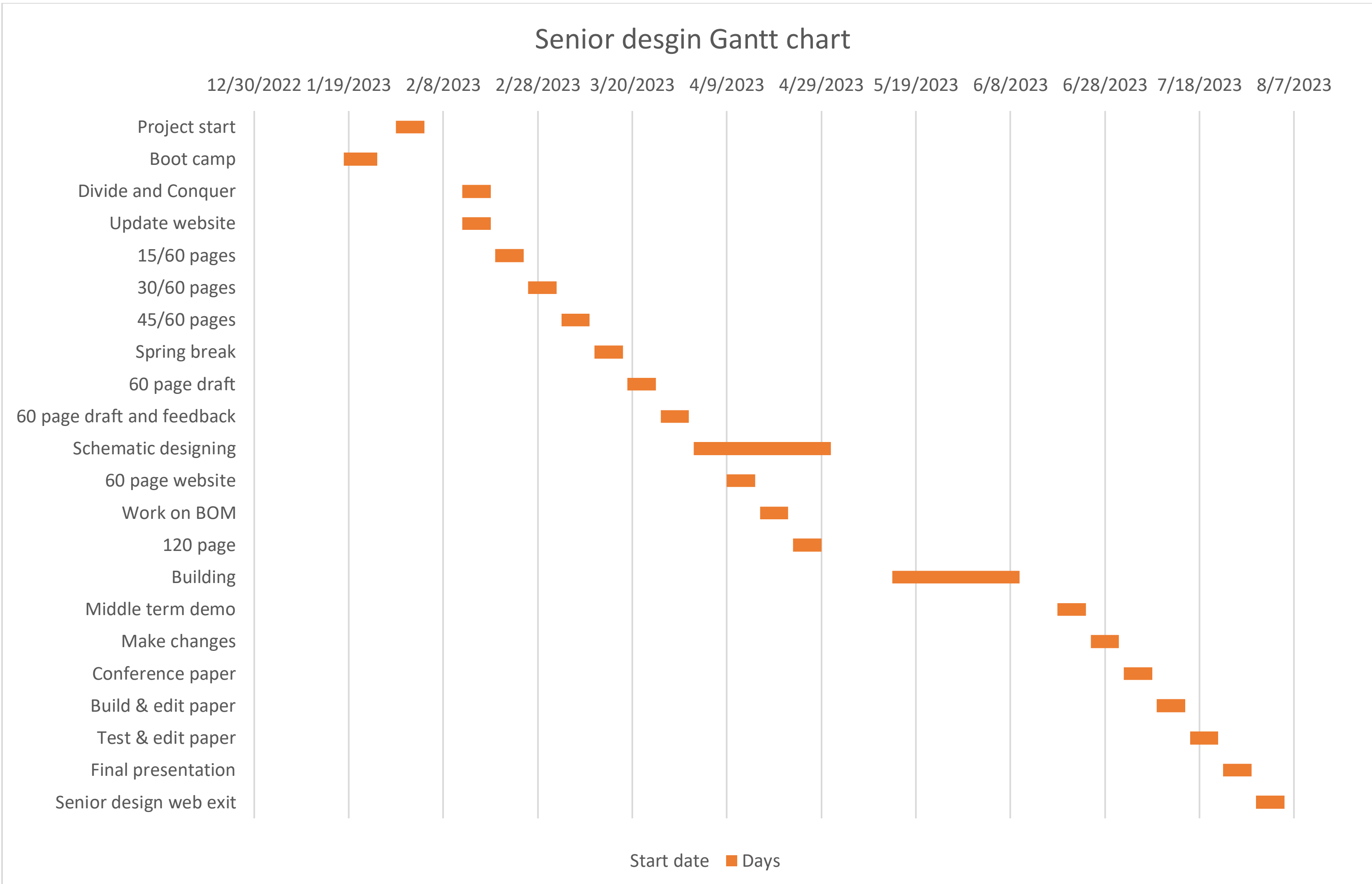

Figure 75: Gantt chart

The software side of development for this project provided its own set of milestones as well. When selecting the technologies the group is using for this project, the group decided to go with Flutter for application development. This selection was detailed more in section 3.10 – App Design, but to provide a little summary: Flutter was a fantastic decision for application development.

While it was a great decision, the research performed for Flutter resulted in community reviews of the pretense that while Flutter is simple to use, its language Dart has a steep learning curve. After exposure to it, it can confidently be stated that these reviews hold truth. Dart was slightly difficult to learn, enough so that getting accustomed to it can be considered a milestone.

After getting accustomed to programming in Dart, getting a sample application up and running was next on the milestone list. Thankfully, when Flutter is installed, it comes preloaded with sample code that, when built and launched, provides a sample application to get started on. The only things provided within this sample application are a welcome message, a counter, and a button to increase the counter by one. The group's application was built off this sample application, using the sample code as reference when needed. After the sample application was up and running, next on the milestone list was adding pages and providing the desired structure to the application based on diagrams the group created previously. Once these pages and structure were added, the sample code from the initial Flutter install was phased out. The only code left in the application is that written by the group. This was another big milestone as now we can confidently say that everything within the application is that of our own, no sample code, no introductory tutorials, just our own work and research.

As of writing this section in the paper, a significant amount of the application has been developed. Not nearly enough to be considered "done" or "finished", but a decent chunk has been taken out of the work to come. Currently, the next major milestone to achieve for the application is to incorporate the Bluetooth connectivity capabilities. From the research performed on enabling a Bluetooth connection with a custom application, this milestone not only may be the most difficult, but is, without a doubt, the most significant milestone pertaining to the application. The application must be able to establish a connection with the e-bike for the heads-up-display (HUD) page to function as desired.

9.0 Project Summary

Essentially with this project we want to be to create an e-bike design that is both useful and intriguing. As stated previously people need to travel almost everywhere in cars, whether they are going to work, school, or the grocery store. The user has more freedom with an e-bike. An e-bike is a more affordable and effective alternative mode of transportation in cities compared to electric automobiles. The affordability of an e-bike in comparison to a car enables more users than the price of an automobile. Though that may be the case we would encounter the obstacle of creating the best design at the lowest price possible. We believe that we can do this through the many team meetings that were helped in the selection process of each component and with each design idea. In addition, we have one of our members of the group with some personal experience in this type of project and one of our committee members can be considered a biking enthusiast that gave us very good criticism. In a lot of the decisions that were made in terms of part selection we would take in consideration all opinions, but we would be particular in making sure that for specific parts we would pick performance over cost.

Overall, we anticipate that the project will offer a straightforward implementation because we have chosen components that are known to work well together. By doing this, we intend to improve our design's efficiency throughout the construction phase. We anticipate using both home resources and the laboratories at UCF during the integration process to complete soldering and mounting operations. When it comes to having to 3D print or possibly acquire an enclosure for the PCB, we will weigh our options and see what phase we are in to make that decision.

On the software end our two CpEs have been going at great lengths to make sure that we are getting the best components for the PCB and in the development of the app. At this point we have a working app that is able to display our test cases such as changing color and display a random numerical value. The application has also been tested and opened on both Apple and Android products through emulation, as well as a physical Apple device. This project demands a lot of teamwork to complete, and a project management strategy must be in place to keep a team operating effectively. No project management plan is flawless but following a schedule might prevent problems in the future. We have done our due diligence to adhere to the milestone chart from Senior Design One, and we hope to produce comparable outcomes in Senior Design Two. We firmly believe that our design will be fully functioning, cater to most riders, and be a great overall product.

9.1 Challenges and lessons learned

In the beginning we decided to do an e-bike because we thought it would be an interesting project to do, especially with all the different components that we would be able to use.

The first challenge that was encountered was the bicycle not having enough space in the rear for the new tire with motor to fit well. The back was much too narrow that while it was possible to put the new tire on it would take more effort than normal to put the tire on. In order to get around this issue we used a different bicycle that Victor provided that has a wider back and could fit the tire much more easily.

The second challenge was with the Bluetooth library we were using in Flutter. Initially, we were unable to transmit information from the application to the Bluetooth module. This seemed to be an operating system issue at first, but it turns out the incorrect form of the device's identifiers were being used. To fix this, we used the correct identifiers. The Bluetooth connection is now a complete, two-way connection.

The third challenge we encountered was the design aspect of the bike since the new frame was a bit smaller. We had to come up with a new way to mount the power supply and PCB to the bike.

Works Cited

- “The Federal Register,” *Federal Register :: Request Access*. [Online]. Available: https://www.ecfr.gov/current/title-16/chapter-II/subchapter-C/part-1512?toc=1. [Accessed: 23-Mar-2023].
- A. Schirn, “ISO 4210-2:2023 - city, trekking, Mountain & Racing Bicycles - ANSI BLOG,” *The ANSI Blog*, 26-Jan-2023. [Online]. Available: https://blog.ansi.org/iso-4210-2-2023-trekking-mountain-racing-bicycles/?amp=1. [Accessed: 23-Mar-2023].
- A. Wang, S. Kadam, H. Li, S. Shi, and Y. Qi, “Review on modeling of the anode solid electrolyte interphase (SEI) for Lithium-Ion Batteries,” *Nature News*, 26-Mar-2018. [Online]. Available: https://www.nature.com/articles/s41524-018-0064-0#:~:text=A%20passivation%20layer%20called%20the,and%20ensure%20continued%20electrochemical%20reactions. [Accessed: 23-Mar-2023].
- B. Kelechava, “ANSI/ESD S20.20-2021: Protection of Electrical and Electronic Parts - ANSI Blog,” *The ANSI Blog*, 10-Jan-2023. [Online]. Available: https://blog.ansi.org/ansi-esd-s20-20-2021-protection-electronic-parts/. [Accessed: 24-Mar-2023].
- “ARM Programming - SparkFun Learn,” *learn.sparkfun.com*. https://learn.sparkfun.com/tutorials/arm-programming/jtag-and-swd (accessed Mar. 24, 2023).
- “Batteries,” *U.S. Consumer Product Safety Commission*. [Online]. Available: https://www.cpsc.gov/Regulations-Laws--Standards/Voluntary-Standards/Topics/Batteries. [Accessed: 23-Mar-2023].
- C. Pao, “The importance of IMU Motion Sensors - CEVA’s Experts blog,” *CEVA’s Experts blog*, Nov. 15, 2018. https://www.ceva-dsp.com/ourblog/what-is-an-imu-sensor/
- “The Federal Register,” *Federal Register :: Request Access*, 2015. [Online]. Available: https://www.ecfr.gov/current/title-30/chapter-I/subchapter-O/part-74/subpart-C/section-74.7. [Accessed: 23-Mar-2023].
- R. T. Corporation, “Li-ion battery and gauge introduction,” *Li-ion Battery and Gauge Introduction | Richtek Technology*. [Online]. Available: https://www.richtek.com/Design%20Support/Technical%20Document/AN024. [Accessed: 23-Mar-2023].
- “UL 2272 and the safety of personal e-mobility devices,” *UL solutions*, 2016. [Online]. Available: https://collateral-library-production.s3.amazonaws.com/uploads/asset_file/attachment/12042/10414_HoverboardSafety_V2R6_FINAL.pdf. [Accessed: 23-Mar-2023].
- “The Federal Register,” *Federal Register :: Request Access*. [Online]. Available: https://www.ecfr.gov/current/title-49/subtitle-A/part-37/subpart-A/section-37.3. [Accessed: 23-Mar-2023].

- "U.S. Energy Information Administration - EIA - independent statistics and analysis," *Use of gasoline - U.S. Energy Information Administration (EIA)*. [Online]. Available: https://www.eia.gov/energyexplained/gasoline/use-of-gasoline.php. [Accessed: 23-Mar-2023].
- "Learn about the different ways heading (azimuth) is determined · VectorNav," *www.vectornav.com*. https://www.vectornav.com/resources/inertial-navigation-primer/theory-of-operation/theory-heading (accessed Mar. 24, 2023).
- "Lithium-Ion Battery Standards: Energy," *U.S. Agency for International Development*, 21-Mar-2023. [Online]. Available: https://www.usaid.gov/energy/powering-health/technical-standards/lithium-ion-batteries. [Accessed: 23-Mar-2023].
- Person and S. D. David Shepardson, "U.S. driving soars in 2021 to 3.23 Trillion Miles, up 11.2%," *Reuters*, 18-Feb-2022. [Online]. Available: https://www.reuters.com/world/us/us-driving-soars-2021-up-112-2021-2022-02-18/. [Accessed: 23-Mar-2023].
- M. Timmons, "Car ownership statistics in the U.S.," *ValuePenguin*, 17-Aug-2022. [Online]. Available: https://www.valuepenguin.com/auto-insurance/car-ownership-statistics. [Accessed: 23-Mar-2023].
- "AN957 sensored BLDC motor control using DSPIC30F2010 - Microchip Technology." [Online]. Available: https://ww1.microchip.com/downloads/aemDocuments/documents/OTH/ApplicationNotes/ApplicationNotes/BLDCMC00957a.pdf. [Accessed: 24-Mar-2023].
- Cythinay, Rick, Ahmed, W. Vieira, and Dejan, "How Brushless DC Motor Works? BLDC and ESC explained," How To Mechatronics, 16-Nov-2022. [Online]. Available: https://howtomechatronics.com/how-it-works/how-brushless-motor-and-esc-work/. [Accessed: 23-Mar-2023].
- "Differences Among Arm® Cortex® Families - Developer Help," *microchipdeveloper.com*. https://microchipdeveloper.com/32arm:differences-among-arm-cortex-families (accessed Mar. 24, 2023).
- "Grin tech torque arm info page," *ebikes.ca*. [Online]. Available: https://ebikes.ca/product-info/grin-products/torque-arms.html. [Accessed: 23-Mar-2023].
- "Gyro sensors - How they work and what's ahead | about Gyro sensor | Technical Information | other Information," *Epsondevice.com*, 2019. https://www5.epsondevice.com/en/information/technical_info/gyro/
- "Highlights of the Automotive Trends Report," *EPA*. [Online]. Available: https://www.epa.gov/automotive-trends/highlights-automotive-trends-report. [Accessed: 23-Mar-2023].
- "Hub Motors - Learn," *ebikes.ca*. [Online]. Available: https://ebikes.ca/learn/hub-motors.html. [Accessed: 23-Mar-2023].
- "Summary of Ebike components - learn," *ebikes.ca*. [Online]. Available: https://ebikes.ca/learn/summary-of-ebike-components.html. [Accessed: 23-Mar-2023].

- "Highlights of the Automotive Trends Report," *EPA*. [Online]. Available: https://www.epa.gov/automotive-trends/highlights-automotive-trends-report. [Accessed: 23-Mar-2023].
- "Introduction to MEMS Accelerometers," *Pcb.com*, 2013. https://www.pcb.com/resources/technical-information/mems-accelerometers
- "5 major differences between IOS and Android App Development," *EGO Creative Innovations - we design apps that your users love*. [Online]. Available: https://www.ego-cms.com/post/5-major-differences-between-ios-and-android-app-development. [Accessed: 23-Mar-2023].
- B. L. E. M. Apps, "Top 5 Factors to Consider When Choosing Bluetooth Module," *BLEMobileApps*, 11-Sep-2020. [Online]. Available: https://www.blemobileapps.com/blog/top-5-factors-to-consider-when-choosing-bluetooth-module/. [Accessed: 23-Mar-2023].
- "Express/node introduction - learn web development: MDN," *Learn web development | MDN*. [Online]. Available: https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction. [Accessed: 23-Mar-2023].
- "Flutter app backend: The Best Five Providers," *Back4App Blog*, 22-Aug-2022. [Online]. Available: https://blog.back4app.com/flutter-app-backend/. [Accessed: 23-Mar-2023].
- M. (N/A), "Which Type of Electric Bicycle Throttle Is Best?," *EbikeSchoolcom*. [Online]. Available: https://www.ebikeschool.com/type-electric-bicycle-throttle-best. [Accessed: 23-Mar-2023].
- M. Li, "Complete Guide on Bluetooth Bodule," *MOKOBlue*, 20-Aug-2021. [Online]. Available: https://www.mokoblue.com/complete-guide-on-bluetooth-module. [Accessed: 23-Mar-2023].
- "Pedal assist or throttle controls," *Juiced Bikes*. [Online]. Available: https://www.juicedbikes.com/pages/chapter-5-pedal-assist-or-throttle-control. [Accessed: 23-Mar-2023].
- Person, "LED vs. Regular Lightbulbs: Do they really make a difference?," *Blog*, 16-Mar-2020. [Online]. Available: https://blog.arcadia.com/led-vs-regular-lightbulbs-do-they-really-make-a-difference. [Accessed: 23-Mar-2023].
- S. L. Staff, "Lighting comparison: LED vs Incandescent Lighting," *LED Lighting Distributor and Implementation Company*, 21-Apr-2016. [Online]. Available: https://www.stouchlighting.com/blog/light-comparison-led-lighting-vs-incandescent-lighting. [Accessed: 23-Mar-2023].
- "Silicon Sensing | MEMS Accelerometers," *www.siliconsensing.com*. https://www.siliconsensing.com/technology/mems-accelerometers/
- "The Piezoelectric Effect - Piezoelectric Motors & Motion Systems," *NANOMOTION*. https://www.nanomotion.com/nanomotion-technology/the-piezoelectric-effect (accessed Mar. 24, 2023).
- "Types of Lighting: Incandescent Bulbs," *Types of Lighting: Incandescent Bulbs | EGEE 102: Energy Conservation and Environmental Protection*. [Online].

Available: https://www.e-education.psu.edu/egee102/node/2035. [Accessed: 23-Mar-2023].

- omega, “Accelerometers & How to Measure Acceleration,” *https://www.omega.com/en-us/*, Aug. 28, 2018. https://www.omega.com/en-us/resources/accelerometers
- “VESC Project,” *vesc-project.com*. https://vesc-project.com/
- “What is a LAMP Stack?,” *Amazon,* 1978. [Online]. Available: https://aws.amazon.com/what-is/lamp-stack. [Accessed: 23-Mar-2023].
- “Which ARM Cortex Core Is Right for Your Application: A, R or M?” Available: https://www.silabs.com/documents/public/white-papers/Which-ARM-Cortex-Core-Is-Right-for-Your-Application.pdf