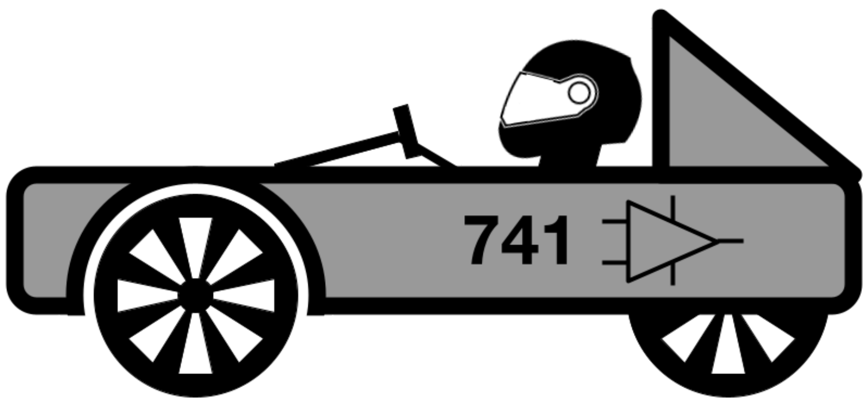
Rebuilt Electric Go-Kart



University of Central Florida
Department of Engineering and Computer Science

Dr. Lei Wei
Senior Design I
11/29/2023

Group 27

Fouad Braimoh - Computer Engineering
Abdullah Arshad Choudhry - Electrical Engineering
Grace Tuomala - Computer Engineering
Julian Yerger - Electrical Engineering

Review Committee

UCF Professors - Dr. Zakia Abichar
- Dr. Chinwendu Enyioha
- Mark Maddox

Workshop Sponsor - Larry Wexler

Table of Contents

Figure List

Table List

1 Executive Summary

There is a broad consensus that Electric Vehicles (EVs) must be adopted on a large scale to reduce greenhouse gas emissions from the transportation sector and minimize anthropogenic climate change. EVs have zero tailpipe emissions, improving local air quality, and they are also quieter and reduce dependence on foreign fuels. However, their adoption has been limited by the high financial and environmental cost of battery production, as well as the speed at which batteries can be recharged. This project aims to alleviate those problems by building and testing a small scale EV and optimizing the energy efficiency and battery charging rate. This minimizes the charging time and the amount of lithium required to travel a given distance.

The project scope is to build a new electrical system for an electric go-kart and race the kart against others with similar battery capacity such that the most efficient kart wins. The brief time between races requires either two battery packs or a DC fast charger, and we decided that the latter was more cost effective. Low voltage fast chargers are not commercially available, so building a custom fast charger is the first major element of the project. The second element is the motor controller, which we will build ourselves to achieve a higher efficiency than commercial off-the-shelf controllers. The third element is the Driver Information System, which will show the driver the car's energy consumption and energy remaining to help them stretch the available energy over the time remaining in the race. Information will be gathered by sensors on the car and also broadcast to spectators and pit crew. The following document shows our design process and our plan for how all of these systems will be constructed.

2 Project Description

This section gives an overview of the proposed project - an electric go-kart with customized hardware and software designed to improve race performance. The go-kart will be raced in the Tampa Bay Electrathon, which is the main basis for the motivation, specifications and constraints related to the project. Background information regarding the Electrathon is further detailed in the following section. Next, goals and objectives and specifications are listed. Related works, including other Electrathon race teams and previous senior design projects, are described. Then, a house of quality chart, hardware block diagram, software flowchart are provided, along with work distributions assigned to each member of the group.

2.1 Motivation and Background

Over the past several years, hybrid and electric vehicles (EVs) have been growing in popularity in an effort to reduce carbon emissions from gas-powered cars. A few years ago, a small group of friends at UCF sought to join this effort by forming the club Relectric. The club revolved around the design of electrical, software, and mechanical systems for electric vehicles. Two members of this senior design team, Julian and Grace, were primary contributors to Relectric's electrical team.

Inspired by other clubs building vehicles for racing, members of Relectric decided to build an electric go-kart. The idea led to the proposal of this senior design project: optimizing the electrical and software systems for an electric go-kart. The go-kart would be raced in the Electrathon - a competition for small electric vehicles that are optimized for efficiency. Current battery technology has much lower energy density than liquid fuels, so to achieve reasonable weight, cost, and range electric vehicles must be much more efficient than their gas powered predecessors. This focus on efficiency instead of power, downforce, or grip makes Electrathon more relevant to commercial electric vehicles than most racing leagues. Due to their small size, the cars are relatively cheap to build and modify, and due to their high efficiency, they have minimal wear on brakes and tires. Many teams race on the same set of tires for an entire season.

Electrathon races are held on a closed-loop track ¼ mile to 1 mile in length, and the goal is to complete as many laps as possible in 60 minutes while using less than 1000 Wh of energy. Electrathon is a nation-wide organization, and there is an active league in Tampa. Races are held throughout the year, typically hosted by high schools. Teams are scored in four separate classes based on education level - high school or "open" (which consists of college students and hobbyists) - and battery type - lead-acid or lithium-ion. Electrathon racing arrived in the United States in the 1990s and has historically used lead-acid batteries, but lithium-ion batteries are now standard in EVs, so the Advanced Battery class was created to accommodate them. We have chosen to race in this class because lithium-ion offers greater performance and is more relevant to future EVs. Besides the battery, other constraints and rules are given in the Electrathon Handbook [1].

The scope of this project does not include the mechanical frame, steering, or suspension; those have already been built. Rather, the project goal is to optimize the electrical system, and the team consists of two electrical engineering majors and two computer engineering majors. The most binding requirement is that rated capacity of all battery cells in the pack must be less than 1000 Wh, and every other electrical component on the vehicle should work towards maximizing the fraction of that energy that is delivered to the wheels. Batteries deliver peak energy under certain conditions, and motors reach peak efficiency under certain conditions, so a major part of the project is finding those optimal operating points. The rulebook dictates that Electrathon cars be controlled by a driver, and since the driver is responsible for keeping the car in an optimal state, the car must quickly present the driver with detailed and accurate information.

This project has three main components. First, a typical Electrathon race day consists of two races, one starting at 10 am and one starting at 1 pm. Since the races are 1 hour long, this leaves slightly less than 2 hours between races to recharge the car's battery pack. Most teams bring two batteries, but this is very expensive. Electric vehicle fast chargers have been proven to bring a vehicle from 0 to almost 100% in this amount of time, and the battery pack is small enough that this can be done from only a standard 120V 15A outlet. Battery chargers in this voltage and current range are not available, so the first element of the project is building an AC-DC converter with a Constant Current Constant Voltage (CC/CV) regulated output such that it can be used as a DC fast charger.

The second component of the project is building a custom motor controller that is optimized for maximum efficiency, the most important specification in an Electrathon car. Off-the-shelf motor controllers in this power and voltage range are not suitable because they are intended for Radio Control (RC) and electric bike applications where cost and weight are more important. The third element is building a Driver Information System that gathers data from sensors and presents it on a screen. The rulebook bans driver assistance features that would control the car directly, so this system will only present information and record it for later analysis. Specific goals and objectives for these systems are outlined in the next section.

2.2 Goals and Objectives

Table 2.1 outlines project goals at the "basic", "advanced", and "stretch" levels. Basic goals are minimum requirements the team feels confident can be accomplished within the given time, financial, and technical constraints of the project. Advanced goals are aspects of the project that the team will work to achieve once all the "basic" goals have been accomplished. These include additional features that will further enhance vehicle performance or contribute to the display output. Stretch goals are features that the team would like to implement, but are likely to be outside the scope of the project.

Table 2.1 - Goals and Objectives		
	Hardware	**Software**

Basic	Build custom DC-DC converters to supply power from battery pack to 15V, -15V, 5.5V, and 3.3V rails	In-car display including current and average speed, average power consumption, time left in race, and button to start/stop race
Advanced	Motor controller with greater efficiency than typical RC controller, finish at least one Electrathon race	Ability to send data to and from car, button for lap counting either in car or from pits with error checking in case a user misses a lap, blindspot monitor, "settings" page for display
Stretch	Win a Tampa Bay Electrathon race, Custom 1800W DC fast charger to recharge battery between races	Automated lap counter to avoid errors if users miss counting a lap, real-time location tracking, screen rotation capability

At the basic level, the car must be able to complete an Electrathon race while meeting all of the requirements specified by the Electrathon handbook. The battery capacity must be under 1000Wh, but still close enough to maximize the energy available. The batteries need to last the entire race without being charged. This goal is largely dependent on the driver, who must budget energy to make the battery last throughout the race while also striving to complete as many laps as possible with that energy within the 60-minute time limit. Without any feedback on the car's status, the driver could greatly overestimate or underestimate how much battery life they have left. In the former case, the driver might be able to complete a lot of laps early on in the race by driving at the car's maximum speed, but risk having the car stop part way through the race when the battery runs out. Technically, the team would *not* be disqualified if this happened, and could theoretically even win the race if the car manages to complete more laps than any other team before it stops, despite not actually continuing for the entire race. Still, this is not an ideal strategy. In the latter case, the car could finish the race with a significant portion of their battery life remaining by driving slower than necessary, and consequently not complete as many laps as it would have if driven at a faster speed throughout the race.

To avoid these two cases while still complying with the Electrathon handbook (which does not allow autonomous driving), a display will be included in the car. At a minimum, this display will include how much time is left in the race, along with the current and average speed of the car and average power consumption. A start/stop button must also be part of the implemented design, because otherwise the race timer would start as soon as the power to the display is turned on.

If time and budget constraints permit, the next step will be further optimizing the car's motor controller and adding additional features to the display. One "advanced" goal is to send data to and from the car so that team members in the pits can view the same race information the driver sees.

At Electrathon races, it is common to hear high school coaches and teammates shouting to their driver, “Use your mirrors!” Although smaller and generally slower than full-size cars, many Electrathon vehicles can still reach speeds of 30 miles per hour or more. Thus, another advanced goal is to implement safety features, such as a blindspot monitor to prevent collisions. Some designs, for example, include this feature in the form of LEDs attached to the car’s side mirrors that blink if another vehicle is too close.

A settings page for the display may be a nice feature to have. This could include visual options such as customized font and color themes, enabling/disabling certain display notifications (such as warnings when there is limited time left in the race or energy use is “too high”), choosing how to display information to the screen (for example, if the driver wants the timer to be displayed in large font at the top of the screen or tucked away in a corner, depending on how much they care about the amount of time left in the race), or choosing a track from a list of race locations. The last option could help estimate the number of laps completed based on the car’s average speed over time, as different race locations have courses of varying lengths.

A lap counter would be useful to help with energy budgeting. This could be implemented as a button pressed by either the driver or a spectating team member if they are able to connect remotely. As a stretch goal, it would be more ideal to design a method of automatically counting laps, and to be able to track the car’s location in real time. On the hardware side, stretch goals include designing a custom battery management system for more precise charging, and winning a Tampa Bay Electrathon race.

2.3 Specifications

Project specifications are shown in table 2.2. Of those listed, the three chosen for demonstration are Speed Measurement Accuracy within 10%, Current and Voltage Measurement accuracy within 5%, and System Response Time within 1 second, highlighted in orange.

Table 2.2 - Required Specifications		
Powertrain	**Battery Capacity (0.1C)**	1000Wh, +/- 5%
	Battery Weight (cells)	< 15lbs
	Motor Controller Peak Efficiency	> 90%
User Interface	**Speed Measurement Accuracy**	+/- 10%
	Current & Voltage Measurement Accuracy	+/- 5%

	Display Response Time	1 second
Battery Charger	**Maximum Voltage**	63V +/- 5%
	Maximum Current	36A +/- 5%

2.4 Related Work

This section outlines three examples of previous works related to this project. The first two are examples of other Electrathon teams in both the high school and hobbyist categories, while the third example is a previous senior design project, also featuring a go-kart but designed for improved driver experience rather than raw performance.

High School Electrathon Teams: Building an Electrathon car can be a valuable learning opportunity for high school students, especially those planning to major in mechanical or electrical engineering after they graduate. Middleton High School in Tampa is a STEM-focused school with multiple active Electrathon teams. Plant City High School also regularly participates in and hosts Electrathon races.

Electrathon cars built by high school teams are usually fairly simple in design - at this level it can be considered an accomplishment to get the car running at all. Examples of Tampa Bay Electrathon cars are featured on the Electrathon of Florida website: https://electrathonofflorida.org/.

Cliff: Consistently a local champion of Electrathon races, Cliff has a webpage dedicated to his award-winning vehicle: https://proev.com/P2Desc.htm. “ProEV”, as it is titled, started from a Blue Sky Design Coupe and has been heavily modified to achieve maximum racing performance. Cliff’s design is highly compact, with just enough space to fit himself. His battery pack consists of nine modules totaling 999Wh. He tests each module to get an ideal combination of stronger and weaker cells. During races, he uses voltage readers to alert him (via loud buzzers) when a cell is too low.

The ProEV’s software includes an Arduino Mega 2560 and an OLED display. The display is attached to the vehicle’s ceiling, just above the driver’s line of sight between two rear-view mirrors. Similar to the goals of this project, it shows information such as speed, energy use, and time in the race. Many sensors are included in the car for enhancing the driver’s ability to budget power throughout the race.

Heart Racer: In 2015, a senior design group of 3 electrical engineering majors built the “Heart Racer Go-Kart”. The website for the project can be found at https://www.ece.ucf.edu/seniordesign/su2015fa2015/g19/. The Heart Racer was intended to improve the driving experience of a standard go-kart. Its design included lights, speakers, and a sensor that measured the driver’s pulse in order to synchronize the changing colors of the lights to their heart rate. Since the Heart Racer was built for racing, performance was valued. However, contrary to the project proposed in this paper, it was clearly not optimized to reduce power consumption or cost.

2.5 House of Quality

Figure 2.1 shows the House of Quality for the project, listing the relative importance of various features the group seeks to implement in the design.

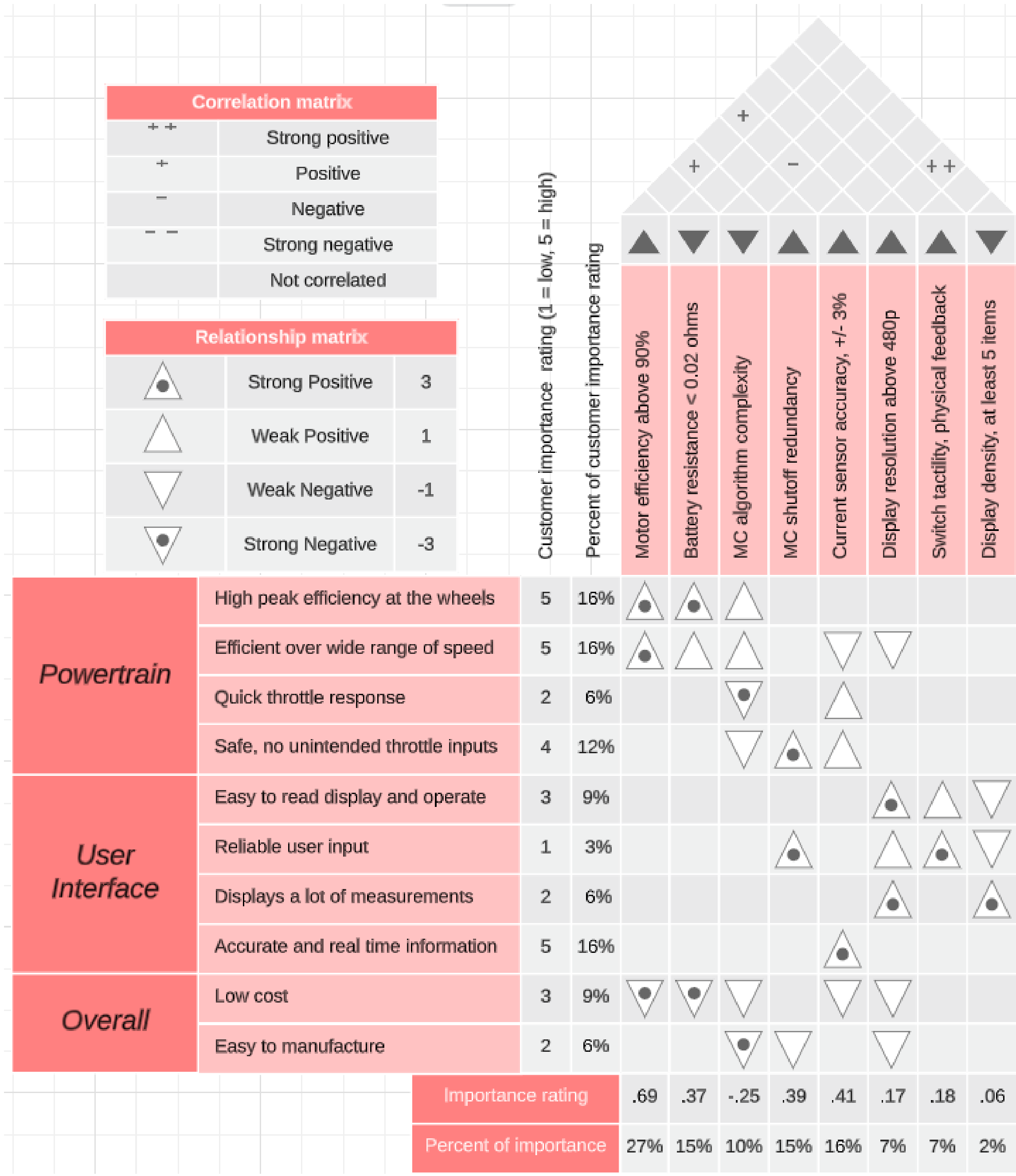


Figure 2.1 - House of Quality

2.6 Block Diagrams

This section includes tentative hardware and software block diagrams for the project, along with work distributions among group members.

2.6.1 Hardware Block Diagram

The hardware block diagram is shown in figure 2.2.

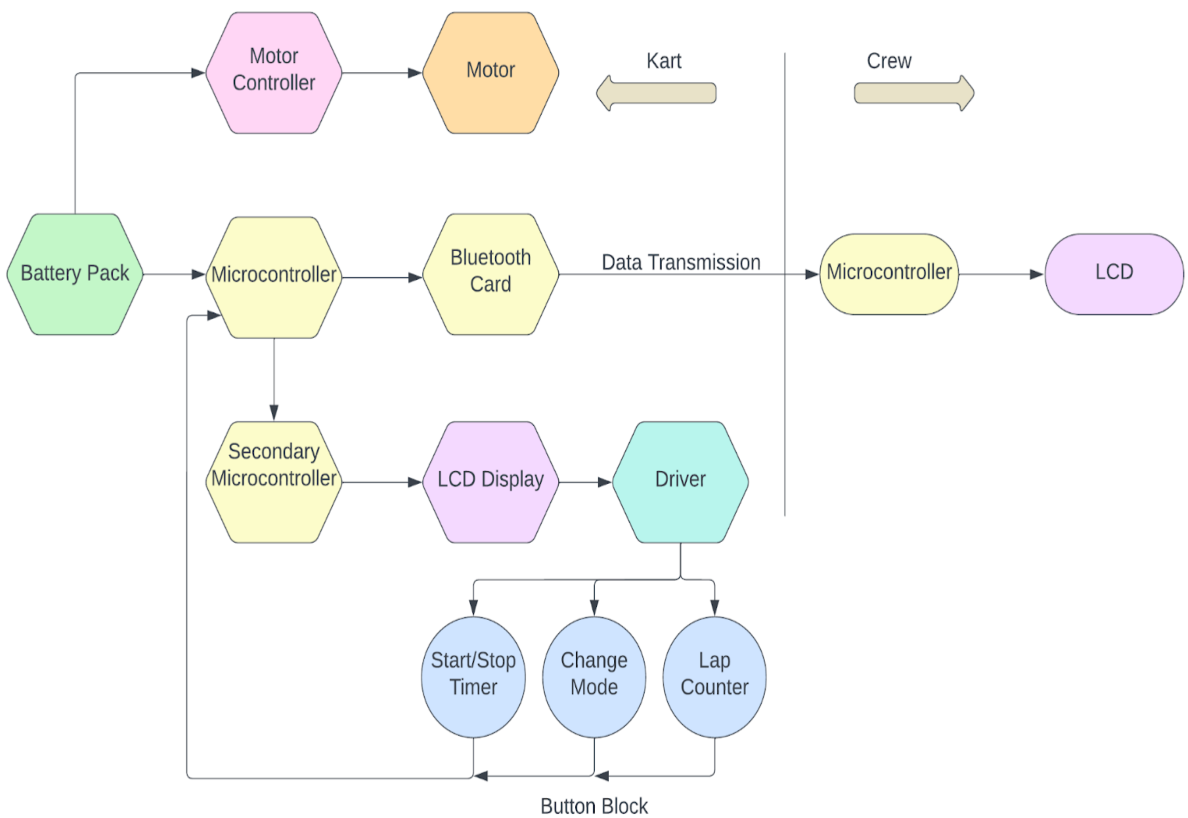


Figure 2.2 - Hardware Block Diagram

2.6.2 Software Flowchart

The group discussed that it would be most logical for Julian and Abdullah (electrical engineering majors) to work on the electrical components of the car, while Fouad and Grace (computer engineering majors) program the microcontroller and display.

An initial software design includes two microcontrollers: a "primary" microcontroller connected to various sensors, and a "secondary" microcontroller dedicated to the motor controller. The primary microcontroller will read sensor data and perform any necessary calculations, then send the results to the display. The motivation for having a dedicated microcontroller for the motor controller is its more robust power and performance requirements compared to the display and associated data analysis.

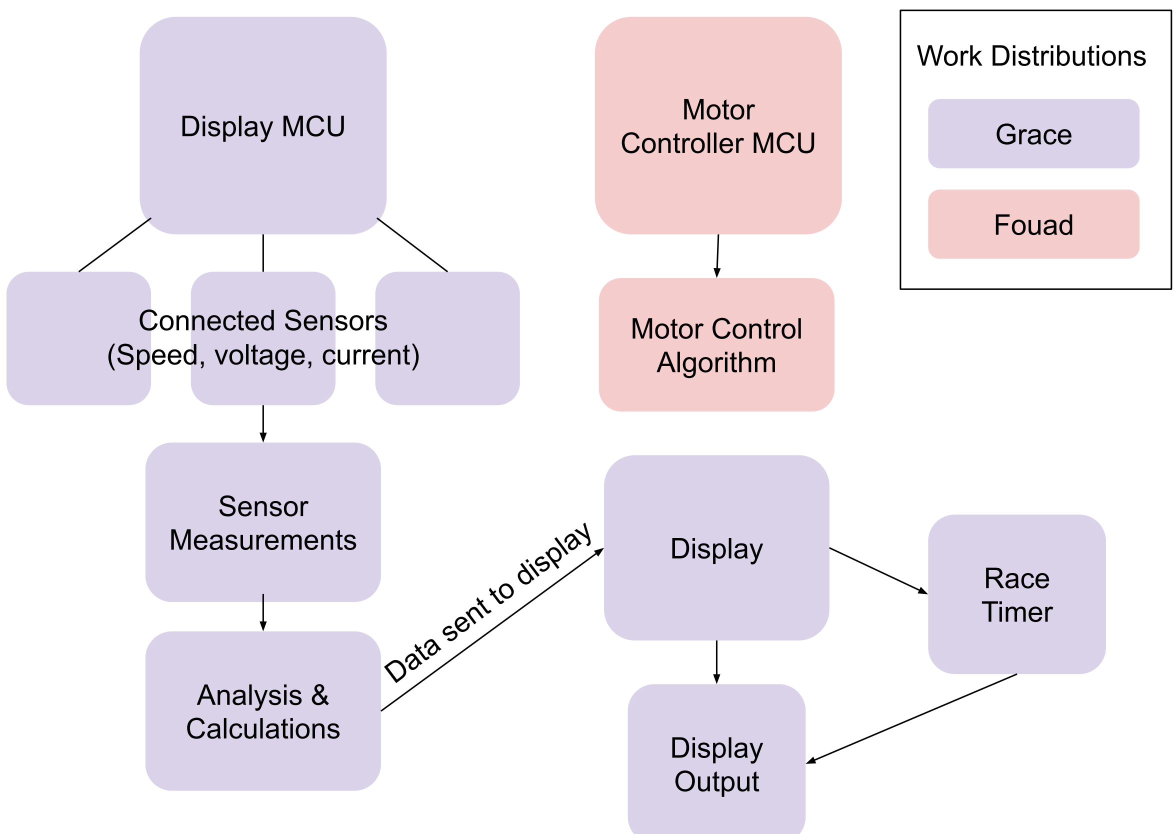


Figure 2.3 - Software Flowchart

3 Research and Part Selection

This chapter contains research comparing different technologies that could be used in the project's major components. Possible technology solutions from both the hardware and software perspective are compared. Once an approach has been chosen, an appropriate part is selected for implementing each technology.

3.1 Technology Comparison: Hardware

This section compares potential hardware solutions for the project. In section 3.1.1, two types of motors - Brushed DC Motors and Brushless DC Motors - are compared, while 3.1.2 compares battery types, 3.1.3 compares material types for transistors, and 3.1.4 compares material types for power capacitors. Tables 3.1, 3.2, 3.3, and 3.4 show the group's considerations when selecting which technology to use.

3.1.1 Comparison of Motor Types

In our Electrathon Senior Design Project, we opted for the use of a DC (Direct Current) Motor. The origins of DC Motors can be seen as far back as the early 19th century. The experiments carried out by Michael Faraday in regard to the connection between electricity and magnetism were what led to the development of electric motors. Thomas Davenport was the one that created the first practical DC Motor in the year 1830. Later on, we saw more pioneers such as Thomas Edison that improved upon the ideas and made the DC Motor more efficient, leading to their widespread use commercially [2].

DC Motors have a very large range of applications, going from smaller use applications to larger scale applications. One of the most common utilizations we can see today is majorly in electric transportation such as powering electric vehicles, trains, and boats. They have the ability to provide high torque at low speeds.

Roughly all DC motors make use of an internal mechanism, which can either be electromechanical or electronic. DC motors are fundamentally made up of two parts: a rotor and a stator. The stator has coils of wires that are conducting electricity, which lead to the creation of a magnetic field. The rotor is positioned inside this magnetic field; it is usually a permanent magnet or another coil. The Lorentz force is produced when a current flows through the rotor and interacts with the magnetic field the stator creates. The rotor rotates as a result of the torque this force applies to it. The commutator is what is utilized in this process since the direction of rotation is dependent upon the direction of current flow through the rotor. When it reaches a specific location, the commutator, a split ring attached to the rotor, reverses the direction of current flow in the rotor coil. The rotor rotates continuously as a result of this current reversal, maintaining spinning. To put it simply, the commutator makes sure that the rotor's magnetic force is constantly directed in the same direction, which produces unidirectional spinning.

By varying the voltage supplied or by adjusting the current that flows through its field windings, the speed and the torque of the DC motor can be varied. The voltage applied is directly proportional to the rotation of the rotor within the motor. The motor accelerates when more voltage is delivered and decelerates when less voltage is applied. Varying the strength of the current supplied has a directly proportional relationship with the torque that the motor provides. By increasing the current, we are able to increase the torque produced by the motor. Because of these features, we are able to precisely control the speed and torque, which makes DC motors appropriate for applications needing high precision and varying speeds.

There are many types of different DC Motors, we are only going to go over a few, in the interest of maintaining relevance to the project.

First off, we are going to look at Brushed DC Motors. The basic principle behind a brushed DC motor which is quite obvious from the name is that they make use of electrically conducting brushes that are connected to the commutator, allowing for a connection to be made between the power supply and the armature. Brushes are a term used to refer to a small component that is usually made out of carbon.

Now we are going to look at the characteristics of a brushed DC motor.

The brushed DC motor is quite a simple form of DC motor technology. They are capable of periodically reversing the direction of the current in the armature coil, which can allow us to change the direction of motion of the motor. These motors also have a unique need for maintenance as the brushes within the motor can be worn down over time and would need to be replaced [3]. In addition, the contact between the brushes within the motor and the commutator can cause unwanted electromagnetic interference.

We can now take a look at some of the pros and cons of a DC Motor, which can lead to understanding why it was a viable technology option that the team considered for our project.

First, we can look at some of the benefits. The design of these motors is quite simple and straightforward, which in turn makes them quite simple to use and also cost-effective. In addition, these motors are able to provide a high starting torque, which is a great benefit when considering the application we needed it for within our project, as it can lead to a higher acceleration capability of the kart. Last but not least, these motors have a compact design, which makes it suitable as it does not take up too much space on the kart and also allows us to have a lower overall weight.

Next up, we can take a look at some of the drawbacks of using a brushed DC motor. The first one to mention is obviously the maintenance that can arise from the wearing down of the brushes within the motor, which in turn can also lead to a lower lifespan of the motor if the commutator that the brushes are attached to gets damaged. Furthermore, the electromagnetic interference that can arise from the contact between the brushes and the commutator might end up causing an issue for us. These motors also have low efficiency, as the friction between the moving parts contributes to energy losses in the form of heat generation. There is also the issue of a voltage drop that can occur due to the resistance of the brushes which can lead to a consumption of energy. They also can cause the generation of carbon dust, which is not environmentally friendly.

After considerably taking a look at Brushed DC motors, we decided to move on and consider the possibility of using a Brushless DC motor.

The brushless DC motor utilizes electronic controllers to switch the current being supplied to the motor windings that in turn produce the magnetic fields. The electronic controller can then vary the amplitude as well as the phase of the DC current, which allows us to vary the speed and torque that the motor provides. The electronic controller can reverse the direction of the current at specifically the right angle so that the electromagnetic field is able to generate the torque for the motor [4].

We can now go ahead and look at the pros and cons of using a brushless DC motor.

The benefits of the brushless motor can be started off by mentioning that these motors have a much higher efficiency due to the fact that the absence of physical contact by the brushes allows the motor to lose less energy and generate less heat. In addition, these motors have a longer lifespan, because they do not consist of brushes that can be worn down over time and are able to provide consistent performance with lower maintenance. They provide a more precise control over the speed and the torque because of the accuracy of the electronic controller. Furthermore, they have a higher power density and a higher power to weight ratio. They also have a reduced amount of electromagnetic interference that would hinder the other electronic components.

Table 3.1 - Comparison of Motor Types

	Brushed DC Motor	**Brushless DC Motor**
Cost	The brushed DC motor allows for a lower initial cost but with time would need the brushes to be replaced and more maintenance costs.	The brushless DC motor is a pricier purchase from the get-go, but with its low maintenance costs, we believed would be beneficial for us down the line
Efficiency	The constant contact between the brushes and the commutator leads to considerable energy loss in the form of heat, which in turn leads to a lower overall efficiency.	The absence of the brushes in these are the reason why they are able to provide better efficiency, which is fairly crucial to us in our project.
Lifespan	Lower lifespan due to the commutator possibly being damaged by the worn-down brushes.	Higher lifespan as the only parts that could be prone to being worn or damaged would be the bearings within.

Complexity	These motors are the fairly simpler option to use.	These motors when bought with the sensor driven technology are quite complex.

Now let's take a look at some of the cons of utilizing a brushless DC motor. First off, these motors are more towards the pricier side and are quite complex, due to the fact that they have sensor-based control systems. There are options available in the market that do not utilize sensors, but those options fail to provide the same level of performance as the ones with sensors.

After going through the options of different motor technologies we could possibly use for the design of our project. The group opted to go for Brushless DC motor technology, as even though it is a pricier investment, we felt that it would be more suited to our specific needs down the line.

3.1.2 Comparison of Battery Types

This section compares three possible choices of battery type to be used in the project: Alkaline, Nickel Metal Hydride, and Lithium Ion.

Alkaline Battery

The reason behind the name of Alkaline Batteries is that the electrolyte used in it has a pH value over 7. Alkaline Batteries account for about 80% of manufactured batteries in the U.S alone and over 10 billion units worldwide. In alkaline batteries Zinc and manganese dioxide are used as the negative and positive electrodes respectively, whereas the electrolyte is Potassium Hydroxide [5]. There are some versions of alkaline batteries that were designed to be recharged a few times as well, however they end up having a smaller capacity after each charge.

We can move on and talk about some of the characteristics of alkaline batteries. Alkaline batteries have a good capacity due to the manganese oxide being pure and dense, they have about 3-5x the capacity of an acidic cell. Depending on how high or low the cut-off voltage is of the device in which the battery is used, will determine the fluctuation of the actual battery capacity. In terms of voltage, the zero-load voltage is between 1-1.5v, which depends on how pure the materials used are and on the electrolyte. As far as current goes, the current is directly proportional to the physical size of the battery, as the internal resistance decreases and the surface area within the battery increases.

We will now cover some benefits and drawbacks of alkaline batteries.

As far as the pros go, one of the main things to consider is the price, these batteries are very cost effective for what they are, which goes a long way when using it for our senior design project as we do we have a specific budget in mind and do not want to go overboard with every selection we make. These batteries have a relatively high energy density as well, which is good in terms of weight concerns. In addition, these batteries are comparatively safer for the environment when compared to some of the other options available within the market today. They are also quite readily available almost everywhere, which is another fairly important argument in terms of the project.

Now for the cons, first and foremost we have to understand that these batteries are not usually rechargeable, therefore they are limited in terms of reusability. This is not cost effective at all and can definitely lead to an increase in waste. Furthermore, the voltage output of these batteries depletes over time, which means it will be quite ineffective when trying to analyze and gather results to make comparisons between different tests. These batteries are also notoriously known for having lower performance output in difficult weather conditions such as temperature. High temperatures can lead to leakages, which is another common flaw for these batteries, whereas colder temperatures can cause them to have a lower capacity than with nominal temperatures.

These batteries are very commonly used due to them being readily available and convenient, but due to their limitations they often fall short in terms of performance output when compared to some of the other possible categories.

Nickel Metal Hydride Battery

They are another type of rechargeable battery. Interest in these grew in the 1970s when nickel-hydrogen batteries were commercialized, but hydride technology was a better suited alternative due to the fact that it was easier to store compared to hydrogen. For the positive electrode in this battery, nickel oxide hydroxide is used and as for the negative electrode, a hydrogen absorbing alloy is used [6]. They were primarily used for smaller household items that required a recharging ability. They were also used as batteries for electric and hybrid engine cars for quite a while prior to being replaced by Lithium-Ion batteries.

Some of the characteristics of the Nickel Metal Hydride Battery are for one of course we know that they are rechargeable batteries. Some research does show that they lose some of their capacity when they are partially discharged but can be fixed by a few charge/discharge cycles. They are good to use in applications or equipment that requires longer battery life from a single charge. They have low internal resistance as well, which allows them to deliver a constant voltage supply until depletion.

We can now go ahead and take a look at some of the pros and cons of Nickel Metal Hydride Batteries.

First let’s take a look at the benefits of these batteries. We have to start off with the fact that these batteries are rechargeable, therefore they can be reused multiple times with low impact on their overall output whilst also making them cost effective. These batteries have a higher battery capacity than most batteries, allowing them to run for longer in between charges. They also have a relatively good energy density, making them suitable for moderate drain applications. These batteries, since they are rechargeable also have a relatively lower impact on the environment since you will not end up needing as many when compared to single use batteries, they also are not affected by the memory effect, which means they can be recharged at any time, without needing to fully deplete them first.

As far as the cons go, Nickel Metal Hydride batteries do have a relatively lower nominal voltage of just around 1.2 volts, which can be an issue in many applications. They also have a higher rate of self-discharge when compared to some similar technologies, which basically means they lose their charge at a higher rate when left unused for a period of time. Furthermore, these batteries have relatively slower charging times, which can only be dealt with if specialized chargers are utilized and that would be quite costly to do and would make it infeasible in terms of the budget for this project. Now as we mentioned in the pros, that these batteries can handle applications in which they will be connected to a moderate drain devices, brings us to the next drawback which is that in situations where they are connected to high drain devices, their overall performance plummets and they can not perform as good as some of the other battery technologies. Last but not least, these batteries also have the same problem, that in conditions that are not optimal, their performance takes a hit and they do not quite hit the mark and uphold the expected standard.

Lithium-Ion Battery

The third and final type of battery to be discussed are Lithium-Ion batteries. These batteries are also rechargeable and majorly replaced Nickel Metal Hydride batteries in a vast number of applications after their development in the 1970s. These batteries conventionally utilize graphite as the negative electrode, whereas a metal oxide is utilized for the positive electrode. The electrolyte used in these batteries is a lithium salt dissolved within an organic solvent [7]. Lithium-Ion batteries are now the majority used battery in various consumer electronics and electric vehicles. In addition to that, they are also used in various commercial, military and aerospace applications.

These batteries have a higher open-circuit voltage when compared to various aqueous batteries. The internal resistance will increase as time goes on and the battery is put through more and more cycles of charging and discharging. The increase in the internal resistance leads to a drop in the terminal voltage which in turn will lead to a decrease in the current drawn. These batteries have a high energy density which led to higher prices in the past, but with the development of technology, the increase in the energy density has now led to decrease in overall prices which makes them a very suitable candidate in today's market.

We can now go ahead and take a look at some of the benefits and drawbacks of Lithium-Ion batteries.

First let’s take a look at the pros. These batteries are very suitable for lightweight applications as they do have a high energy density and can provide a good output relative to their weight and size. They are also rechargeable which makes them cost effective and therefore a good option to be selected when pursuing a project such as ours. These batteries also have a lower self-discharge rate when compared to other battery technologies, which means they can hold their charge for longer periods of time even if they are left unused. They also do not suffer from the memory effect and can be recharged at any time regardless of whether or not the discharge cycle was completed or not. They perform relatively well when it comes to non-optimal conditions such as high or low temperatures, which is definitely a big positive.

Now let’s look at some of the cons. These batteries do have a limited lifespan, which means that there are only a set number of cycles of charging and discharging they can go through before their overall capacity is permanently affected. There are also some safety concerns linked to lithium-ion batteries, as they are susceptible to overheating and catching on fire which can be catastrophic. These fires are mainly caused by either manufacturing defects or misuse of these batteries. Charging up these batteries requires the use of a special charger which makes it cost ineffective and if the special chargers are not utilized it can lead to a drop in overall capacity and performance. They are also the cause of many environmental concerns, as their production requires the mining and processing of raw materials. And last but not least, these batteries also sometimes may experience a voltage sag as they discharge, which negatively impacts the performance of the devices that require a constant voltage supply to function.

In a nutshell, Lithium-ion batteries come with a lot of benefits that lead you to believe they are the obvious choice when compared to various battery technologies, in terms of their reusability, the fact that they are great at holding their charge and also that they perform well in sub-optimal temperature conditions, however, it is also crucial to keep in mind the drawbacks in terms of safety and environmental impacts that these batteries have.

We will now look at how each of these battery technologies compare against each other.

- First thing to note is reusability, the Lithium-ion and NiMH batteries have high reusability since to them being rechargeable, whereas Alkaline batteries are not rechargeable therefore they have low reusability.
- Secondly, we will look at energy density. Lithium-Ion batteries are the best in terms of energy density, which makes them better for use with higher drain devices, whereas Alkaline and NiMH batteries both have energy densities a bit lower making them used in lower and moderate drain devices respectively.
- In terms of Nominal Voltage Lithium-Ion has the highest with about 3-3.5 volts per cell, Alkaline comes in second with about 1.5 volts per cell and NiMH with the lowest at 1.2 volts per cell.
- Lithium-Ion batteries have the lowest self-discharge rate and can hold a charge for an extended period of time, Alkaline and NiMH both have a moderate self-discharge rate and are not very good at holding their charge over time.
- In terms of price, Lithium-Ion is the priciest upfront but makes up for it with its extensive reusability, Alkaline batteries are very cost effective upfront but due to not being rechargeable, they are pricey in the long run. NiMH batteries are moderately priced upfront and also in the long run.
- Li-ion batteries are the best in terms of performance under wider temperature ranges, whereas alkaline batteries perform the worst. NiMH batteries do not do well either in the extreme temperature ranges.
- Lastly, in terms of safety, Lithium-Ion batteries are prone to safety issues such as overheating and catching on fire if misused. Alkaline batteries are safer; however they are also prone to leakage when mishandled. NiMH are generally safer than Lithium-Ion batteries but can have issues when used incorrectly.

Table 3.2 - Comparison of Battery Types

Types of Battery Packs	Description
Alkaline	Best in terms of upfront cost and are generally mediocre in terms of all other aspects and characteristics and also are not rechargeable.
Nickel Metal Hydride (NIMH)	These were originally very good in terms of reusability but have since then been replaced by better technology in the form of Lithium-Ion batteries.
Lithium Ion	These batteries seem like the obvious choice for us to use in this project, in terms of their terminal voltage and especially their ability to do well in varying circumstances. They are slightly pricey, but the pros outweigh the cons.

3.1.3 Transistor Semiconductor Material Comparison

Power electronics circuits consist of passive and active components, and the most important active components are the power transistors. They convert logic level signals from microcontrollers and specialized Power Management Integrated Circuits (PMICs) into full scale voltages and currents. Passive components filter these pulses into a smooth output, so pulse shape and timing is important, and the performance of the overall power converter heavily depends on the capabilities of the power switch.

The first type of transistor to be invented was the Bipolar Junction Transistor (BJT), which consists of three layers of semiconductor with an alternating P or N type doping pattern. The high power input and output terminals are the collector and emitter, connected to the two outer layers, and the base terminal is connected to the middle. The middle layer is made very thin so electrons or holes shoot through from one side to the other, and the number that passes through primarily depends on the current through the middle terminal, especially in the active region. This is the region where the voltage across the collector and emitter output terminals, V_{CE}, is greater than 1.4V so the base to collector junction is reverse biased. While excellent for amplifying small signals, this causes high power losses as the output current flows across the V_{CE} voltage drop. In power switching applications BJTs are instead usually operated in the fully on state, called the saturation region. In this mode the base current is much higher, causing greater shoot through and a lower V_{CE} drop for a given output current.

The Metal Oxide Semiconductor Field Effect Transistor (MOSFET) was invented a decade after the BJT, and it has since taken over in nearly all applications. It operates on the principle of a capacitor whose electric field creates a channel, and by modulating the voltage on the capacitor, the channel width and resistance can be modulated. In terms of construction, the most common N-channel variant uses a bulk piece of P type silicon with N-channel Drain and Source terminals that power flows between. The top layer of silicon is oxidized to silicon dioxide, a dielectric insulator, and a metal layer is deposited on top. This forms the gate capacitor structure. When a positive voltage is applied to the gate, the resulting electric field pushes the holes in the semiconductor away from the oxide and creates a depletion region with an N type character. Power can flow through this channel of N-type silicon, and since there is no PN junction, the resistance can be made very low. Additionally, while BJTs require a base current to remain on, FETs only require that the gate voltage be maintained, and the parasitic current is negligible. The gate charge on this capacitor must still be charged and discharged when the FET is turned on and off, so it's important to minimize this capacitance if the FET is to be used at high frequencies. MOSFETs have taken over in digital logic applications due to their lower power consumption, and they are equally dominant in low voltage power switching due to their low voltage drop, but until recently BJTs were still used for high voltage switching.

Wide Bandgap Semiconductors
In recent years FETs have become more capable in high voltage applications due to the rise of Wide BandGap semiconductors (WBG). In solid-state physics, the semiconductor band gap refers to the energy difference between the top of the valence band and the bottom of the conduction band, and it determines the electrical conductivity with conductors having a small bandgap and insulators having a wide bandgap. It is desirable for undoped semiconductors to have a wide bandgap with low conductivity, then they can be doped selectively to increase the conductivity. Silicon Carbide (SiC) and Gallium Nitride (GaN) are the two most commonly used WBGs, and they have bandgaps of 3.4 electron-volts (ev) and 3.2 ev respectively, much greater than the 1.1 ev bandgap of silicon [8]. This allows wide bandgap power switching devices to operate at higher temperatures and electric field strengths. The former reduces the weight of heatsink required for a given power dissipation, and the latter improves the Figure of Merit (FOM), which is the product of the channel resistance ($R_{DS\text{-}On}$) and the gate charge (Q_G) [9]. Gate charge is important because it is a major determinant of switching losses, and switching losses are the main limit on switching frequency. Higher frequencies allow for smaller and cheaper capacitors and inductors because a given size of passive component has a stronger effect at higher frequencies. As the heatsinks also get smaller due to higher efficiency and hotter running transistors, the entire power supply shrinks to be remarkably compact, easily achieving power densities over 100 W/in^3 [10].

Reverse Recovery Losses
Another large advantage of WBGs is their lower or non-existent reverse recovery losses, which also allows for faster switching. Diode reverse recovery is the time that it takes for the diode to switch from a forward biased conducting state to a reverse biased blocking state. If the voltage across the diode increases too rapidly, a large reverse current can flow, decreasing the converter's efficiency and possibly overheating and destroying the diode. Recovery time results from stored charge carriers (electrons and holes) within the semiconductor material during the forward conduction period. When the diode is reverse-biased, these stored carriers need to be removed before the diode can block the reverse current effectively. This phenomena is different from output capacitance, which causes a similar but smaller effect where a reverse current flows as the voltage increases. Schottky diodes use a metal-semiconductor junction that has a lower forward voltage drop than a PN junction, and because it only uses minority carriers, it in theory has the added benefit of zero reverse recovery time. In practice these devices have output

capacitance that causes a soft recovery, high reverse leakage current, and a PN junction guard ring that becomes forward biased if the forward current is high, thus needing time to clear at turnoff. They only have lower power dissipation at low voltages of approximately 100V or less, so until recently PN silicon rectifier diodes were still used for higher voltages [11]. SiC diodes use a combination PN and Schottky structure that can reach high voltages and has a much shorter and softer reverse recovery than silicon, about 20 vs 40 nanoseconds [12].

Reverse recovery time is important to power supply design because it determines the maximum gate drive current and switching speed. In a buck-derived converter, the diode or low side switch conducts during the off time of the high side switch, which is the freewheeling time of the output inductor. With the exception of soft switching converters, which use complex resonant effects to halt the current, the high side switch must turn on while the low side switch or diode is forward biased and conducting. As the high side switch turns on the voltage across the low side switch or diode rises rapidly, and the high side gate drive must be slow enough to prevent a large reverse recovery current spike. The problem is that most switching losses occur during this transition time when the current and voltage in the device are both high, and the reverse recovery time limits how short it can be. To maintain a given efficiency, the transition time must be a given fraction of the switching cycle, so if it's limited to a certain number of nanoseconds, that limits the power converter to a certain cycle time, and thus a certain switching frequency. This is the fundamental "speed limit" of silicon power converters.

SiC diodes with faster reverse recovery improve the situation, but Gallium Nitride (GaN) is interesting because it has zero reverse recovery time, both in theory and in practice. Conventional FETs have a body diode between drain and source that allows them to conduct in the reverse direction as if there was an anti-parallel diode, and many applications add an external anti-parallel diode to improve power handling and reduce the voltage drop. This means that there will always be reverse recovery in the low side FET. High Electron Mobility Transistors (HEMTs) made with GaN do not have a body diode, and yet they are still capable of reverse conduction. For a conventional enhancement mode device, E-HEMT, the channel is off when the gate is at zero volts with respect to the source, $V_{GS} = 0$, and V_{GS} above the threshold voltage turns it on. When the device is reverse biased and the drain voltage goes negative, the drain acts as the source and the device will turn on if V_{GD} is greater than the threshold [13]. Interestingly, this threshold can be modified by placing a different voltage on the gate, pulling it more negative to make the channel harder to turn on, and more positive to make it easier. This applied gate voltage is measured with respect to the source, so it's consistent with the idea that under reverse bias conduction is ultimately determined by the V_{GD} voltage. The voltage drop across the device is the threshold voltage, modified by the gate voltage, plus the current times the channel resistance. GaN devices have a lower threshold voltage than MOSFETs at only 2.5V, but that still leads to a ~3V reverse conduction drop that is higher than diodes. In many applications this is more than offset by the lower parasitic capacitances and lack of reverse recovery, which together allows for very high switching frequencies in the megahertz range.

Comparison

As manufacturing volumes rise and costs fall, WBGs are being used across the power industry in everything from laptops and small USB chargers to electric vehicle chargers, inverters, and other drives. When researching power transistors for this project, we needed to decide whether to use WBG transistors. They are more expensive than silicon, but only by a few dollars, and the performance benefits are compelling, as shown in the table below [14]. Note that GaN has a large advantage in electron mobility, and SiC has a large advantage in thermal conductivity, but otherwise they are similar.

Table 3.3 - Semiconductor Material Properties

Materials Property	Si	SiC	GaN
Band Gap (eV)	1.1	3.2	3.4
Critical Field (10^6 V/cm)	0.3	3	3.5
Electron Mobility (cm^2/V-sec)	1450	900	2000
Electron Saturation Velocity (10^6 cm/sec)	10	22	25
Thermal Conductivity (Watts/cm^2 K)	1.5	5	1.3

The benefits of WBG become most significant at higher voltages and power levels, particularly for SiC since its higher thermal conductivity makes it easier to conduct heat out of the device. The most common device rating is 650V, and while this is perfect for power supplies with a 400V DC bus and electric vehicles with a 300-400V battery pack, it's far more than we need for our 50-60V battery pack. The industry is moving towards even higher voltages with 1200V devices becoming common for 800V battery systems, but as these devices proliferate, lower voltage models are finally also available. Transistors of similar size, cost, and technology have a similar Figure of Merit (FOM), so designers trade off voltage rating against resistance. Using a device with a much higher voltage than necessary would result in a higher resistance, so it would not make sense to use a 650V rated transistor in a 60V system. GaN is more popular in low voltage applications than SiC, and low voltage devices are now available.

In the motor controller on the Electrathon car, efficiency is by far the most important consideration. GaN was selected for the main power switches that use a Pulse Width Modulation (PWM) pattern to convert up to 63V and 50A DC into 3-phase AC for the motor. The determining factors in our decision to use GaN were device availability, the lack of reverse recovery losses, and an acceptable additional cost of only a few dollars per transistor. For the output diodes in the AC-DC power supply, their maximum repetitive voltage stress is at most 63V, so silicon schottky diodes work well. The main switches are exposed to the higher voltage 400V bus, but for the AC-DC power supply cost is more important than efficiency, and silicon is still the most economical option.

3.1.4 Power Capacitors Material Comparison

Capacitors are essential components in almost every circuit, but they are especially important in power conversion circuits because they stabilize voltage sources. If voltage goes too high it can cause insulator, dielectric, or semiconductor breakdown, and if voltage goes too low it can cause reverse current or device shutdown. Switch mode power converters move power between voltage levels by creating pulses of current, and something must absorb those pulses to limit voltage fluctuations.

It's important to remember that circuit elements are not ideal, and this is especially true for capacitors. They have a parallel leakage resistance that prevents them from storing charge indefinitely, and more importantly, they have a series resistance that dissipates heat when current passes, and they have a series inductance that reduces the capacitive reactance at higher frequencies. All capacitors have two metal current collecting plates that are connected to the terminals, and while the size and shape of these plates has some impact on the parasitic inductance and resistance of the capacitor, most of the properties are determined by the dielectric in between them. Capacitors made with different dielectric materials have different desirable properties and limitations, so it's important to use each one in the correct application [15]. The applications we are considering are the output capacitors and DC link capacitors in AC-DC power supply, and the DC bus capacitors in the motor controller.

Ceramic Capacitors
Ceramic capacitors use Class 1 or Class 2 ceramic as their dielectric material, and they are very commonly used because they are close to ideal in most respects. Particularly in the surface mount MultiLayer Ceramic Capacitor (MLCC) packages, they have a very low parasitic inductance that allows them to be effective at high frequencies. Class 1 materials have very low losses and are highly stable over temperature, voltage, and frequency [16]. They are used in applications where high stability and accuracy are crucial, such as in resonant circuit applications, timing circuits, and high-frequency filters. Class 2 materials have higher capacitance values compared to Class 1, but they have a higher dissipation factor and some non-linearity. They are used in applications where high capacitance values are required, such as in decoupling, bypassing, and filtering applications in electronic circuits. While Class 2 materials have high capacitance, the value is unstable and changes with temperature and DC bias. The change is more severe for cheap materials such as X5R, which has roughly half its rated capacitance when half the rated DC voltage is applied. The decrease is less severe with X7R and larger package sizes. Due to their compact size and low Equivalent Series Resistance (ESR) and Equivalent Series Inductance (ESL), they are effective at frequencies of up to 1 MHz and sometimes higher, so they are great for decoupling IC power pins and absorbing ripple current.

Aluminum Electrolytic Capacitors
Aluminum electrolytic capacitors use aluminum oxide as the dielectric and an electrolyte as one of the electrodes. The aluminum oxide layer forms on the positive electrode, which is etched for greater surface area, and the electrolyte transports charges to the

oxide [17]. This asymmetry makes the capacitor polarized, so voltage can only be applied in one direction. Reverse voltage will overheat and burn the capacitor, causing it to split open and vent the electrolyte. These capacitors are very widely used because they are cheap to manufacture and provide more capacitance per unit volume than most other types of capacitors. However, in most other areas their performance falls short of other types. Their ESR and ESL are very high, so they are ineffective as capacitors above 100 kHz, and ripple currents cause major heating. Their liquid electrolyte also makes them temperature sensitive and prone to aging whereby the electrolyte dries out over time. Their main advantage is high capacitance and voltage at a low cost, and they are best used where the ripple currents and frequencies are low.

Polymer Capacitors
Polymer capacitors are best understood as an incremental improvement over traditional electrolytic capacitors. In most respects they are the same, except their electrolyte is a conductive solid polymer instead of a liquid. This reduces the ESR and improves the temperature stability and lifetime [18]. They typically replace aluminum electrolytic capacitors in higher end applications such as motherboards, graphics cards, and PC power supplies.

Tantalum Capacitors
Tantalum capacitors use tantalum metal for the anode and an oxide for the dielectric, supported by a solid electrolyte that is created by pyrolysis. They have high stability and reliability and very high capacitance per unit volume, higher than even aluminum electrolytic capacitors [19]. Their capacitance value remains close to the rated value throughout their lifetime, and they last longer than even polymer capacitors. Their main downside is high cost, but they are still used in high value applications such as smartphones, tablets, laptops, and telecom equipment. As with other electrolytic capacitors, they are polarized.

Mica Capacitors
Mica capacitors use mica as the dielectric material. Mica is a natural mineral with low dielectric losses and high thermal stability. Mica capacitors are used in applications where high precision, stability, and low losses are required. Mica paper capacitors have high voltage ratings at a wide temperature range, and silver mica capacitors have exceptional precision and stability [20]. Mica capacitors are often used in power RF filters and oscillators, but they have limited use elsewhere due to their low capacitance values.

Polyester Film
Polyester film capacitors use polyester film, aka Mylar, as the dielectric material. They are stable, reliable, low cost, and have moderate capacitance values [21]. They are often used in audio, oscillators, and other medium frequency circuits.

Polypropylene Film
Polypropylene film capacitors use polypropylene as the dielectric material, and they are similar to other film capacitors in having high insulation resistance, high stability, and

moderate capacitance values. They are noteworthy because of their low dissipation factor of only 2 to 25 x10^-4 at 1 MHz as shown in the table below [22]. This is comparable to mica and Class 1 ceramic capacitors, but with much higher capacitance, allowing them to be practical in power supplies. The limiting factor on ripple current is capacitor heating, so these low losses allow polypropylene capacitors to pass high ripple currents.

Table 3.4 - Losses of Capacitor Materials

Capacitor Type	Relative Permittivity	Dielectric Str. (V/μm)	Typical Value (μF)	Dissipation Factor x10^-4	Notes
Ceramic Class 1	12 to 90	< 100	10^-6 to 1	10 at 1-MHz	Ex. NP0, P100, N33
Ceramic Class 2	200 to 14,000	< 35	10^-6 to 1	251 at 1-MHz	Ex. X7R, X5R, T5V
Electrolytic	9.6	710	1 to 47,000	100 at 120-Hz	Polarized
Tantalum	26	625	1 to 100	600 at 120-Hz	Polarized
Mica	5 to 8	118	10^-6 to 3*10^-3	4 at 1-MHz	
Polyester Film	3.3	470/220	10^-4 to 10	170 to 300 at 100-kHz	
Polypropylene Film	2.2	650/450	10^-4 to 102	2 to 25 at 1-MHz	

Selection

For the DC bus capacitors in the motor controller, minimizing the dissipation factor is paramount crucial to minimizing losses in the controller. The current from the battery will be DC, but the FETs will turn on and off at a frequency of approximately 10kHz. Whenever the low side FET is on and the current is freewheeling through the motor winding, the capacitors must absorb the full input current as ripple, and they must discharge the extra energy when the high side switches are on. The frequency of this ripple current is high enough that the minimum required capacitance is low, but the high ripple current needs to be absorbed with minimal losses. Optimizing for this goal, polypropylene film capacitors were selected.

In the AC-DC converter, efficiency is less important, and aluminum electrolytic capacitors are appealing due to their low cost. Their ESR also provides a damping effect that reduces ringing between the output capacitor and inductor. However, the power supply's output current is high enough that no practical amount of electrolytic capacitance could withstand the ripple current, so the output capacitance will be a combination of electrolytic for damping and Class 2 ceramic for ripple. On the 400V DC link between the Power Factor Correction (PFC) and main converter the ripple current is much lower, so electrolytic capacitors are practical. The purpose of DC-link capacitors is to smooth the imbalances between the constant load of the converter and the 120 Hz power pulses of the PFC. Due to the low frequency, the required capacitance is high, which works in favor of aluminum electrolytic, and that is the technology we selected.

3.2 Part Selection: Hardware

In this section, parts are selected for each of the hardware technologies chosen. First, in 3.2.1, a brushless DC motor is chosen from several available options. Next, 3.2.2 compares Lithium Ion battery choices, and a battery is selected for the project. 3.2.3 details multiple transistor models with GaN as the material of choice. Lastly, 3.2.4 compares capacitor models.

3.2.1 Motor Selection

The primary goal of Electrathon is to propel the vehicle as efficiently as possible, and most losses occur in the motor. We quickly realized that brushless DC motors are the most efficient, but there are thousands of models to choose from, and efficiency is not clearly advertised. To select the best motor, we constructed a simple model of motor losses and created theoretical performance curves for each motor under consideration.

There are three types of losses in a brushless Permanent Magnet Direct Current motor (PMDC) [23]. Resistive losses are the easiest to understand, and they're simply equal to the copper winding resistance multiplied by the square of the current, $P = I^2R$. Frictional losses are the bearing resistance and air resistance of the rotor, and these are usually minor. The third and most interesting category are the core losses, which are created by hysteresis and eddy currents. These losses are caused by the permanent magnets on the rotor, which must be very strong to interact with the magnetic fields caused by currents in the stator windings. Unfortunately, they also interact with the iron in the core, which must be there to efficiently channel the magnetic field between the permanent magnets and the copper windings. Much like electricity, magnetic field lines flow in loops, and "magnetic circuits" can be described in terms of voltage sources and resistances. Reluctance is the ratio of magnetomotive force (mmf) to magnetic flux, essentially the field over the flux density. Air has a much lower permeability and thus a much higher reluctance than iron, so in a magnetic circuit pieces of iron are wires, air gaps are resistors, and permanent magnets are voltage sources. The neodymium magnets used in the rotor are expensive, so to maximize flux density for a given magnet count there needs to be a closed loop of iron between the magnets and the windings. To accomplish this, DC motors are composed of an iron core that the copper windings are wrapped around, and the magnets spin less than 1mm away from this core, usually surrounded by more iron called the "back-iron" which completes the magnetic loop.

Hysteresis losses are energy that is converted from magnetic flux to heat when a magnetic material undergoes a complete magnetization cycle. They occur due to the energy dissipated when magnetic domains are reoriented, and they happen in motors when alternating magnets sweep over the core. When a conductive material is exposed to a changing magnetic field, circulating currents known as eddy currents are induced, and they dissipate energy through resistive heating. Eddy currents are minimized by increasing material resistivity and separating the conductor into smaller pieces through lamination or powdered nodules. It is worth focusing on these losses because they are the main source of loss in large brushless DC motors at moderate to high RPM. Large motors of 3-10 kW often use 20 to 30 poles to maximize torque production, and they spin to about 10,000 RPM, so the electrical RPM and thus the frequency seen by the magnetic material is up to 300 kHz, which is fairly high by power electronics standards. Unlike 300 kHz rated transformers and inductors, these cores usually do not use ferrite; they use laminated steel sheets to reduce costs. Thicker sheets are cheaper to manufacture but have much higher eddy current losses than thinner sheets, so that is an important point of comparison.

Modeling

When a PMDC motor spins, the voltage applied to the windings is opposed by the winding resistance times the current, by the winding inductance over short timescales, and by the back-EMF induced in the windings by the changing magnetic field. The back-EMF is proportional to the motor RPM, and the current in the windings is proportional to the load on the motor. With no load, the motor will spin to an RPM equal to the applied voltage times a certain constant, and this is called the Velocity Constant K_V (or kV). In order to calculate motor efficiency from the limited data available, we added one more assumption to this common model of motor performance. We assumed that hysteresis, core, and frictional losses are proportional to RPM, and they can all be measured together by the no load current test. This consists of applying a given voltage to a motor that is spinning on a testbench with nothing connected to the shaft and measuring the load current. An ideal motor would draw zero current since any current that is drawn is not being used for useful work, and it's assumed that when the motor is under load, the current being drawn is the useful current plus this wasted current. This test is very easy to do, so it's one of the few data points that is always available. By multiplying the applied voltage and the velocity constant, we find the RPM that the motor was spinning at, and by multiplying the voltage and no load current we find the wasted power. Wasted power is assumed to be proportional to RPM, so this gives us a Watts/RPM constant for later equations.

A key figure of merit for a DC motor is the Torque Constant K_T, the amount of torque produced by 1 amp of current in the stator windings. In SI units of Newton-Meters per amp, this is inversely related to the velocity constant such that $K_T = 60/(2*3.14*kV)$. Knowing this, we can now calculate the current that would be drawn to make a given power at a given RPM, and from the current we can find the copper losses and total efficiency. Following the equations below, we start with Equation 1, which finds watts from Nm and RPM. This is transformed to Equation 3 that finds the winding amps (A) using the torque constant, and L represents the other sources of loss per RPM that were calculated earlier. This all goes into Equation 5, which plots the efficiency as a function of RPM. In order to generate a motor efficiency curve, all that is needed is the velocity constant kV, winding resistance R_M, and the no load current I_0 (at a specified voltage). The velocity constant only depends on how the motor is wound, but R_M and I_0 depend on the frame size. Larger frames have more room for copper but higher I_0, so they reach peak efficiency at higher power levels.

1. (W) = 0.105 * (Nm) * (rpm)
2. (Nm) = (W)/(rpm*0.105) = K_T*(A)
3. (A) = (W)/(rpm*0.105*K_T)
4. Losses = A^2*R + L*(rpm)
5. Efficiency = (P - [Losses])/P

Comparison
We began our comparison by looking at the Rotomax 150cc, a large frame powerful motor that's very economically priced. It has a low winding resistance but an inefficient magnetic circuit, causing no load losses 5-10x higher than other motors on the list. It becomes remarkably efficient at high power levels, but that may be because our model is neglecting some forms of loss. We compared the Rotomax to the CA120 because it is the only other large frame motor available, and online commenters suggested that it was better. According to their testing, that is no longer the case. The new CA120 uses 0.5mm laminations, much thicker than the 0.2mm laminations in the Rotomax, so it has an extremely inefficient magnetic circuit and enormous no load losses. When it was spun by another motor on a test rig to determine losses, the back iron became so hot that the glue holding the magnets melted [24]. We seriously considered several other motors, and their key characteristics are summarized in the following table.

Table 3.5 - Motor Performance Parameters

	Rotomax 150cc	Astro 3220	KDE7 215XF	CA120	Hacker A50-16L	Hacker Q100	Hacker A60-18L
kV (RPM/v)	150	137	135	150	265	110	149
R_M (ohms)	0.011	0.05	0.057	0.005	0.031	0.0106	0.02
I_0 (amps)	5.2 at 51.8V	1 at 50V	0.5 at 10V	13 at 20V	0.95 at 8.4V	1.86 at 8.4V	1.6 at 8.4V
Loss (W/RPM)	0.035	.0073	0.0037	0.087	0.0036	0.0169	0.0107
Weight (kg)	2.53	1.8	0.56	2.73	0.45	1.83	0.91
Peak Efficiency	93.1	94.8	95.7	90.8	94.3	96.1	94.9

In Electrathon the most important specification is efficiency, but we can't simply compare the peak efficiencies of different motors because efficiency varies a lot under different speed and load conditions. To determine this, we first plugged the parameters in the above table into the last equation in the Modeling section to plot efficiency vs RPM for a given power level. Plots were generated starting at 500W and increments of 500W initially, then 1000W at higher power, and the peak efficiency at each power was measured. At this point it was clear that large motors are NOT more efficient than small motors because of the higher drag of their magnets, so we looked at smaller motors with more efficient magnetic circuits. In our comparison the Astro 3220 was the "motor to beat" because it's a hand-assembled $700 motor that's used by the current dominant Electrathon racer in the Tampa Bay series. That price is a bit too high for us, and all the other motors we considered were approximately $300. A key question we could not conclusively answer was whether no load current should increase or be constant with RPM, and thus whether hysteresis losses are linear or quadratic with RPM. In the equations we assumed that they were linear, but they are most likely slightly more than linear, so it matters what voltage is applied when the no-load current is measured. The Rotomax and Astro were measured at 50V, the CA120 at 20V, and the other motors at approximately 10V. Their efficiencies are shown in the following table.

Table 3.6 - Motor Efficiency Curves

	Rotomax 150cc	Astro 3220	KDE7 215XF	CA120	Hacker A50-16L	Hacker Q100	Hacker A60-18L
η at 500W	84.1	91.3	94.5	78.9	92.8	92.1	91.5
η at 1000W	87.4	93.1	95.4	83.2	94.2	93.7	93
η at 1500W	89	94	95.7	85.3	94.2	94.5	93.9
η at 2000W	90	94.5	95.4	86.7	–	95	94.4
η at 2500W	90.7	94.8	94.9	87.6	–	95.4	94.8
η at 3000W	91.2	94.8	94.4	88.4	–	95.7	–
η at 4000W	92	94.5	93.1	89.4	–	96.1	–
η at 5000W	92.6	94	–	90.2	–	–	–
η at 6000W	93.1	93.3	–	90.8	–	–	–

The German company Hacker is widely considered to make the highest quality Radio Control (RC) hobby motors, although they are expensive. The A50 has a maximum power of 1600W and is thus a bit too small, and the Q100 is a bit too large and expensive, reaching peak efficiency at its maximum of 4kW. The Astro is larger still, maintaining good efficiency up to 6kW, and the driver who uses it operates it in a pulsed mode where it outputs 4kW bursts at the start of each straightaway. In Electrathon races the average power level is 1000W, but the driver must slow down for turns, so the most efficient driving mode is to use pulses at the start of each straightaway and coast for the rest of the time. In full size cars this is known as “pulse and glide,” and it is a common strategy used by hypermilers, people who drive solely to maximize fuel economy. This works in our application because the one-way clutch in the freewheel mechanism allows the motor to stop turning while the car coasts, eliminating hysteresis and eddy current losses for much of the race duration. Besides the efficiency of the motor, the optimal pulsing power also depends on the efficiency of the battery, motor controller, and wires at high load. Battery voltage falls significantly under load due to Equivalent Series Resistance (ESR), and resistance in the wiring and motor controller can also be significant at high currents, so a motor that is most efficient at 4kW or above would be a poor choice. At those power levels, losses in the battery and motor controller would be high, and it would be more optimal to use a smaller motor that reaches peak efficiency at a lower power level.

The other important factor in motor efficiency is RPM. Cars, trucks, and bicycles all have multiple gear ratios available, but in Electrathon the broad consensus is that the flexibility is not worth the time lost while shifting. Car or golf cart transmission would not fit, and bicycle derailleurs would not withstand the torque. With only one gear ratio available, that ratio must be set relative to the vehicle speed to ensure that the motor operates in an efficient RPM range. Within limits, the absolute motor RPM is not important because the gear ratio can be adjusted by changing the output sprocket size. In some Electrathon leagues, the cars normally race on mile-long oval tracks with very wide turns. In this case the cars maintain a nearly constant speed, so their motors operate at 1000W and close to their maximum RPM. The Tampa Bay league primarily races around cones in parking lots, and the turns are sharp enough that the cars slow down significantly, then they must accelerate at the exit of each corner to get back up to speed for the straightaway. The KDE motor is interesting because it is a drone motor, so it is optimized to drive a propeller, and those have a predictable load curve. The aerodynamic force on a propeller scales with the square of the velocity according to the quadratic drag approximation, so the power to turn it scales with the cube of the RPM. As the motor increases in speed, the load quickly rises from zero to an insurmountable wall of resistance, preventing the speed from increasing any further and ensuring that RPM remains in a narrow range, even with large changes in load. If the propeller is sized correctly, this narrow range of RPM is close to the maximum that the motor can reach, and the KDE motor is optimized to run in this high RPM range. It has a velocity constant of 135 RPM/volt, so at about 60V it redlines at 8,200 RPM. It has a very small core, which both reduces weight (important for drone applications) and reduces the hysteresis and eddy current losses, opposite to the very large Rotomax and CA120 motors. The downside of this design is that it does not leave much space for copper, so the wires are thin and the winding resistance is 0.057 ohms, the highest of any motor in the comparison. At high RPM the voltage is high and the current is low, so the high winding resistance is not important and efficiency can still be very high, but producing the same power at lower speeds requires more current, increasing losses dramatically.

If the Tampa league raced on oval tracks, the KDE motor would obviously be the best, but since the car needs to accelerate out of the low speed corners, it is actually a very close competition between the KDE7215 and the Hacker A60-18L. If we take the A60 as the baseline, the KDE motor has 3x higher winding resistance and 3x lower hysteresis losses, so it has excellent efficiency at high RPM, but efficiency is poor at low RPM unless the power is kept down to limit the current. Despite having only a slightly higher voltage rating and much higher resistance, it can reach higher power levels than the A60 because the open frame outrunner design has much better cooling than a sealed inrunner like the Hacker and Astro motors. Designed for a 32.5 mph top speed with 7,000 and 8,200 RPM redlines, the two motors should have 7000/547= 13:1 and 8200/547= 15:1 gear ratios. This will be done with a #25 chain on a 10 tooth motor sprocket and a 130 or 150 tooth output sprocket, custom made with Send-Cut-Send laser cutting because sprockets in this size are not available. The sprocket will have four 0.25” holes on a 2.559” circle to fit the bolt pattern of a freewheel with ISO standard 1.375” x 24 TPI threads, screwing onto the 20” BMX wheel.

- Wheel circumference in miles: 20*3.14/(12*5280)= 0.0009912 miles
- Maximum wheel speed: 32.5/(0.0009912*60)= 547 RPM
- Typical minimum: 14/(0.0009912*60)= 168 RPM
    - 168*13= 2,184 and 168*15= 2,520
- Typical maximum: 25/(0.0009912*60)= 420 RPM
    - 420*13= 5,460 and 420*15= 6,300

The plot below shows efficiency vs RPM at 2000W for the A60 in green and the KDE in red, and it’s obvious that the A60 has the advantage at low RPM. It is obviously not possible for efficiency to be negative, and in reality the graphs should stop when the motors reach their current limits. For the A60 this happens at 149*2000/90= 3,311 RPM, and for the KDE it happens at 135*2000/85= 3,176 RPM.

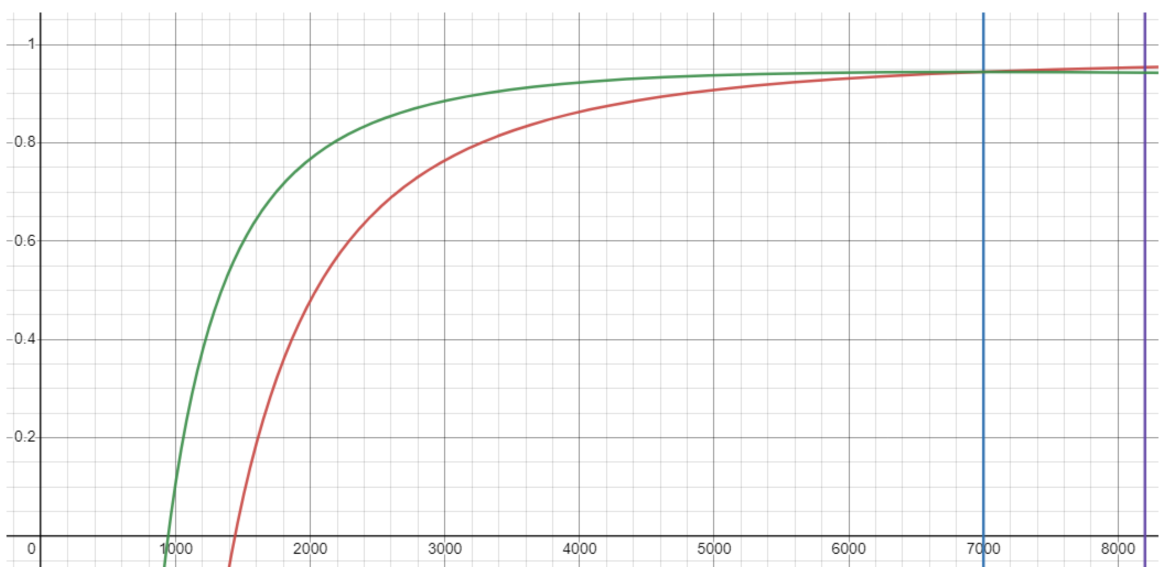


Figure 3.1 - A60 & KDE Motor Efficiency vs RPM at 2000W

A group member with experience watching Electrathon races estimates that the cars have corner exit speeds near 14 mph depending on driver skill, and at corner exit the driver will want to apply a burst of power. According to the calculations above, this speed corresponds to 2,184 and 2,520 RPM at the motor, so neither motor is able to apply 2000W at this speed, even with steep gearing. The lowest speed where they can is 3311/13*(0.0009912*60)= 15.1 mph, and at that speed we have efficiencies of 90.2% for the A60 and 85.1% for the KDE. The following plot shows the same curves at 1000W, and here the KDE does better, particularly at high RPM. At 15.1 mph we have 92.4% for the A60 and 91.5% for the KDE, only a 0.9% efficiency gap, and that swings to 1% in the other direction at maximum RPM. The crossover speed where KDE becomes more efficient is 4388/15*(0.0009912*60)= 17.4 mph. In summary, the A60 is more efficient when accelerating out of corners, and the KDE is more efficient on straightaways.

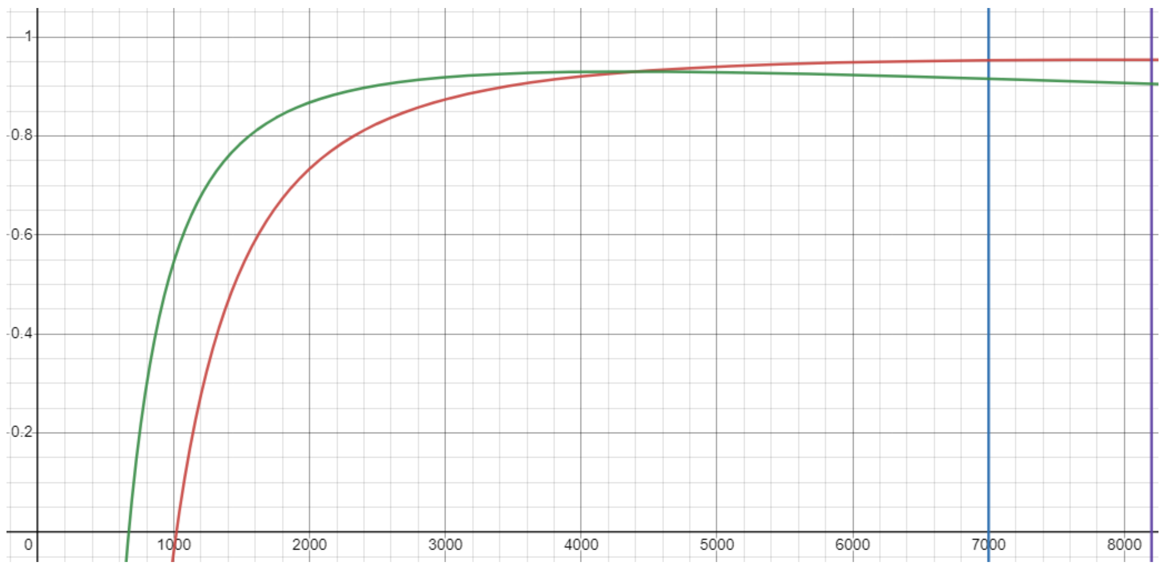


Figure 3.2 - A60 & KDE Motor Efficiency vs RPM at 1000W

With the data that we have, we can’t say conclusively which of the two motors is better. Motor winding resistance and no-load current often differs considerably from what is claimed, and fortunately those characteristics are easy to measure. Thus, the A60 and KDE motors are in a tie for now, and we will order and measure both. We will decide the winner by looking at updated efficiency models and return the other motor.

Testing showed 1.7A for the A60 and 1.73A for the KDE, so no advantage there
33.6 vs 76.8 mohms

3.2.2 Battery Selection

We decided that the best battery technology for us to use is definitely Lithium-Ion, due to its ability to perform well in suboptimal temperatures and low self-discharge rate. It was then time to decide which battery specifically we were going to use. The market nowadays is quite saturated in terms of options for which specific brand and their respective capacities.

The best configuration for us to use will be 15S (Series) x 9P (Parallel), with 2Ah batteries, which comes out to a total of 135 batteries.

We were able to obtain a data set online from Lygte that tested a very large set of batteries with varying milliamps per hour ratings, dimensions, brands, and models. The most interesting data set within their testing covers the capacity against the discharge rate at up to 5A for a 2.8V cutoff [25].

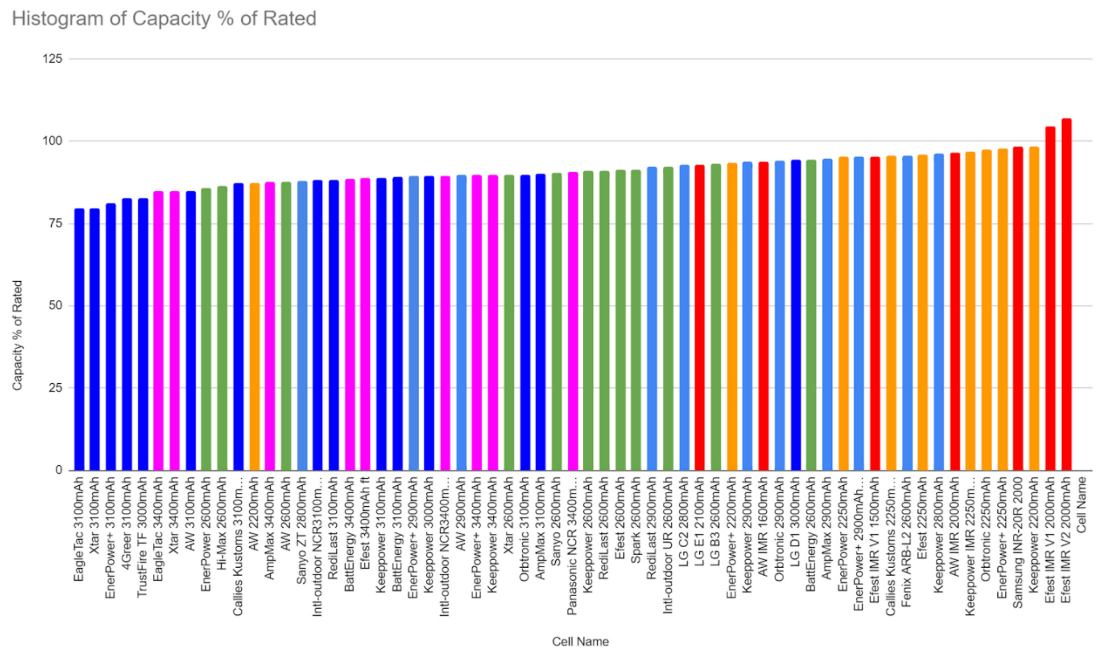


Figure 3.3 - (Tested Capacity) / (Rated Capacity) for 18650 Battery Cells

Here we can see a histogram showing the different batteries and how well they performed in terms of their rated capacities and how much they were actually able to output. In the above histogram the data is color coded in terms of their rated capacities. The red lines are 2000mAh, orange lines are 2200 mAh batteries, green lines are 2600 mAh batteries, blue lines are 2800 mAh batteries, dark blue lines are 3000 mAh batteries and the pink lines are the batteries that are 3400 mAh or above. This reason why this testing is helpful in our case is that it's being carried out at 5A, and the motor is most efficient near 2kW, which is a good power for the pulse and glide system, if internal resistance was not a factor, the race would consist of 50% at 2kW and 50% coasting, however, this reduces the usable capacity when compared to a more constant discharge. By restricting all the tested batteries to the same dimensions, we can also see that there is a direct inverse relationship between the capacity and the charge rate.

The Electrathon race has a 15 lbs weight limit for Li-Ion batteries, but this only refers to the cells, not the busbars, holders, and other parts of the pack. The testing results from Lygte shows that 18650 cells weigh between 45 and 50 grams, and if we use 50, 15 lbs is 1000*15/(2.2*50) = 136 cells. This means that each cell must have a capacity of at least, 1000/(3.7*136)= 1.987 Ah, so 2Ah cells are allowed.

High quality cells like the Panasonic NCR 3400mAh did better than their capacity would suggest, but in general the high-capacity cells held about 85% of their nominal capacity while low-capacity cells were 90-95%, and the Samsung 20R 2000mAh reached 98.4%. This is almost the highest measured and it also had the lowest measured internal resistance.

So as the data suggested, we decided to look into the Samsung 20R 22A 2000 mAh batteries. The unfortunate drawback we faced from using an older set of data was that majority of the batteries used in the test have been replaced by a newer technology, as in this case Samsung has put out the 20S battery to replace the 20R, so it was near impossible to purchase the 20R batteries from a reputable source.

The Samsung 20S battery is rated for 30A and was tested at a DC Resistance of 11 mΩ which is better when compared to the 40 mΩ that was used in testing the 20R.

We were able to find data online that showed the results of the tests for the 20S battery.

According to the tests carried out, the battery performed extremely well. At 0.4A, the Samsung 20S capacity was 2165 mAh, which is significantly higher than the declared 2000 mAh capacity. Even at higher currents all the way up to 20A. We can see that the battery's actual capacity was pretty much bouncing around the 2000 mAh mark, which is what it was rated for.

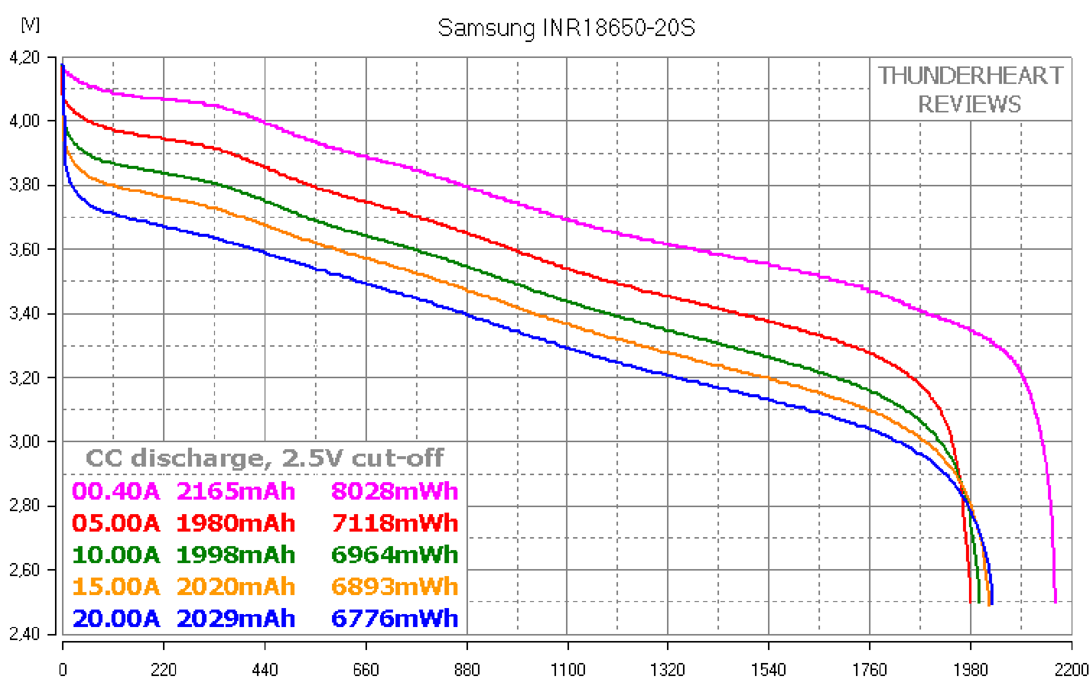


Figure 3.4 - Samsung 20S Discharge Curves at 0.4A, 5A, 10A, 15A, & 20A

Reprinted from ThunderHeart Reviews (Permission granted) (Appendix B) [26]

One of the key things to be pointed out from this graph, is that simply looking at the rated capacity of a battery can be misleading because it only measures ampere-hours, and the energy loss from high discharge shows up in the lower terminal voltage under load.

The only concern we faced with this battery was that it was a tad bit on the pricier end of things, at about $5 per cell, and with the requirement of 135 batteries, it would take the budget up quite a lot.

There were a few more options that we looked at in the category of 18650 batteries:

One of which was the BAK 2050 mAh battery. These batteries are priced at nearly as high as the 20S battery from Samsung, however their higher rated capacity is problematic for the battery pack design according to the Electrathon rules. The next one was the Queen Battery 2000 mAh 20A. Even though this is a high drain battery, the issue with this battery was that it could only be ordered from foreign websites and the datasheet they provided on the official website was adequate at best. Last but not least was the EVE 20P battery. This battery was very cost effective at only $2 a cell, however, test data reveals that it is quite handily beaten by higher capacity cells.

With the majority of the viable options in this category exhausted, we decided to move our testing to different categories to see if we could derive better results.

First off, we explored the possibility of using $LiFePO_4$ (Lithium Iron Phosphate) batteries. These batteries are a type of Lithium-Ion batteries that utilize lithium iron phosphate as the material for the cathode and a graphitic carbon with metallic backing as the anode. The trouble we ran into with this technology was that there were very few options available within the market, and even fewer when looking at the ones made for higher discharge applications. Headway was the only consumer-facing company that made high discharge LiFe batteries, and they had models at 8, 10, 15, 17 and 20 ampere-hours [27]. They claimed that these batteries have single digit mΩ internal resistances, however further reviews revealed that the internal resistance climbs up when the batteries reach a lower level of charge. This may seem like a decent option until you realize that these batteries have 4 to 5 times the capacity when compared to 18650 batteries, so their internal resistances should be lower, rather than the same. Therefore the LiFe batteries are not suitable for the Electrathon Project.

The next available technology was LiPo (Lithium Polymer) batteries. These are another sub-category of Lithium-Ion batteries, that utilize a polymer electrolyte in place of a liquid electrolyte. The conclusion drawn from the research of these batteries was that there is not a lot of information available online for this category which makes them harder to research. Though one thing is for certain that these are also on the pricier side, costing as much as the 20S batteries by Samsung, however, they do not last as long as them.

Therefore, after exploring the options and diving into the different technologies available and their prices and reviews. We made the decision to opt for the Samsung 20S 30A battery. Even though these batteries are on the pricier side of the market, they were the ones that had the optimal characteristics we needed to use for our project.

3.2.3 Transistor Selection

This section discusses the transistor selection for the motor controller and AC-DC converter with consideration to material type, current availability, and compatibility with the project design.

Motor Controller

Having established in the previous section that we will use GaN for the motor controller FETs, that did not leave many options to choose from. We searched for MOSFETs on the electronics website Mouser.com with the filters of "GaN" and "In Stock" and found only 70 results. All but eight of these were intended for 600V systems, which is not optimal because their resistance is too high, and of the eight, only four were different models. The four FETs are compared in the table below. Given that we plan to operate these at low frequency, $R_{DS\text{-}On}$ is much more important than gate charge, so to maximize efficiency we selected the third option, the GAN3R2.

Table 3.7 - Motor Controller Transistor Comparison

Model Name	Maximum Voltage (V)	$R_{DS\text{-}On}$ (milli-ohms)	Gate Charge (nC)
GS61004B	100	22	3.3
GS61008T	100	9.5	8
GAN3R2-100C	100	3.2	12
GAN7R0-150L	150	7	7.6

AC-DC Converter

In the main switches that operate the primary of the full bridge converter, the average current will be the same as the DC bus, 1800/400= 4.5A, but the pulsed operation will make the RMS current higher. These specifications are not very demanding, and searching for FETs rated to at least 600V and 10A continuous yielded 1,748 results, far too many to compare here. The IPD65R400 is rated at 650V, 15.1A, 0.4 ohms, and 39nC, which seem to be acceptable specifications. The package is TO-252, which can be easily soldered.

3.2.4 Capacitor Selection

This section discusses the capacitance requirements of the motor controller DC bus and AC-DC converter link.

Motor Controller DC Bus Capacitance

The DC bus capacitance in the motor controller must withstand high ripple current and minimize voltage ripple on this main power rail. The design conditions are a 55.5V battery pack, which is an intermediate voltage of 3.7V per cell, 50A to the motor, and a switching frequency of 10kHz where only $10V_{Pk\text{-}Pk}$ ripple is acceptable. Higher ripple voltage may cause torque ripple, and higher switching frequency would increase losses. The capacitance requirement is 500 µF, and this can be met by two 260 µF polypropylene capacitors, each rated to 23A ripple current. This is far more ripple current than an electrolytic of similar capacity could withstand.

- R_{Load} = 55.5/50= 1.11 ohms
- $V_{Ripple} = V_{Peak}$ / (f*R*C) = 55.5/(10000*1.11*10) = 500 µF

AC-DC Converter DC Link Capacitance

The limiting factor on the DC bus capacitance is withstanding the 4.5A of ripple current, although voltage ripple should also be limited to less than 50V. To meet these specifications with electrolytic capacitance, the ripple current should be distributed over as many capacitors as reasonably possible. Small 100uF 450V are cheaply available and rated to 1.56A ripple at 105°C, although it is recommended to use less ripple current for longer lifetime. Using 10 in parallel reduces the unit cost, limits ripple current to 0.45A on each capacitor, and keeps voltage ripple to only 37.5V.

- R_{Load} = 400/4.5= 88.9 ohms
- $V_{Ripple} = V_{Peak}$ / (f*R*C) = 400/(120*88.9*1000*10^-6) = 37.5V

3.3 Technology Comparison: Software

There are a few major decisions to make in regards to the project's software design at a high level. The choice of motor controller design affects what sort of microcontroller would be needed and how it would be programmed. The second main decision is how the display would be implemented. The group had two ideas for this, both of which involved sending data from a "central" microcontroller to another source. One option was a third microcontroller connected *physically* to an LCD; the other was a mobile app connected to the central microcontroller via bluetooth. Again, this decision affected which microcontroller(s) were best suited for the job, as well as the additional technologies that would be included in the design. For the first option, a microcontroller and display pair would be chosen, while for the option, there would be a choice of which app development tool to use.

For each microcontroller used in the project, an Integrated Design Environment (IDE) would need to be selected. Because some IDEs are more compatible with or even specifically designed for use with a particular brand/type of microcontroller, it was possible that up to three separate IDEs would be used for microcontroller programming. Although not a *major* concern, the desire to minimize the learning curve of learning multiple IDEs was motivation for choosing microcontrollers that could be used with either the same IDE or at least IDEs the programmers were already familiar with in the context of embedded programming. Similarly, for each application (motor controller and display), the team made a point to analyze whether a microcontroller they already owned and/or had prior experience with could be used.

Microcontroller Considerations

There are several factors to consider when selecting microcontrollers to use. Obviously, one of the minimum requirements of the microcontrollers chosen is compatibility with hardware used in the project, including the motor, motor controller, sensors, and display. Built-in bluetooth capabilities would be a useful feature, as one of the advanced goals of the project is remote connectivity. Ideally, team members in the pits will be able to connect remotely and see the same information displayed to the driver during the race. This way, the entire team could collectively analyze performance in real time and develop an improvement plan for future races. Built-in bluetooth would not be *required;* other microcontrollers would still be able to connect to bluetooth via cards. Still, having the feature already included would make the process easier.

Power conservation is one of the main focuses of the overall project, so it may seem natural to assume that this notion applies to the microcontrollers as well, and that low-power modes would be a desirable feature of the chosen microcontroller. While significant levels of unnecessary power usage by the microcontroller should be avoided, this is not a major concern in the scope of this project. Low-power modes are a fairly standard feature among microcontrollers, and can typically reduce idle power consumption from milliwatts down to microwatts. This can be critical for certain field applications, such as when the microcontroller is operating with a battery-powered device that must last multiple years in the field. However, in this case, the difference of milliwatts compared to milliwatts is not a major factor in the overall power consumption of the system.

In this case, it is more critical for the system to operate in real time and have sufficient processing power to meet the needs of the motor controller. For the motor controller, the two design approaches under consideration were Trapezoidal Control (more simple) and Field Oriented Control (more complex). The details of hardware choices regarding each of these approaches was described in further detail earlier in this chapter. For the purposes of software comparison, the main factor is processing power. Using Field Oriented Control (FOC) would improve efficiency, but requires a lot of processing power.

For the display, processing power is less of a factor. Here, the focus is real-time performance. Noticeable delays in output would cause frustration for the driver, limiting the usefulness of the measurements and display. Given the project's time constraints, user-friendliness, built-in libraries, and ample documentation are of value. Neither of the group's main programmers have extensive experience with embedded programming. To avoid wasting time overcoming steep learning curves, it is desirable for the microcontroller and its associated IDE to be fairly easy to program out-of-the-box and have adequate user support available.

For data-logging, the central microcontroller needs to have sufficient memory capacity to store information throughout the race. There is little need for a significant amount of Flash memory, as any data collected will only be used during the race and can be discarded after the system is powered off. Therefore, only the microcontroller's volatile memory capacity must be considered. How much memory is "enough"? It is best to be cautious and choose a microcontroller with more memory than the project is expected to need, but a great excess of memory storage beyond the project's requirements would not provide any additional benefits.

Personal Motivations

Beyond the technical requirements of this project, one of the indirect goals of Senior Design is to prepare engineers for their future careers. Consequently, an additional consideration of microcontroller selection is which products and IDEs are commonly used in industry. For example, while Arduino products are appreciated by students and hobbyists for being "easy" to program, they are typically less suited for industry applications.

Group members' career aspirations also impacted decisions on which technologies to use. A motor controller using Field-Oriented Control, for example, would be more mathematically rigorous and more complex to program for overall. For the display, some potential designs only required embedded programming, while others included app development. The programmer responsible for the display was skeptical about pursuing a career in embedded programming, and thought it would not make sense for her entire senior project to revolve around developing skills she did not intend to use. This prompted a shift from the group's initial plan to use an LCD, instead focusing on developing a smartphone app, which aligned more closely with the programmers' goals outside the project.

Time, Cost, and Learning Considerations

Hoping to save on costs by not buying a new microcontroller, the team first considered products already owned by group members. These include the MSP430g2553, ESP32, STM32, and several Arduino products - Aduino Uno, Nano, ATTinys, and Mega. All of these options offer low-power modes, though their other features differ. MSP430 products are designed for "ultra low-power" applications, particularly when operating with devices that run on battery power. The main draw of ESP32 is its built-in WiFi and Bluetooth capabilities, which would be useful for this project. STM32 has good peripheral options and can interface with LCD displays via LCD controllers. The Arduino Mega 2560 has an LCD specifically designed to be used with it - the HiLetgo 3.5" TFT display, which the group already owns multiple copies of. This display includes many built-in libraries for display output.

Both a perk and potential drawback of using a microcontroller designed for "beginners", such as products from the Arduino brand, is the removal of the "real" programming skills for embedded programming. In the case of the Arduino Mega with HiLetGo display, the programmer does not necessarily need to know how the display libraries work in order to use them. In theory, this is useful for quickly configuring and obtaining display outputs. This can speed up the development process if done responsibly.

However, because a proper understanding of microcontroller-display interfacing is not required, the programmer loses an opportunity for valuable skill development, and may become confused when attempting to customize the outputs generated by the basic library functions. This is especially true if the programmer is spoiled by a library in which the syntax can be easily “guessed at” and/or copied straight from a tutorial with slight modifications. Writing code lazily using this method may seem to save time in some cases, as there is less motivation to read documentation, but problems may arise when the programmer fails to fully understand the finer details of how their microcontroller operates *because* they neglected to read the documentation.

There are similar considerations regarding the choice of an app development software, if applicable. Some available tools offer “codeless” development, in which users are able to build their app from a template and/or “drag and drop” features (buttons, text, etc.) rather than coding them manually. More complex backend development is built into the software. These tools are commonly used by small businesses or hobbyists who want to build an app but don’t necessarily know how to or have time to write code for it, much like designing a website using Weebly or Google Sites instead of writing an HTML file.

Again, there are pros and cons to codeless app development. For basic applications consisting of “simple” user interactions, such as a few buttons and/or text boxes, these tools can save time, allowing the designer to create their app in a matter of minutes. However, with this approach, there is less of an opportunity to learn the skills necessary for an industry job in app development. Obviously, technology companies like Google and Microsoft do not hire engineers to “drag and drop” buttons onto a pre-coded user interface. Therefore, although codeless development could be a fast, “easy” solution, it may be in the team’s best interest to choose an app development tool that requires actual programming. This approach better suits the career goals of the primary display programmer, who has previous experience with backend development using Golang.

Figure 3.3 summarizes the technology decisions involved in the overall project. At the top level, the group first chooses an approach for the motor controller and display. Trapezoidal and Field-Oriented control included both sensored and sensorless approaches. Since sensored designs require a hall effect sensor, which the motor chosen for the project doesn’t have, these were excluded as possible choices. Based on the motor controller and display implementations chosen, the group chooses which microcontrollers to use.

A dedicated microcontroller will be used for the motor controller, and a “central” microcontroller will be used for the other sensors and display. Most microcontrollers could theoretically “work” for each case (for both the motor controller and display), but certain aspects of the designs make some types of microcontrollers more suitable than others.

The choice of central microcontroller is impacted by whether an LCD or smartphone app will be used for the display. If the LCD route is chosen, the group could either choose an LCD that connects directly with the central microcontroller, or add a third microcontroller specifically for the display, which would receive data from the central microcontroller. If the app route is chosen, an app development tool will be selected. In this case, there is more motivation for the central microcontroller to have built-in

Bluetooth features so it can easily send data to the app. Finally, the last step in the decision tree is choosing an IDE for programming each microcontroller.

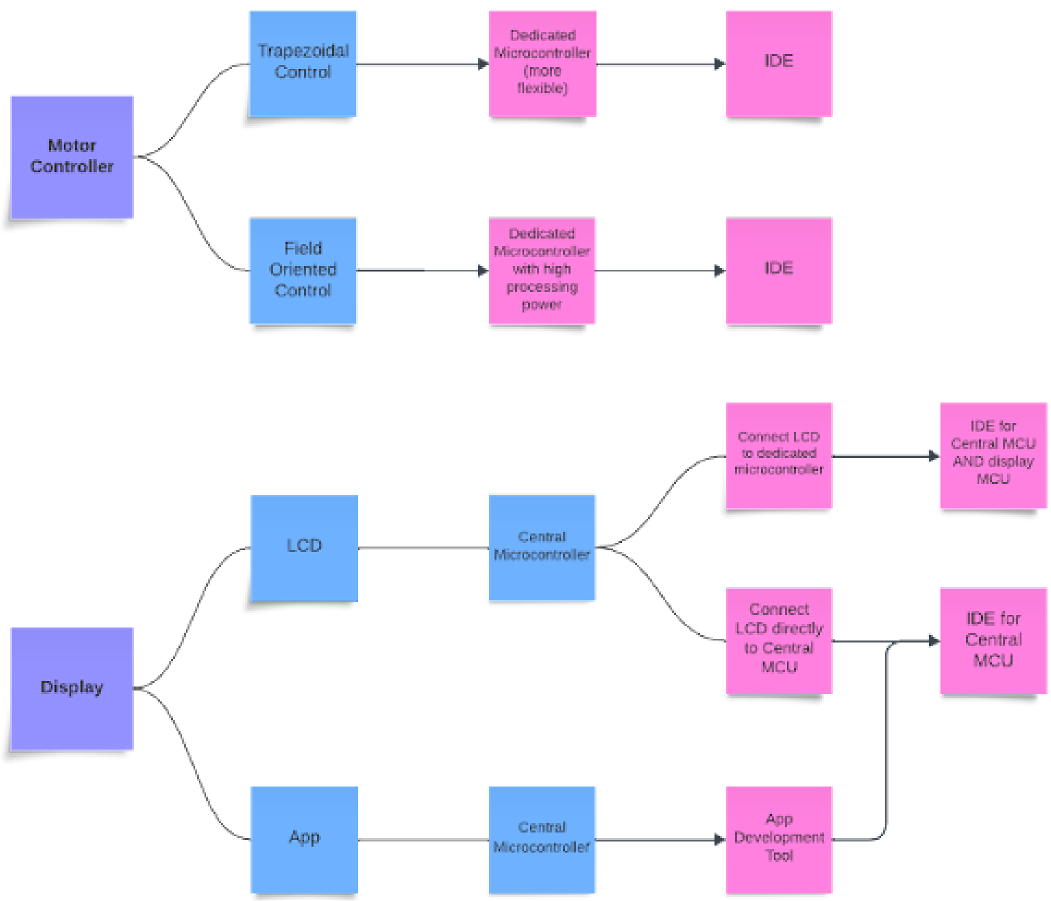


Figure 3.5 - Technology Decision Tree

A research article published in AIP Conference Proceedings compared the implementation of trapezoidal vs. field oriented control on a brushless DC motor (the same type of motor chosen for this project) [28]. Experimental results showed that trapezoidal control had a faster response time than FOC, but FOC seemed more effective overall as it had a smaller torque overshoot, smaller steady-state torque, and smoother speed increase. Also, the maximum power generated by the FOC implementation occurred at a lower rpm than for the trapezoidal control implementation.

Table 3.7 compares the use of trapezoidal vs. field oriented motor control. Comparison factors considered include complexity of the design, resulting efficiency when implemented, power and torque generation, and cost of the microcontroller needed to program the motor controller.

Table 3.8 - Comparison of Motor Control Approaches (Software Perspective)		
	Trapezoidal Control	**Field Oriented Control**
Implementation	"Six-step" block communication with 6 inverter states, algorithmically simpler than FOC	More mathematically complex, using Park and Clark transformations
Efficiency	Less Efficient	Up to 97% more efficient than Trapezoidal Control [29]
Maximum Power Generation	5330W at 1074rpm	5330W at 862rpm
Maximum Torque Generation	67Nm at 300rpm	78Nm at 324rpm
Development Support Available	Six-step firmware library for STM32, TrapeZoid Arduino library	InstaSPIN-FOC for select TI C2000 products, SimpleFOC for Arduino
Cost	None - most microcontrollers already owned by the group could be used	$10-$15 microcontroller, plus $35-$50 development board (optional, but useful for prototyping)

Table 3.8 compares three possible display implementations: an LCD connected to a dedicated MCU with an additional central microcontroller for other sensors in the project, an LCD connected directly to a central MCU, and a smartphone app that reads data from a central MCU via Bluetooth connection. Factors considered include the total number of microcontrollers needed (assuming the motor controller will have its own dedicated MCU), languages used, number of IDEs used, and associated learning opportunities.

Table 3.9 - Comparison of Display Approaches			
	LCD With Dedicated MCU	**LCD Connected to Central MCU**	**Smartphone App**
Number of Microcontrollers	3	2	2
Languages Used	C only	C only	C, along with a development

			language such as JavaScript, Dart, or Kotlin
Number of IDEs	1-3; IDE for each microcontroller may be different	1-2; IDE for motor controller MCU and display MCU may be different	2-3; separate IDEs for programming microcontroller vs. app, possible additional IDE depending on MCUs selected
Learning Opportunities	Embedded programming only; display interfacing	Embedded programming only; display interfacing	Embedded programming and app development

Table 3.9 compares several IDEs that could be used for programming, depending on the microcontroller chosen. These include Code Composer Studio, Arduino, Eclipse, Codeblocks, Aurix, and Vim, each of which the group's programmers have varying levels of familiarity with.

Table 3.10 - Comparison of IDEs

	Code Composer Studio	**Arduino**	**Eclipse**	**Aurix**	**Vim (text editor)**
MCUs Supported	TI Products (MSP430, C2000)	Arduino and ESP32	STM32	Infineon	Any
Familiarity	Most familiar	Somewhat familiar	Somewhat familiar	Never used	Somewhat familiar
Beginner Friendly?	Somewhat	Yes	Somewhat	Somewhat	No

3.4 Part Selection: Software

This section discusses part selection for the microcontrollers that will be used to program the car's motor controller and display. 3.4.1 compares microcontrollers that could work for implementing field oriented control. 3.4.2 compares microcontrollers that could be used as the project's "central" MCU for reading sensor measurements and sending data to the display. 3.4.3 compares app development platforms for creating a display app.

3.4.1 Motor Controller MCU

For the motor controller, processing power and peripherals were key features the group looked for when choosing a microcontroller. Initially, Arduino and MSP430-based motor controllers were considered. These microcontrollers are generally more suitable for Trapezoidal control than Field-Oriented Control.

Once the group had decided on implementing field oriented control (FOC) in the motor controller design, it became doubtful that an Arduino or MSP430 would be sufficient. FOC motor controllers are more efficient than the previously mentioned designs, at the cost of requiring far more processing power.

It is also important to consider that "high performance" products may add unnecessary cost if they supply more advanced features than necessary. Some examples of this are highlighted among the options described below. The goal is to achieve the desired level of performance without purchasing an excessively expensive product.

A list of possible microcontrollers was compiled from multiple sources, including a sales representative, a senior engineer in industry, and ChatGPT.

While attending an informational session hosted by Texas Instruments (TI), one group member inquired about TI products designed for motor control, specifically field oriented control. Sales representative Carlos Ventura recommended TI's C2000 series, many of which have built-in support for FOC algorithms [30]. Some models also include support for bluetooth applications. This series of microcontrollers was investigated further by visiting the TI website. Carlos mentioned TI's "E2E" forum, which offers software development support from the developers of TI products. This points to a potential advantage of choosing a microcontroller from TI, as availability of assistance is valued by the group. When comparing microcontrollers from different brands, the existence of E2E inspired group members to investigate whether the other brands offered similar technical support.

The same group member consulted a former employer, senior Software Engineer Jason Krasnow at ICR Inc, regarding possible microcontroller choices [31]. Jason suggested STM32, noting the availability of evaluation kits to aid the programming process.

Lastly, ChatGPT was asked to generate a list of microcontrollers that could be used for Field Oriented Control. Of the five series it recommended (STM32, Texas Instruments' C2000, NXP Kinetis, Microchip PIC32, and Infineon XMC), three were chosen for further research: STM32, C2000, and Infineon XMC [32]. The other two were neglected due to lack of brand familiarity - the group predicted that these microcontrollers would require a greater learning curve to program compared to products from brands they already knew from previous projects and/or online tutorials.

Based on recommendations from the above sources, the group researched the following microcontrollers:

- STM32
  - STM32F4
    - Balance of cost, processing power, and energy efficiency
    - Hardware acceleration, which may be useful for FOC algorithms by speeding up complex mathematical operations
  - STM32F7
    - Common for motor controllers, have many GPIO pins and range of processing power
    - Built-in hardware for motor control
- C2000
  - Specifically designed for motor control and have support for FOC algorithms
  - TMS320F2806x/TMS320F28069
    - Balance of features and processing power (example project found)
  - TMS320F28027
    - Lower cost
    - Fewer features than more expensive models, but still include peripherals and processing power needed for motor control
  - TMS320F28335
    - High-performance
    - Advanced peripherals, more memory and higher clock speed
    - Good for complex algorithms
- Infineon XMC
  - XMC3100
    - Suitable for FOC
  - XMC4700
    - Focus on performance and connectivity
    - Motor control/industrial applications
  - XMC4800
    - High-performance
    - Industrial automation, advanced motor control applications

Brief notes were included on each family/model based on additional prompts provided to ChatGPT. Microcontrollers from this list were then screened for availability of reference material. While the group did not intend to directly copy previous motor controller implementations, previous works could serve as a helpful guide, and their existence would verify that an FOC-based motor controller could be built using the particular microcontroller included in the documentation.

Documentation on FOC-based motor control was found for the following microcontrollers:

- STM32F4DISCOVERY with STMF407VGT6 processor [33]
- STM32F446RE [34]
- STM32F103C4 [35]
- TMS320F28335 [34]

- TMS320F28069 [34]
- TMS320F2803x [36]

The first example is an induction motor, designed to be used in asynchronous machine control research and testing. Besides implementing FOC, the author also aims to make the hardware platform customizable and easy to program. Setup and testing for the design are explained for readers to recreate the project themselves. Associated project files and Simulink models are publicly available. Some of their C code is generated using MATLAB Embedded Coder to simplify the programming. Previously unaware of MATLAB Embedded Coder, the writer of this report investigated whether or not the tool would help with embedded programming for a go-kart motor controller.

An article from MathWorks explains the capabilities and possible applications of MATLAB Embedded Coder, with the tagline, "Generate C and C++ Code optimized for embedded systems" [37]. The tool is designed for mass production in a variety of industry applications, including image processing, signal processing, computer vision, and machine learning.

The "request a quote" button at the top of the page raised suspicion that the tool would come with a high price tag. Out of curiosity, an educational quote was requested for both the Embedded Coder and another tool called "C2000 Microcontroller Blockset" [38], which sounded useful if the group decided to use a C2000 microcontroller. MathWorks also offers a "Motor Control Blockset" that can be used with the C2000 Microcontroller Blockset [39]. For each of these tools, there are links to documentation and example projects. MathWorks is known to have an active support community, at least for the notorious MATLAB software.

The description for the C2000 Microcontroller Blockset specifically mentions motor control and real-time execution, so it is worth considering whether the tool could be useful for this group's project. As expected, licensing is expensive; the "MATLAB and Simulink Student Suite" (which the C2000 Microcontroller Blockset can be added to) and Motor Control Blockset are priced at $99 and $29 respectively. The University of Central Florida provides students free access to MATLAB through Citrix, so it may be possible that the C2000 Blockset and/or Embedded Coder could be obtained as an add-ons to UCF's license. The group predicted that trying to figure out who to contact to *potentially* gain access to these tools could be a confusing and time-consuming process.

Another time-consuming aspect of using such software tools is the time investment of learning how they work. Software tools designed to "save time" are counterproductive if the user has difficulty understanding how to use them. Plus, it is likely more beneficial to an aspiring engineer to learn *actual* embedded programming rather than learning niche software tools for generating it. For these reasons, the group rejected the use of MathWorks' softwares to help with embedded programming.

The next example is an article that conveniently compares three of the microcontrollers on the above list (the TMS320F28069, TMS320F28335, and STM32F446RE) based on performance in motor control applications. A dedicated senior design group could choose to conduct these sorts of tests manually, but referring to existing testing documentation saves the time and cost of purchasing multiple microcontrollers. The article first provides a general description of each microcontroller (as well as three additional microcontrollers not included on the group's initial research list) and the code used for testing, then goes on to analyze the results in regard to several performance factors.

While TI's C2000 microcontrollers in general are designed for high-performing real-time control, the TMS320F28069 is from the higher-end "Delfino" series, while the TMS320F28335 is a lower-cost option from the "Piccolo" series. The TMS320F28069 is an older model and the first of its series to offer a Floating Point Unit (FPU), and lacks a few of the features included in newer Delfino models, such as the Control Law Accelerator (CLA) and trigonometric accelerator (TMU). CLA, which is included in the TMS320F28335 but not used for comparison in the article, allows for parallelization of floating point operations via a separate core. TMU is a trigonometric accelerator, which has potential to improve the performance of speed estimation. The article gives a less detailed backstory for the STM32F446RE, mainly chosen for its popularity.

In terms of execution time, the TMS320F28335 was one of the highest-performing microcontrollers (shortest execution time), while the TMS320F28069 and STM32F446RE were the lowest (highest execution time) when tested using their maximum clock rates. The author of the article explains that for a more proper comparison of efficiency, the microcontrollers should be normalized to use the highest clock rate common to all of them (90MHz). In this case, the TMS320F28069's performance was closer to that of the TMS320F28335, though the STM32 was still considerably slower. The instruction sets for TI's C2000 microcontrollers are designed for efficient motor control, and here it shows.

Next, the author compares the microcontrollers with respect to memory access speeds. Even with a high clock rate, code execution can be slowed down if the code has to be retrieved from memory rather than being immediately available. Pre-fetching is one strategy to mitigate this, by "fetching" and storing the code in fast cache memory before it is needed at runtime. The author tested the microcontrollers' pre-fetch routines by modifying the previous testing procedure to copy the code into RAM and execute it from there. Now, little cache access is necessary aside from standard library code. The C2000 microcontrollers showed a significant reduction in execution time from doing this, while the execution time for the STM32 did not change at all. This is significant because it suggests the C2000 performs best only when its code is run from RAM. RAM storage is much more limited than flash storage, so the programmer may have to decide which parts of the code are most vital to real-time execution if not all of it will fit in RAM. Lastly, the author demonstrates the effectiveness of TI's "IQmath" library, which optimizes fixed point operations for the C2000 microcontrollers.

The author concludes that of the microcontrollers tested, the C2000 line "seems to be the most effective at real time control loops". Additionally, the C2000s feature impressive peripherals for Pulse Width Modulation and Analog-Digital Conversion. These results give motivation for choosing a C2000 for the motor controller in this project.

Not included on the original list of microcontrollers to research is the STM32F103C4, for which an example motor controller project was found by engineer Shane Colton. Shane has designed several motor controllers for various projects; his work is one of the inspirations for the motor controller design presented in this report. In an article summarizing his designs, most use the STM32F103C4 for sensorless field-oriented control. He notes the impressive processing power provided by the STM32 compared to other microcontrollers he had used in other projects or previous versions of the designs.

Examples using Infineon products were harder to find. Infineon offers documentation on FOC capabilities for their XMC1300/1400 and 4400 [40], but initial research showed a general lack of documentation on projects using these microcontrollers. They may be more geared toward industry rather than hobbyist/academic applications, and implementation of FOC using these products may be better left to more experienced engineers.

Examples of field-oriented control using microcontrollers from well-known “beginner-friendly” series, the MSP430 and Arduino, were also found. Two sensored FOC motor controllers using an MSP430F2274 was designed by Shane Colton [35]. Shane refers to the first design as his “first and probably most frustrating motor controller project”, though not necessarily because of the microcontroller. The second design is an early version of his “3ph line” of motor controllers. The project includes extensive documentation, but the version using the MSP430 did not last very long before switching to later versions featuring the STM32F103C4, which is more powerful. The example using the MSP430 is also not useful because it uses *sensored* field-oriented control. As previously explained, sensored field-oriented control requires hall effect sensors, and the motor for this project’s car does not have those. Furthermore, although the group’s programmers have experience using MSP430 microcontrollers in general, they have not used and do not own the particular model used in this example. Thus, prior experience and the potential to save on component costs does not apply here.

The Arduino example uses Arduino SimpleFOC [41], a library for the Arduino IDE that works with several microcontroller families, including STM32, ESP32, and many Arduino models (Uno, Mega, Nano, etc.). As the name suggests, the library is designed for “simple”, “plug and play”-type implementation of field-oriented control. The library is open-source, and there is documentation available for multiple example projects. Unfortunately, SimpleFOC also only runs in sensored mode, so it will not be useful for this project.

Table 3.11 - General Comparison of Product Lines from Multiple Brands

Brand/Product	**Benefits**	**Drawbacks**	**Overall Thoughts**

Line			
C2000	-Designed for motor control applications, including FOC support and impressive peripherals -High processing power -Brand & IDE familiarity -Availability of documentation -Active support community, including “E2E” forum for guidance from experienced engineers -Many options to choose from -Proven performance in motor control applications -Optimized for real-time applications	-Group does not own one of these microcontrollers already and would need to purchase one -Group members do not have previous experience with this product -May be less “beginner-friendly” than Arduino/MSP430	After comparing several product lines from multiple brands, this was the group’s top choice for perceived performance and availability of user support.
STM32	-Successful example projects with FOC found using STM32F103C4 -Additional examples of motor control found for other STM32 products, demonstrating they are decently suited for this purpose -High processing	-While they may “work” for field-oriented motor control applications, they are not necessarily specifically designed for it -Lower demonstrated performance in motor control applications compared to C2000	This product line would most likely meet the needs of the group’s intended application, and would be a convenient choice considering there would be no delivery delay or costs incurred from buying a new microcontroller.

	power -Group already owns a few STM32s and would not need to purchase one	-Uses Eclipse IDE, which group members are unfamiliar with	However, a C2000 may perform slightly better and be easier to use.
XMC	-High performance and processing power -Documentation available for FOC applications	-Less widely-used than other brands -Lack of example projects -Uses Aurix IDE, which group members are unfamiliar with	This brand was ruled out early on as the group was skeptical about the availability of user support and lack of brand familiarity.
MSP430	-Brand, product, and IDE familiarity -Group already owns an MSP430g2553 -Beginner friendly -Active support community	-Limited processing power -Main examples using MSP430s show a *lack of success* for FOC implementation and require hall effect sensors	MSP430s were determined not to be suitable to the group's goals for field-oriented motor control due to inadequate processing power and the need for sensors the car's motor does not have
Arduino	-Brand, product, and IDE familiarity -Beginner friendly -Active support community -Available libraries specific to FOC -Group owns several models from this brand	-The relevant library, SimpleFOC, only works for sensored applications	Arduino products were ruled out for reasons similar to the MSP430

Narrowing down possible microcontroller choices, the group looked toward Texas Instruments' C2000 line as the most promising direction, as these microcontrollers are widely used for field-oriented motor control and include features and optimizations specific to FOC implementation. Group members felt more comfortable choosing a microcontroller from Texas Instruments as opposed to other brands because they had used other TI products in their classes and were therefore already familiar with the associated IDE (Code Composer Studio) designed for those microcontrollers. Although group members had only used MSP430s rather than C2000s, they predicted the learning curve would be less steep compared to learning how to use a microcontroller and IDE from an entirely new brand. Since it had been decided that an MSP430 would not be best suited for field-oriented motor control, this seemed to be a necessary investment the group would have to accept.

Texas Instruments provides documentation on how to implement Field-Oriented Control on products that support them, such as the TMS320F2803x family, which was one of the main contenders for the group's final choice of microcontroller for the motor controller [36]. In a video released by TI titled "Field Oriented Control of Permanent Magnet Motors", [42] the narrator explains Field Oriented Control in detail and gives an overview of TI's C2000 line and associated libraries.

As mentioned in the earlier example comparing the performance of various microcontrollers, the C2000 line is separated into the high-end "Delfino" series and the lower-cost "Piccolo" series. The Piccolo series includes the TMS320F2802x and TMS320F2803x families, which the narrator of the video describes as being sufficient for most motor control applications. The Delfino products are more expensive and offer higher processing power, though the higher processing power would likely be excessive for this group's purposes. Assuming this project falls under "most motor control applications", the group leans toward choosing a less expensive but still capable microcontroller from the Piccolo series.

Based on the documentation found through the research process, three models from TI's C2000 product line were chosen for comparison: the TMS320F28335 from the performance comparison article, the TMS320F28034PNTR from a group member's prior research for another project, and the TMS320F280049, which was suggested by Mouser Electronics while browsing products in the Delfino category. The microcontrollers were compared based on the family documentation provided by Texas Instruments. All three models are from the Piccolo series.

Figure 3.12 - Comparison of C2000 Models

Model	**TMS320F28335 [43]**	**TMS320F28034PNTR [44]**	**TMS320F280049C [45]**
Clock Frequency	150MHz	60MHz	100MHz
Flash Memory	512KB	128KB	256KB
RAM	68KB	20KB	100KB
Processing (MIPS)	150	60	200

Notable Features	FPU (floating point unit)	CLA (control law accelerator)	CLA, FPU, TMU (trigonometric accelerator), InstaSPIN-FOC
Unit Price	$29.53 [46]	$8.86 [47]	$10.17 [48]

From price and specifications alone, the TMS320F280049C seems like an obvious choice; it is significantly cheaper than the TMS320F28335, and similarly priced to the TMS320F2803 while offering twice as much Flash memory, five times as much RAM, and much higher processing speed.

One group member pointed out that the TMS320F2803x had a dedicated application note from TI on how to implement sensorless field-oriented control [49], which could be a valuable resource. Reading through this example project, the 5630 words and 1400 words are required for program and data memory, respectively, on the 2803x. For 32-bit words with 4 bytes per word, more than 20KB is needed to support the program memory alone. Although the 128KB Flash memory is sufficient to meet this requirement, recall from an earlier example in which C2000 microcontrollers showed a significant improvement in performance by executing code from RAM. It may be beneficial to purchase a microcontroller with more RAM memory for this purpose.

C Framework				
System Name	Program Memory Usage 2806x	Data Memory Usage[1] 2806x	Program Memory Usage 2803x	Data Memory Usage[1] 2803x
PM_Sensorless	5000 words[3]	1384 words	5360 words[2]	1400 words

Table 3.13 - Sensorless FOC Memory Usage from TI Application Note

With these ideas in mind, related documentation was parsed for potential usefulness. In a document comparing the TMS320F28004x family with the 2803x and 2806x families, the 28004x seems to be an “improved” version of the other two families with newer features. Two comparisons mentioned are the addition of the 28004x’s TMU for accelerating trigonometric operations, and the use of a “Type 2” CLA rather than the “Type 0” CLA used by the 2803x and 2806x. Type 0 CLAs can access two or three RAM blocks and have a 4K memory address reach, while Type 2 CLAs can access eight RAM blocks and have a 64K memory reach. This means increased flexibility for the 28004x’s CLA.

The group found an application note titled “Sensorless-FOC for PMSM With Single DC-Link Shunt” featuring the TMSF280049C [50], as well as for a feature called “InstaSPIN-FOC”, which is included with that model.

The InstaSPIN-FOC documentation [51] lists “Traction applications” such as e-bikes, scooters, or in this case, a go-kart, as example motor control applications. TI advertises that this feature is designed to make motor control development on select C2000 devices easier through special built-in libraries, and that InstaSPIN-FOC “enables designers—even those with limited motor control experience—to identify, tune and fully control any type of three-phase, variable speed, sensorless, synchronous or asynchronous motor control system in just minutes.” While “just minutes” may be too lofty of an expectation for a team of relatively inexperienced programmers, the flexibility and ease-of-use presented by the article led the group to feel confident about the 280049C as a viable choice for programming the motor controller.

3.4.2 Display MCU

The group initially had a couple ideas on which microcontroller would be the best option for the driver’s display system - either an Arduino or MSP430 series microcontroller. After more research and consideration, the MSP430 series microcontroller seemed like a better choice because the programmers had more experience with the microcontroller brand (Texas Instruments), which would make it easier to work with.

The original plan was to connect the microcontroller to an LCD display. Before starting research on potential microcontrollers, the group described the display system design to ChatGPT and asked it to give a list of microcontrollers that could be used. It suggested the following options suited for working with an LCD display, based on its latest knowledge from 2021:

Arduino Nano: The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328 (Arduino Nano 3.x). It has more or less the same functionality of the Arduino Duemilanove, but in a different package. It lacks only a DC power jack, and works with a Mini-B USB cable instead of a standard one.

Raspberry Pi: Raspberry Pi 5 is built using the RP1 I/O controller, a package containing silicon designed in-house at Raspberry Pi. USB 3 has more total bandwidth, for much faster transfer speeds. Camera and DSI display connectors are interchangeable, so you can have one of each, or two the same.

ESP8266 and ESP32: A feature-rich MCU with integrated Wi-Fi and Bluetooth connectivity for a wide-range of applications.

STM32 series: The STM32 family of 32-bit microcontrollers based on the Arm Cortex-M processor is designed to offer new degrees of freedom to MCU users. It offers products combining very high performance, real-time capabilities, digital signal processing, low-power / low-voltage operation, and connectivity, while maintaining full integration and ease of development.

PIC Microcontrollers: Various PIC microcontrollers like PIC16, PIC18, PIC32 can be used for LCD interfacing. The PIC family offers truly differentiated general-purpose microcontrollers that are ideal for many space-constrained, low-power, cost-sensitive applications.

Atmel/Microchip AVR: AT mega series, including ATmega328P and ATmega2560. Microchip Studio is an Integrated Development Environment (IDE) for developing and debugging AVR and SAM microcontroller applications. It merges all of the great features and functionality of Atmel Studio into Microchip's well-supported portfolio of development tools to give you a seamless and easy-to-use environment for writing, building and debugging your applications written in C/C++ or assembly code

Teensy: Teensy 3.x and 4.x boards are known for their powerful features and support for LCDs (Liquid Crystal Display). Teensy is a complete USB-based microcontroller development system, in a very small footprint, capable of implementing multiple project tasks. All programming is done via the USB port.

BeagleBone Black: Suitable for embedded systems and more complex applications. BeagleBone Black is a low-cost, high-expansion, community-supported development platform for developers and hobbyists.

NXP/Freescale i.MX Series: i.MX6 and i.MX7 are more suitable for applications and programs that require more processing power.

Cypress PSoC: PSoC microcontrollers offer flexibility for LCD applications.

Texas Instruments MSP430: Suitable for low-power LCD applications. The MSP 430 provides affordable solutions for all applications.

ARM Cortex-M Microcontrollers: Various microcontrollers based on ARM Cortex-M cores from different manufacturers. The Arm Cortex-M4 processor is a highly-efficient embedded processor that was designed to fix digital signal control markets that require an efficient microcontroller.

Intel Edison/Galileo: These are suitable for more advanced projects.

Odroid C1/C2/XU4: Suitable for multimedia and display applications.

Particle Photon: Suitable for IoT applications with LCD displays.

Espressif ESP8266/ESP32: Great for low-cost projects with Wi-Fi capabilities.

Adafruit Feather M0/M4: Designed for lightweight and portable projects with LCD displays.

Nordic Semiconductor nRF52 Series: Ideal for Bluetooth-enabled applications with LCDs.

Noting that ChatGPT may not have the most up-to-date information on available microcontrollers, the group briefly searched for additional options released after 2021. No significantly better options were found.

Table 3.14 compares the specifications of microcontrollers suggested by ChatGPT.

Table 3.14 - Microcontroller Specification Comparison

	Power Efficiency	Processing Power	Memory	Hardware Interface	Software Architecture	Cost
[52]Arduino Uno	5V Operating Voltage, 7-12 recommended input voltage	16 MHz Clock Speed	32 KB Flash memory and 2 KB SRAM	USB connection, Power jack, reset button	C and C++	$27.60
[53]Arduino Mega	5V Operating Voltage, 7-12 recommended input voltage	16 MHz Clock Speed	256 KB Flash memory and 8 KB SRAM	USB connection, Power jack, reset button	C and C++	$48.40
[54]Arduino Due	3.3V Operating Voltage, 7-12 recommended input voltage	84 MHz Clock Speed	512 KB Flash memory and 96 KB SRAM	USB connection, Power jack, reset button, Erase button	C and C++	$48.40
[55]Raspberry Pi	5V Operating Voltage	1.8 GHz Clock speed	1,2,4 or 8GB SDRAM, micro-SD card slot for storage options	USB connection, Bluetooth connection, Gigabit ethernet, Micro-HDMI ports, camera port, display port	Python, Scratch, Java, C and C++, JavaScript	$35 – $85 (Depending on RAM requirements)
[56]ESP 32	3.3V Operating Voltage	240 MHz Clock Speed	448 KB ROM and 520 KB SRAM	Low power options, Bluetooth connection, Wi-Fi, touch sensors,	C and C++, python	$8.00

				USB connection, reset button		
[57]STM 32	1.8-3.6 Operating Voltage	120 MHz Clock Speed	1 MB Flash memory, 128 KB SRAM	Low power options	C and C++, assembly language	$13.52
[58]PIC	2-5.5 Operating Voltage	20 MHz Clock Speed	14 KB Flash Memory, 368 KB SRAM	Low power options	C, assembly language	$6.71 - $8.55
[59]Atmel /Microchip AVR	1.8-5.5 Operating Voltage	16 MHz Clock Speed	256 KB Flash Memory, 8 KB SRAM	-	C	$20.18
[60]Teensy	3.3 Operating Voltage	72 MHz Clock Speed	256 KB Flash Memory, 1024 KB SRAM	USB connection, Micro SD available	C and C++	$19.95
[61]Beagle Bone	5V Operating Voltage	1 GHz Clock Speed	4 GB Flash Memory, 512 MB SDRAM	USB connection, Ethernet Connection, Power button, Boot button, Power Button	C and C++, Java, Python	$76.25
[62]NXP	1.4 Operating Voltage	800 MHz Clock Speed	256 KB Flash Memory, 128 KB SRAM	USB connection	C	$8.80
[63]MSP 430	1.8 – 3.6 Operating Voltage	25 MHz Clock Speed	512 KB Flash Memory, 32 KB	Ultra-low power mode	C and C++, Assembl	$7.926

			SRAM		y language	
Intel Edison ([64]Discontinued)	3.3-4.5 Operating Voltage	19.2 MHz Clock Speed	512 KB Flash Memory, 32 KB SRAM	USB connection, Wi-Fi connection, Bluetooth connection	C and C++, Python	$49.55
[65]Intel Galileo (Discontinued)	7-15 Operating Voltage	400 MHz Clock Speed	256 MB Flash Memory, 512 KB SRAM	USB connection, Ethernet connection, Reboot Button, SD card expansion	C and C++, Python	$39.99
[66]Odroid C1		1.5 GHz Clock Speed	Micro-SD expandable (8-16GB) Flash Memory, 512 KB SRAM	Ethernet connection, USB connection	C	$35.00
[67]Particle Photon	4.5-5 Operating Voltage	200 MHz Clock Speed	2048 KB Flash Memory, 3072 KB SRAM	USB connection, Wi-Fi connection, Bluetooth connection	C	$17.95
[68]Adafruit Feather M0	3.3 Operating Voltage	48 MHz Clock Speed	256 KB Flash Memory, 32 KB SRAM	USB connection, Reset button	Circuit Python	$19.95

Based on this research, a list of pros and cons for each microcontroller was produced:

Table 3.15 - Microcontroller Pros and Cons Comparison

Microcontroller	Pros	Cons
Arduino Uno	Easy to use, large language libraries available, Active user community	Higher cost value, Low memory capacity, Limited programming language availability, lack of connection
Arduino Mega	Easy to use, large language libraries available, Active user community	Higher cost value, Limited programming language availability, lack of connection
Arduino Uno	Easy to use, large language libraries available, Active user community	Higher cost value, Limited programming language availability, lack of connection
Raspberry Pi	Extremely fast processing time, exceptionally large memory capacity, great connections (Bluetooth, Ethernet, HDMI, and display port), wide programming language selection pool	Higher cost value
ESP 32	Extremely fast processing time, large memory capacity, great connections (Bluetooth, Ethernet, HDMI, and display port), great pricing for performance range	
STM 32	Low power options	
PIC	Low power options, very cost efficient	Low storage capacity
Atmel	Low power options	Cost inefficient, Low storage capacity
Teensy	Good peripherals connect-ability	Cost inefficient
Beagle Bone	Large memory availability, Great peripheral connect-ability, Hardware for software aid	Bad for the project budget, over resourced for the workload
NXP	Cost efficient	Lack of programming language options
MSP 430	Low power options, Cost efficient	

Intel Edison	Wireless connection availability, Peripheral connect-ability	Cost inefficient, Discontinued product with limited availability
Intel Galileo	Peripherals connect-ability, Memory expansion options	High operating voltage, Cost inefficient, Discontinued product with limited availability
Odroid C1	Peripherals connect-ability, Memory expansion options	Cost inefficient
Particle Photon	Wireless connection availability, peripheral connect-ability, cost efficient option	
Adafruit Feather M0	Peripheral connection	Cost inefficient, no choice of commonly used and library available programming language options

Our original idea was to use a microcontroller to calculate and display certain information to an LCD screen because LCD screens are more power efficient than LED screens. Later, one group member suggested looking into using an ESP32 microcontroller. Rather than programming a microcontroller to take in data and display the information to the driver on a LCD screen placed in the car, the data would instead be displayed to a mobile app to be developed by the group. This option seemed easier to implement, and more efficient for the project. Additionally, this method would consume less power due to the fact that a smart phone would be placed in the vehicle and would not cost the battery of the car any power.

After deciding on the ESP 32 microcontroller to get the data and display all the information the driver will need throughout the duration of the race to a smartphone, a programming language had to be chosen for both the microcontroller and the mobile app associated with the display. The group's programmers had experience with C and assembly language so those were the two main options to choose from, but other programming languages compatible with the ESP32 were also considered:

- JavaScript
- Rust
- Micropython
- Arduino IDE (C/C++)
- Espressif IDF (C/C++)

A pros and cons table was produced for each of the programming languages:

Table 3.16 - Comparison of Microcontroller Languages/IDEs		
Programming Language	Pros	Cons
[69]JavaScript	Large active community for resource purposes	Lack of low-level controls, little library support
[70]Rust	Good for safety-critical work	Less commonly used leading to less resources available, steep learning curve and the group is not familiar with this language
[71]Micro python	Easy to use and comprehend, active programming community leading to available resources	Consumes more processor resources than C and C++, less useful library functions compared to c and C++
[72]Arduino IDE	User friendly platform, large community leading to available resources, large library for more efficient work, bridges hardware and software efficiently for easier comprehension	Lack of low-level controls, less efficient for larger tasks
[73]Espressif IDF	Access to low level controls which allows for more optimized performances, better handles larger tasks, active community for more resources	Steep learning curve but group familiarity with C and C++ helps
Assembly	Sometimes necessary for embedded applications in industry	Unnecessarily complicated

The C language, via the Arduino IDE was the obvious choice. Although assembly was technically an option, almost no one actually writes code in assembly anymore, and doing so would be an unnecessary complication for the project. The Arduino IDE is generally recommended for the ESP32 and has included libraries to expedite the programming process.

Next, a language was selected for programming the mobile app. The group needed a programming language that was particularly good for application development. A list of such programming languages includes the following:

- JavaScript
- Java
- Swift
- Kotlin
- Dart
- Python
- C++

A pros and cons table for each of these programming languages was produced:

Table 3.17 - Comparison of Programming Languages		
Programming Language	Pros	Cons
Java	Faster and easier to develop with, can access more efficient data structures, more compatible with multiple browsers (Cross platform language), Exceptionally large active community	Less ideal for front-end development
JavaScript	JavaScript is more commonly used among developers leading to more resources available, more of a lightweight and less complex language	Lack of access to more efficient data structures, less compatible with multiple platforms, relies on an interpreter
Dart	More flexible and comprehensible syntax, Primary use fits our exact project needs	Small development community leading to less resources than some other languages
Python	Lower application sizes, easily readable code	Android does not use python, so the code will have to be converted to Java; macOS device is required for app development with python
C++	Fast compiling time, C++ is a very portable programming language	C++ is liable to memory leaks, C++ sometimes compiles different on different platforms
Kotlin	Good active community for resources, Platform flexibility	Slower compilation rate, Kotlin is a Java

	for ease of code transfer, Compiles with Java code	dependent programming language, more difficult syntax
Swift	Extremely fast compile time, Error handling built-in, cross platform support	Designed for iOS devices and not Android devices

Dart, which is associated with the app development platform “Flutter”, was chosen as the best language for the mobile app. This decision, based on the programmers’ prior experience with supposedly similar languages (Java and C), is further detailed in the next section.

3.4.3 App Development Platform

As with the microcontroller decision process, the choice of an app development tool largely revolves around what the group’s programmers feel most confident about using, as neither have extensive experience with app development, especially that involving bluetooth connection to a microcontroller.

Because there are a plethora of free, publicly available app development tools available, it makes sense to only consider free softwares, considering the relatively simple needs of the project from the display perspective. The next major factor is the availability of tutorials specifically related to the group’s chosen microcontroller, the ESP32. An article titled, “How to Develop a Mobile App That Communicates With Your Product Using Bluetooth” presents a few popular options: Android Studio, Swift, Flutter, and Blynk [74]. Android Studio, as the name implies, works with Android devices, while Swift is for iOS devices. Since users of the app to be developed could have a smart phone of either type, these two options were ruled out to avoid compatibility issues. Blynk was also ruled out per the article’s mention that it “should only be used for prototyping or constructing a Proof of Concept”. The group did not care to spend extra time and energy on transferring a prototyped design to another software later on.

This leaves Flutter as the only contender so far. Created by Google, Flutter is free, open-source, and works with both Android and iOS devices. The author of the article advertises that Flutter is easy to use and good for fast prototyping. It uses a language called “Dart”, which is supposedly similar to Java but more niche, less verbose, and mainly used for mobile and web app development [75]. Based on example Dart codes, the group’s programmers imagined this language would be fairly straightforward to learn and use, though perhaps not particularly useful beyond the scope of this project, as its developer community is fairly small compared to other languages.

Taking note of this option, the group searched for other app development tools that might fit the project’s needs. A quick Google search leads to many examples of how to read data from an ESP32 to a mobile app, using a variety of app development tools, including the following:

- Thunkable [76]
- React Native [77]
- Flutter [78] [79]

Several factors were considered when comparing these tools: programming language (if applicable), perceived ease of use, availability of tutorials and/or support communities, and how well-suited each was specifically for receiving data from a microcontroller.

First, Thunkable is considered. Thunkable is a codeless platform, allowing users to drag and drop components onto their app without writing any code. Thunkable supports Bluetooth connectivity, a minimum qualification for the project. On Thunkable.com's landing page, it advertises the ability to publish apps directly to the web, Apple App Store, or Google Play. However, looking over pricing plans, this feature is only available through the "Pro", "Business", or "Team" plans, costing $45/month, $200/month, and $500/month respectively.

Luckily, the free plan appears promising for creating a simple display app. Free users can create up to 10 projects using up to 100MB of storage (total). Live testing is available, along with up to two downloads per month. The main limitations here are the storage space and allocated downloads. To avoid timeline delays in the project, the group would need to be cautious about thoroughly pre-testing the app to make sure no additional changes need to be made before deployment. To determine if 100MB would be enough, the group browsed the Google Play store, noting the storage requirements of apps with similar capabilities. A "BLE Scanner" (Bluetooth connectivity, [80]) and basic timer app [81] require 4-6MB, while trip-tracking apps such as State Farm's "Drive Safe & Save" [82] and Driversnote's "Mileage Tracker" [83] need around 36MB. Seeing that these apps are all under 50MB gave confidence that 100MB would be sufficient for monitoring and displaying race statistics.

A basic tutorial for Bluetooth connectivity with Thunkable from instructables.com [76] uses an Android phone, ESP32 Development Board, and micro USB cable. Sensors and LEDs may be added later to "spice up the project", but these are not necessary to complete the tutorial. The author of the tutorial links a source of inspiration - a Youtube series providing further guidance on Bluetooth setup for ESP32s [84].

Asking peers about app development tools they had used led to the group's consideration of React Native as a potential option. React Native, created by Facebook, is a mobile app development tool that uses JavaScript, along with the React library for building user interfaces. It can be used to build apps for both iOS and Android devices. React is free and open-source, and has an active community of developers, as well as a variety of learning resources.

Using React would necessitate learning how to use JavaScript, which the group's programmers have no experience with. However, this would be an easily-justifiable time investment, as JavaScript is widely used in industry for frontend development.

A few tutorials featuring an ESP32 were found for React Native. The first, titled, "Building a live data feed BLE application with React Native and ESP32", was written as an intentionally-generic guide for Bluetooth setup. The tutorial uses a HX711 load cell amplifier to demonstrate the reading of sensor data, but notes that other sensors can be used with small code modifications. Other tutorials include Youtube videos for a step tracker using a breadboard [85] and basic Bluetooth app [86] available on GitHub, which includes links to additional resources and tutorials. Although learning an unfamiliar programming language under limited time constraints is intimidating, the availability of tutorials for React Native gave the group confidence that it could be done.

Other tutorials for the ESP32 were found using DroidScript [87] and MIT App Inventor [88], but unfortunately these platforms are again only designed for Android devices. Additional suggestions from ChatGPT include Xamarin, PhoneGap, Ionic, Appgyver, Sencha, Corona SDK, and Unity. A brief search for relevant tutorials yielded unsatisfying results, so the group returned their research focus to three main options: Flutter, Thunkable, and React Native. The following table compares these platforms:

Table 3.18 - Comparison of App Development Platforms

	Flutter	**Thunkable**	**React Native**
Language	Dart	Codeless	JavaScript
Support Community	Support groups on Discord, Slack, Stack Overflow, Reddit, and Google Developer	Thunkable Community Forum with about 20 posts/day	Multiple Discord groups, Slack; tagged content on Medium, Hacker News, and Reddit
Industry Relevance	Some skills may be applicable to industry	Little to none	Highly applicable to industry (JavaScript/frontend development)
Setup Requirements	Need separate IDE and Android Studio; multi-step installation process	Online interface and live testing app downloaded to phone	Need separate IDE
Limitations to free access	N/A (Open-source)	100MB storage and 2 download/month	N/A (Open-source)

		limit	

From the initial comparison, all three seem like “good” options, with no clear “winner”. The group decided a more robust comparison may be completed by attempting to build a basic app using each to get a better idea of how easy they would be to work with. Because Flutter, Thunkable, and React Native are free to use and multiple tutorials were found using only the ESP32 and optional sensors, this is easy to accomplish (in theory) at no cost besides a small time investment.

After briefly experimenting with each of the three options and considering the programmers’ technical background, Flutter was chosen as the preferred app development platform.

4 Design Constraints and Standards

This project is subject to many design constraints based on the Electrathon rulebook and the race courses on which the car will be driven. Given our limited resources, cost, manufacturability, and time are also very binding constraints.

Engineering standards are established norms developed through consensus by engineers working in private industry. They represent a collection of best-practices to ensure the safe operation of devices, and in the case of electrical standards, to protect consumers from electric shocks. The Occupational Health and Safety Administration (OSHA) considers voltage below 50V to be inherently safe because the resistance of dry skin is high enough to prevent serious injuries. The International Electrotechnical Commission (IEC) standards organization, the Institute of Electrical and Electronics Engineers (IEEE) professional organization, and the safety certification company Underwriter's Laboratories (UL) set a similar threshold at 25V AC or 60V DC.

Due to limits on the availability of motors and battery cells, the Electrathon car we're building in this project needs to use a 15S battery pack, one with 15 levels of cells in series. The maximum charging voltage of lithium ion cells is 4.2V, so the battery pack voltage will briefly reach 15*4.2= 63V across the positive and negative terminals. Cell voltage quickly falls below 4V, bringing the pack voltage below 60V, but the main takeaway is that our battery pack is on the limit of being a "hazardous voltage" and should be treated with care. To maintain safety when working with these voltages, we will follow all applicable engineering standards as closely as possible. The IEEE 1573 and IEC 62368-1 standards will be discussed in detail.

4.1 Related Standards

This section discusses several engineering standards related to the project, including the Joint Test Action Group (JTAG) Communication Protocol, ARM Cortex Debugger Connector, Inter-Integrated Circuit (I^2C) Communication Protocol, Bullet Connectors, Component Sizes, Electromagnetic Compatibility, IEEE 1573 (electric power subsystem standards), and IEC 62368-1 (safety standard for audio, video, information, and communication technology equipment).

4.1.1 Joint Test Action Group (JTAG) Communication Protocol

JTAG stands for Joint Test Action Group, which is a standard for testing integrated circuits, especially those used in digital electronics. It is a common interface that provides access to the individual pins or internal signals of a microcontroller, or other electronic devices. Originally, JTAG was developed as a standardized test access port that would allow for testing printed circuit boards using boundary scan. It has since evolved to become a useful tool for tasks beyond testing, such as debugging, programming, and in-circuit emulation [89].

JTAG allows for the examination and modification of internal data and control signals, which can be valuable for tasks such as debugging software or firmware, or for accessing the memory and registers of a microcontroller during its operation. This capability is particularly useful in situations where other debugging methods, such as using an in-circuit debugger, are impractical. JTAG is commonly used in the development and testing of electronic devices, especially in the fields of embedded systems, microcontrollers, and digital signal processors. It has become an essential tool for hardware and software engineers working on complex systems, allowing them to diagnose and fix issues at the hardware level. Common implementations use 14 or 20 pin connectors, and while they provide greater access for debugging, they use more pins than the 2-pin Serial Wire Debug interface, a common alternative.

4.1.2 ARM Cortex Debug Connector

The Cortex Debug 10-pin connector is a standardized debug connector used for connecting and debugging ARM-based microcontrollers and microprocessors. This connector is commonly used for programming and debugging ARM Cortex-M based devices, which are widely used in embedded systems and IoT (Internet of Things) applications. The connector typically consists of ten pins, enabling communication between the target device and the debugger. The pins serve various functions, including debugging, programming, and providing power to the device. The specific pinout and functionality can vary slightly depending on the specific implementation and the target device's architecture [90].

This connector is often used in conjunction with industry-standard debuggers and programmers that support the Cortex Debug Interface. The Cortex Debug Interface provides a standardized way to communicate with and debug ARM Cortex-M based devices, making it easier for developers to work with these microcontrollers and microprocessors during the development and debugging process. Interestingly, Texas Instruments (TI) also uses this connector as the port that links the XDS110 programmer to the microcontroller on their F280049C development board. Electrically, it implements a 10 pin JTAG interface, and mechanically the connector consists of two rows of 5 pins with 50 mil or 1.27mm spacing. When moving from the development board to a custom Printed Circuit Board (PCB), we will program the microcontroller by linking the programmer on the development board to a 10-pin Cortex Debug Connector on our custom PCB.

4.1.3 Inter-Integrated Circuit (I^2C) Communication Protocol

The Inter-Integrated Circuit (I^2C) protocol is a widely used serial communication protocol that enables communication between multiple integrated circuits in electronic devices. It was developed by Philips Semiconductor (now NXP Semiconductors) in the early 1980s. The I2C protocol is commonly used for communication between various peripherals and integrated circuits on the same circuit board. Key features of the I2C protocol include [91]:

1. Master-slave architecture: In I^2C communication, there is a master device that initiates and controls the communication, and one or more slave devices that respond to the master's requests.
2. Bi-directional data transfer: The I^2C protocol supports bidirectional data transfer between the master and slave devices, allowing for both the transmission and reception of data.
3. Serial communication: Data is transmitted sequentially over a two-wire serial bus, consisting of a data line (SDA) and a clock line (SCL). This two-wire interface simplifies the hardware requirements for communication between devices.
4. Addressing: Each device connected to the I^2C bus is identified by a unique address, allowing the master to select and communicate with specific slave devices.
5. Multi-master capability: The I^2C protocol supports multiple master devices on the same bus, allowing for complex communication scenarios where multiple masters may need to control various slave devices.

I2C is commonly used for various applications in embedded systems, such as connecting sensors, EEPROMs (Electrically Erasable Programmable Read-Only Memory), real-time clocks, and other peripheral devices. We will use I^2C to pass data between the main microcontroller that broadcasts via Bluetooth and the microcontroller in the motor controller, and it will be used to interface other sensors and modules as necessary.

4.1.4 Bullet Connectors

Bullet connectors are a type of electrical connector that is widely used in various applications, particularly in automotive, marine, and other electrical systems. They are named for their cylindrical shape, which resembles a bullet. These connectors consist of a male and female component that can be easily mated and unmated for quick and reliable electrical connections. Their key features include easy connection, reliable contact, compatibility, insulation options, and availability in a variety of sizes with different current ratings.

Bullet connectors are commonly used in automotive applications for connecting electrical components such as lights, switches, and motors. They are also found in marine and other electrical systems where quick and reliable connections are needed. Their simple design and reliable performance make them a popular choice for various applications where electrical connections need to be established and disconnected frequently. We will use bullet connectors for all high power connections in the car, such as the wires between the motor and the motor controller, the motor controller and wiring harness, and the wiring harness and battery. To minimize electrical resistance, we will use 4mm bullet connectors on the charging leads and larger 6.5mm connectors everywhere else.

4.1.5 Standard Electronic Component Sizes

Instead of one standard, this is a lot of different standards, but each one does not require its own section. When making a PCB, it's vital that the size, shape, and configuration of the pads and holes match the physical dimensions of the component that you're trying to mount onto the board. To simplify this process, manufacturers make their parts according to standard component sizes, but in this case the classic joke applies that "the good thing about standards is that there are so many to choose from" [92]. A package is the injection-molded plastic case that surrounds the silicon die, which is connected to the metal pads by the leadframe. The table below shows simply the most common package types for integrated circuits, and within each type there are variations in pin number, pin pitch, and the distance between rows of pins [93].

Table 4.1 - Common Integrated Circuit Packages		
Dual In-Line Package (DIP)	Thin Small Outline Package (TSOP)	Small Outline Package (SOP)
Quad Flat Package (QFP)	Quad Flat Package-Ext (QFP-EP)	Quad Flat No-leads (QFN)
Micro Ball Grid Array (μBGA)	Ceramic Ball Grid Array (CBGA)	Ball Grid Array (BGA)
Plastic Ball Grid Array (PBGA)	Fine-Pitch Ball Grid Array (FBGA)	Chip Scale Package (CSP)
Small Outline Transistor (SOT)	Plastic Leaded Chip Carrier (PLCC)	Dual Flat No-leads (DFN)
Small Outline Diode (SOD)	Single In-line Memory Mod (SIMM)	Dual In-line Mem Mod (DIMM)
Small Outline J-Lead (SOJ)	Quad Flat Pack No-lead (QFN-ML)	SOT-23
SOT-223	UltraThin Quad Flat NoLead (UQFN)	SOT-363
SOT-323	Small Outline IC (SOIC)	SOT-523

Additionally, there are countless sizes for resistors, capacitors, inductors, diodes, transistors, and every other circuit component. In order to make a PCB where all the parts fit, we need to have a basic familiarity with the dimensions that correspond to a given name. Resistors and capacitors are simple because the name refers to the dimensions in hundredths of an inch, so a 0603 capacitor or resistor measures 0.06 in × 0.03 in, and a

0805 part measures 0.08 in × 0.05 in [94]. A footprint, aka a land pattern, is a physical layout that represents the electrical and mechanical components of a board. Most footprints we will not need to draw ourselves because they come in the library of the Electronic Design Automation (EDA) software, but it's important to be aware of variations and check the dimensions if uncertain. For example, SOIC and TSSOP packages come in many different widths, and if the wrong one is selected the IC will be too wide or too narrow to fit on the board. Some components use strange footprints that would not be found in a library, such as the SOT-8072 footprint of the Gallium Nitride (GaN) transistors that we plan to use in the motor controller, pictured below.

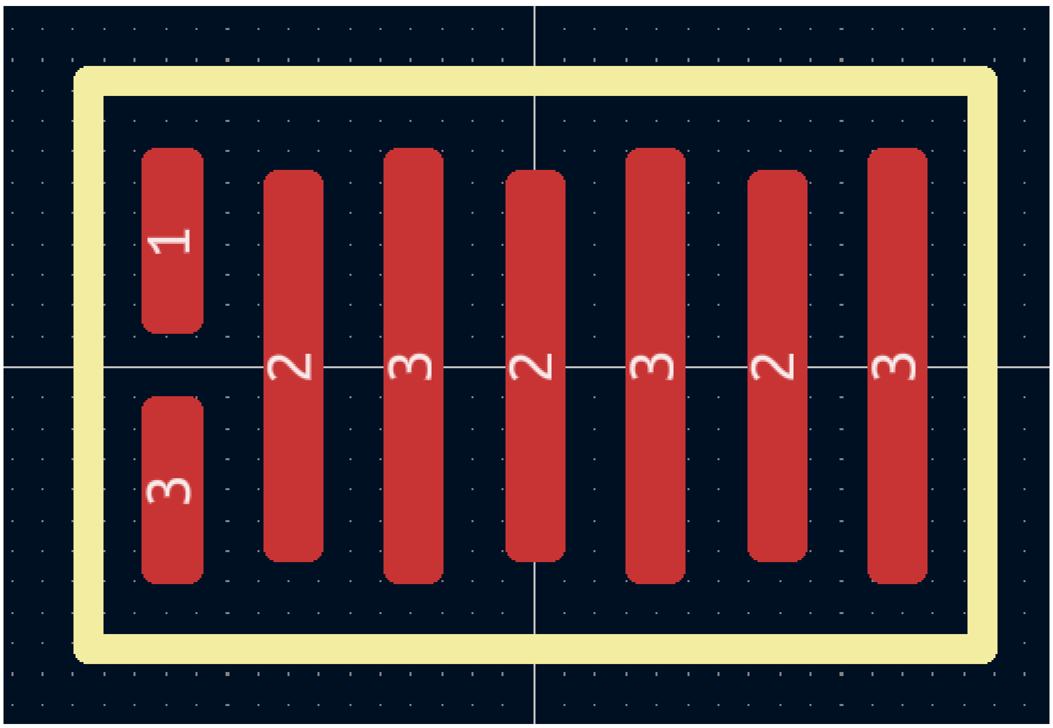


Figure 4.1 - SOT-8072 footprint or land pattern, drawn in KiCad 6.0

4.1.6 EN 55032: EMC Emission Limits for Multimedia Equipment

Switch mode power supplies have switching frequencies in the range of 50 kHz to 2 MHz, but the squarewave pulses produced by the power switches have higher frequency components. The sharp voltage transitions and high dv/dt in the power stage create high frequency noise of high enough magnitude that it can conduct to other devices through input and output power lines, and at high enough frequency it can also radiate. Regardless of the circuit's intended function, if it has wires that carry high dI/dt pulses of current in loops, the loops will act as antennas. There are many standards for Electromagnetic Compatibility (EMC) from the perspectives of minimizing emissions and reducing sensitivity to emissions, and systems are tested to different standards depending on what environment they will be used in. The international standard for EMC is IEC 61204, but we are unable to access that and it may have been superseded. In the US, FCC Part 15 Subpart B establishes conducted emission limits that are equivalent to the EU standard

EN55032, and the latter is easier to access. EN55011 applies to Industrial, Scientific, and Medical (ISM) equipment, EN55015 applies to lighting equipment (such as LED bulbs, dimmers, and fluorescent ballasts), and EN55032 applies to multimedia equipment, which includes the Information Technology (IT) that covers all power supplies [95].

EN55032 establishes limits for conducted and radiated emissions, and we will only measure conducted emissions because measuring radiated emissions would be prohibitively difficult with the tools we have available. Products that are used in a residential environment fall under the stricter Class B limits because they are expected to be closer to other devices, and all other environments are Class A. Since this charger will be used outside at a race track, we will attempt to comply with the Class A limits, which range from 150kHz to 30MHz as shown in the table below [96]. There are different limits for the average voltage and the quasi peak voltage, with the latter simulating an AM radio receiver, and we will need to find a spectrum analyzer with these input modes to perform the test.

Table 4.2 - EN5022 and FCC Part 15 Conducted EMC Limits

Frequency Range (MHz)	Limit dB (μV)	
	Quasi Peak	Average
0.15 to 0.5	79	66
0.5 to 30	73	60

4.1.7 IEC 61000-3-2: EMC Limits for Harmonic Current

IEC 61000-3-2 is an international standard that specifies limits for harmonic current emissions produced by equipment that is connected to the power grid. This is different from conducted and radiated EMC and EMI, which are high frequency signals that disrupt common communication frequencies. Technically EMC signals are also harmonics, but they produce negligible current, so in the power system "harmonics" are integer multiples of the fundamental frequency that are only 10-40x higher. The largest currents occur with the lowest harmonics, and these currents are problematic because they cause additional heating in transformers without transporting energy, reducing the efficiency of the grid. This standard is crucial for ensuring the efficient use of energy and minimizing the negative impact of electrical equipment on the power supply system. Specifically, IEC 61000-3-2 addresses the limitation of harmonic currents injected into the public supply system by the equipment under specified load conditions. Harmonics can be seen on an oscilloscope as a high frequency wave riding on the fundamental, or more commonly as a distorted waveform, such as that caused by a rectifier and capacitor. Compliance with this standard helps in reducing these disturbances and maintaining the quality and reliability of the power supply.

Manufacturers of electrical equipment typically use this standard to ensure that their products meet the required criteria for harmonic emissions. Compliance with IEC 61000-3-2 is often a mandatory requirement for certain types of electrical and electronic devices to be sold in various international markets. Part 3-2 applies to equipment with a rated current up 16A, which includes the 15A converter that we plan to build, and higher power equipment uses Part 3-12. Within Part 3-2, our converter falls under Class A, so the harmonic current limits in the following table apply [97]. We will test for compliance with a current probe on the AC line connected to an oscilloscope in FFT mode.

Table 4.3 - IEC 61000-3-2 Class A Maximum Harmonic Currents

Even Harmonics		Odd Harmonics	
Harmonic Order (n)	Max Current (A)	Harmonic Order (n)	Max Current (A)
2	1.08	3	2.3
4	0.43	5	1.4
6	0.3	7	0.77
$8 \leq n \leq 40$	0.23 x 8 / n	9	0.4
		11	0.33
		13	0.21
		$15 \leq n \leq 39$	0.15 x 8 / n

4.1.8 IEEE 1573

IEEE 1573 provides a technical basis for designing electronic power subsystems using a systems engineering approach. It covers DC, single phase, and three phase systems for power levels up to 20kW, voltages up to 600V, and switching frequencies over 1kHz. It first discusses element-level parameters, which are applicable at the element or system level and are commonly tested at the system level [98]. It gives definitions and test methods for:

- Voltage accuracy
- Contact resistance
- Undervoltage and overvoltage protection
- Over Temperature shutdown
- Reverse voltage protection
- Line, load, and temperature regulation

The bulk of the standard is a list of definitions and measurement techniques for different parameters of interfaces, sorted into four categories: Electrical, mechanical, environmental, and system [98]. The idea of systems thinking is that systems are constructed by specifying the interface parameters for each element.

The electrical interface consists of connections and interactions between elements that can be described by electrical parameters, such as voltage, current, and power. It also includes control signals that will change the operating conditions and the means of measuring the element, such as test points.

The mechanical interface consists of connections and interactions between elements that can be described by mechanical parameters, such as size, mounting methods, thermal aspects, and internal and external connections.

The environmental interface consists of the interactions between elements that can be described by environmental parameters, such as temperature, altitude, and humidity. There may be different environments that the system must operate in, and environmental conditions influence life expectancy, robustness, and reliability.

The system effectiveness interface consists of logistics, support, and life cycle parameters created by design trade-offs and interactions among the three other interfaces. It also includes issues that are not covered in the three other interfaces such as product life, warranty, system configuration management, cost, schedule, etc.

4.1.9 IEC 62368-1

IEC 62368-1 is an international safety standard for audio, video, information, and communication technology equipment, including the power supplies for these devices. It's used as a guideline for manufacturers to ensure that their products meet the safety requirements necessary for global distribution. This standard was developed to unify the safety standards for information technology equipment (ITE) and audio-visual equipment (AV). These were previously covered by two standards, IEC 60950-1 for the safety of information technology equipment and IEC 60065 for the safety of audio, video, and similar electronics. By combining the safety requirements of these two previous standards, IEC 62368-1 aims to provide a comprehensive and unified set of safety guidelines for a wide range of modern technology products.

The standard covers a broad range of safety aspects, including protection against electric shock, fire hazards, and energy hazards. It emphasizes a risk-based approach, focusing on the identification and mitigation of potential hazards in electronic equipment. Anyone who interacts with the equipment is classified as a skilled person, instructed person, or ordinary person. The main idea of this safety standard is that under normal or abnormal operating conditions or a single fault condition, ordinary persons should not be exposed to energy sources capable of causing pain or injury. Skilled persons are expected to be able to recognize energy sources capable of causing pain or injury and to take action to protect themselves, as well as having Personal Protective Equipment (PPE) that protects them against unintentional contact. Energy sources are classified as Class 1, 2, or 3 according to their effects as shown in the table below [99].

Table 4.4 - Response to Classes of Energy Source		
Energy Source	Effect on the Body	Effect on Combustible Materials
Class 1	Not painful, but may be detectable	Ignition not likely
Class 2	Painful, but not an injury	Ignition possible, but limited growth and spread of fire
Class 3	Injury	Ignition likely, rapid growth and spread of fire

More details on the classification process are given in Section 5.2. A Class 1 electrical energy source (ES1) can have a touch voltage or current over the ES1 limits and still be an ES1 source, but if *both* voltage and current are over the limits it becomes an ES2 source. If both exceed the ES2 limits it is an ES3 source, and the thresholds are shown in the figure below [99]. By the given criteria, everything with a low impedance path to mains voltage is an ES3 source, and the output of the charger is an ES2 source because the voltage limit is over 60V.

Energy source	ES1 limits		ES2 limits		ES3
	Voltage	Current [a, c, d]	Voltage	Current [b, c, e]	
DC [c]	60 V	2 mA	120 V	25 mA	> ES2
AC up to 1 kHz	30 V RMS 42,4 V peak		50 V RMS 70,7 V peak		
AC > 1 kHz up to 100 kHz	30 V RMS + 0,4 f 42,4 V peak. + 0,4 $\sqrt{2}\,f$	0,5 mA RMS 0,707 mA peak	50 V RMS + 0,9 f 70,7 V peak + 0,9 $\sqrt{2}\,f$	5 mA RMS 7,07 mA peak	
AC above 100 kHz	70 V RMS 99 V peak		140 V RMS 198 V peak		
Combined AC and DC	$\frac{U_{DC}(V)}{60} + \frac{U_{AC\ RMS}(V)}{U_{RMS\ limit}} \leq 1$ $\frac{U_{DC}(V)}{60} + \frac{U_{AC\ peak}(V)}{U_{peak\ limit}} \leq 1$	$\frac{I_{DC}(mA)}{2} + \frac{I_{AC\ RMS}(mA)}{0,5} \leq 1$ $\frac{I_{DC}(mA)}{2} + \frac{I_{AC\ peak}(mA)}{0,707} \leq 1$	See Figure 23	See Figure 22	

As an alternative to the requirements above, the values below can be used for purely sinusoidal waveforms

Energy source	ES1 limits	ES2 limits	ES3
	Current [c] RMS	Current [c] RMS	
AC up to 1 kHz	0,5 mA	5 mA	> ES2
AC > 1 kHz up to 100 kHz	0,5 mA × f [d]	5 mA + 0,95 f [e]	
AC above 100 kHz	50 mA [d]	100 mA [e]	

Figure 4.2 - Electrical Energy Source Limits for ES1 and ES2

A safeguard is a device, scheme, system that goes between a person and an energy source capable of causing pain or injury that reduces the likelihood of energy transfer. Among the three general types, equipment safeguards do not require any knowledge or actions to use, installation safeguards are only useful after installation, and behavioral safeguards are useful when the energy source must be accessible. The most common basic safeguard against an electrical energy source is basic insulation, such as the insulation on wires, and protective bonding and protective earthing can be used as a supplementary safeguard. Double or reinforced insulation can also be used as a supplementary safeguard, and it is used in power supplies where a ground connection is not available. Power supplies that use a grounded metal housing are Class I, and power supplies with an insulated plastic housing and double insulation are Class II [100]. Class I construction is more common at high power levels, and we will build a Class I power supply due to the superior thermal conductivity of metal.

For the required safeguards, no safeguards are required between an ES1 energy source and an ordinary person. At least one basic safeguard is required between an ES2 source and an ordinary person, and if the basic safeguard is defeated for maintenance, there must be an instructional safeguard. A basic safeguard and a supplementary safeguard are required between an ES3 energy source and an ordinary person, and examples of supplementary safeguards include additional insulation, a fire enclosure, and an overcurrent protective device. No safeguards are required between an ES2 or ES3 source and a skilled person because it is assumed that they have a skill safeguard.

We have established that enclosure grounding and basic insulation are adequate safeguards, but what counts as enough insulation? Insulation consists of insulating materials, clearances, creepage distances, and solid insulation (potting resin) that is providing a safeguard function. Different classes of insulation have different temperature limits, so the correct type should be selected to prevent its temperature limit from being exceeded in operation. Creepage is the distance between two conductors on the surface of the board or along the surface of the insulating material. Clearance is the line-of-sight distance between two conductors through the air [101]. The amount of pollution, aka dirt and debris, that could be inside the device affects the minimum required creepage distances. The charger will be used outside, so it is reasonable to assume the highest pollution level, which is 3, and according to Table 10 in the standard, the minimum clearance for 400V is 0.8mm for basic insulation and 1.5mm for reinforced insulation. According to Table 17, the minimum creepage distance under the same conditions is 6.3mm for basic and reinforced insulation.

Finally, to protect the input against voltage spikes, there must be a varistor rated to at least 125% of the power supply's maximum rated voltage with a disc diameter of 14mm to 20mm [99]. Specification UL1449 for Surge Protective Devices (SPD) requires that varistors be put in series with a Thermal Cut Off (TCO) device to prevent failure when clamping a voltage spike, but IEC 62368-1 does not require it. In summary, compliance with IEC 62368-1 ensures that electronic devices are designed and manufactured with a high level of safety, reducing risks for end-users and providing a standardized framework for global trade.

4.2 Rulebook Constraints

The overall goal of the Electrathon rulebook is to promote safe and fair competition, and each rule fits into one of those categories. Given that most competitors are high school students with little to no experience building race cars, most of the rules focus on safety. Given that we plan to compete in these races, all of the rules in the book are also constraints on our design. The constraints on the electrical system are the most relevant.

The rulebook has two performance constraints, and both involve the battery. First, the combined capacity ratings of all cells used in the battery pack must add up to less than 1kWh according to the manufacturer's data, typically provided on a datasheet [1]. Second, the weight of the battery cells or blocks in the pack, the parts that come from the manufacturer, must be less than the weight limits shown in the following table [1]. Different battery chemistries have different energy densities and thus different weight limits to accommodate them. Lithium-ion has the highest energy density and thus the lowest weight limit at only 15 lbs. Cell holders, cases, wires between cells, etc are not included in the weight limit.

Table 4.5 - Allowable Battery Weights			
Battery Chemistry	Maximum Allowable Cell Weight (lbs)	Maximum Allowable Cell Weight (g)	Maximum Number of 49g 18650-size Cells
Nickel-Metal-Hydride	41	18,597	379
Silver-Zinc	23	10,433	212
Nickel-Zinc	44	19,958	407
Nickel-Iron	58	26,308	536
Lithium-Ion	15	6,804	138
Lithium-Iron-Phosphate	29	13,154	268

For safety, the rules are geared towards ensuring that the car is easily controllable and does not have an electrical fire. Remote control of vehicles is not permitted, and "power to the motor must be controlled by the driver and turn off automatically when the driver releases the accelerator (dead man cut-off)" [1]. This constrains the types of driver assistance systems that can be designed. To prevent fires, all circuits connected to the battery must have a fuse or circuit breaker, and the fuse or breaker must be sized to protect the wiring on the circuit. Wire ampacity and fuse sizing is shown in the table below, which follows the National Electric Code (NEC) Handbook for standard automotive cable, single conductor and not in a raceway or conduit [102].

Table 4.6 - Required Fuse and Wire Sizing												
Fuse or Breaker Size (amps)	5.5	9	12	15	20	30	80	105	140	200	250	300
Wire size (AWG)	20	18	16	14	12	10	8	6	4	2	1	1/0

To disconnect the battery in case of a crash or unintended operation, each car is required to have an isolation switch (kill switch) with a break current rating that exceeds the maximum current draw of the vehicle. The switch must be connected to the positive terminal of the battery; interrupting the ground is not allowed. The switch must be accessible from inside or outside the vehicle, so the driver must be able to actuate it from driving position without reaching outside, and race officials must be able to actuate it from outside without reaching in. To meet this requirement, one or more switches can be installed in series. To ensure that it can be found quickly, the actuator located on the outside of the vehicle must be mounted within a solid red triangle at least 4 inches on a side and whose color is in contrast to the vehicle's color and graphics. The circuit breaker may function as the isolation switch if it can be manually tripped [1].

For the wires, the rulebook requires them to be insulated and secured to the frame so that they are protected from moving parts and chafing. There must be rubber grommets where wires pass through holes with sharp edges or through sheet metal. Terminals must be tightly secured, and the vehicle frame must not be grounded.

The final racing requirement is that the car must survive on the courses that it is raced over, and while this is not in the rulebook per se, it's an intrinsic part of racing in the Tampa Bay series. Most races are held in high school parking lots where the asphalt is far from smooth, and the cars spend the hourlong races bouncing over cracks, bumps, and potholes. The number of these that the car must endure depends on the driver's skill at avoiding them, but many go across the entire width of the course and are unavoidable. Thus, all electrical components should be rigidly mounted to the frame with screws and mounting holes. Items held down with tape or adhesive will likely come loose during a race.

4.3 Other Design Constraints

This section covers other limitations of the project, including cost of parts and transportation, timeline, and equipment storage, as well as safety, manufacturability, and environmental constraints.

4.3.1 Economic Constraints

This project faces economic constraints because racing is expensive in general, and while Electrathon is a relatively inexpensive form of racing, it still costs at least $1000 to $2000 to build a car. Steve Archer, a vice president of Electrathon America with a lot of experience building cars, recently set out to build a car as cheaply as possible while still being somewhat competitive. By using an unsuitable low discharge rate battery, a cheap Ebike motor controller, and a Ryobi lawn mower motor, he kept the electrical system cost down to only $509 out of the $1297 total cost [103] [104]. Teams typically spend a few thousand dollars on a car, out of which approximately several hundred goes to the batteries, a few hundred to the motor, a few hundred to the motor controller, and a hundred dollars to the wires, fuses, disconnects, and gauges.

Besides simply making a functional electrical system, the goal of this project is to produce an optimized electrical system that is very competitive against the best cars in the Tampa series. The economic constraint is exacerbated by our lack of a corporate sponsor, and the project is entirely self funded by the senior design group, primarily by the student who will own the car at the end. To save money while increasing performance, we will use our electrical design and programming skills to build components ourselves instead of buying them. Specifically, the battery charger, motor controller, battery management system, and driver information system are all parts that are typically purchased, but we will build them ourselves.

There is no fixed budget, but when comparing parts we will try to avoid the point of diminishing returns where 2x or 3x more money only improves performance by 1% or 2%. For example, a prominent racer in the Tampa Electrathon series uses a brand of motors that are wound by hand instead of by machines for better quality control, but it doubles the motor price. More expensive parts are often only backed up by a brand name, not performance data, so whenever possible we will test parts and compare data to make our decision. We do not have the luxury of buying multiple parts and only putting some of them on the vehicle, so when we buy parts we will quickly test them and return them if they do not meet their advertised specifications. For each part we will make an informal cost benefit analysis to determine if the additional performance is worth the higher price.

Another cost to consider is transportation. The Tampa Bay Electrathon races are held in Plant City, which is about 77 miles from Orlando. We will need to pay for gas to travel to each race, and possibly a vehicle rental as none of our group members own a reliable vehicle that is well-suited for transporting the go-kart.

4.3.2 Time Constraints

This project faces time constraints because it must be completed in only two semesters, and the students in the group have other classes and obligations, such as jobs and athletics. In contrast, the product development cycle of a company is measured in years, and that is with teams of engineers working on the problem. Furthermore, we are essentially developing multiple products because each component in the electrical system could be considered its own device. Given our limited time, money, and experience, it would be highly unrealistic for us to expect our circuits to perform as well as those made by professionals in a better environment.

We will address these time constraints by limiting the scope of the project and using familiar integrated design tools wherever possible. We chose a Texas Instruments (TI) brand of microcontroller because our programmers are familiar with Code Composer Studio (CCS), and TI's software toolchain should be able to communicate with the boards and open a debug window without us needing to think about drivers and bootloaders. For the battery charger, we are using TI Power Stage Designer to size the components in the power stage and using the ExcellentIT SMPS Transformer calculator for the magnetics design [105] [106]. For all other voltage converters we are using TI Webench Power Designer to generate the schematic and component values [107]. It would be possible to calculate all of these values from datasheets, and group members have experience doing this, but calculation software saves time. All schematic capture and PCB design activities will be done in KiCad because it is a familiar and intuitive software [108].

4.3.3 Equipment Constraints

This project faces equipment constraints because there is no secure storage area for the car on campus. Fortunately, the local Electric Vehicle (EV) enthusiast Larry Wexler has donated the use of his garage and tools, but these are hand tools and have limited precision. We can make small parts in the UCF machine shop, but these must be transported to and from the car, and using the machine shop has a long lead time.

For electrical equipment, students in the group have cheap versions of the necessary tools (multimeters, current shunts, oscilloscopes), but we will need to go to the senior design lab to use the higher end versions. The lab likely does not have the more specialized test equipment that would be needed for professional-grade power electronics development, so we will design our test procedures to use more common tools. For electronic components such as transistors and integrated circuits, we will skip separate testing unless it is required for troubleshooting. We will not attempt to measure the Miller capacitance and rise time of the power stage, and we may or may not be able to do double-pulse testing. Double-pulse testing is a standard test-method used to evaluate the performance and characteristics of power semiconductor devices, such as insulated gate bipolar transistors (IGBTs) and power MOSFETs. It involves applying two pulses of different polarities to the device under test in order to assess its switching behavior and performance in realistic operating conditions [109].

4.3.4 Safety Constraints

There are many safety hazards associated with building and racing vehicles, but they can be greatly mitigated with a well-designed rulebook and good safety culture. The majority of the rulebook covers the mechanical design of the car, requiring items such as a roll cage that fully encloses the driver, a full face motorcycle helmet, a five point harness, and castle nuts with safety wire to prevent the wheels from coming off. The race organizers understand the risk of spectators or crew being hit by the cars, so they have strict rules on who can be on the track at a given time, rules that are continually enforced. The main electrical hazards involved in this project are shock and fire, and these can be prevented by insulating all live conductors and stopping overcurrents before parts can overheat.

For the battery, we will address safety constraints by ensuring that the Battery Management System (BMS) circuit cuts off charging at 4.2V for each cell. The most common cause of battery fires is overcharging, and a properly designed charger and protection circuit can prevent this. We will also use a type of battery that is durable enough to easily withstand race conditions, as well as rapid charging and a bit of overvoltage. For example, some Electrathon racers use Lithium Polymer (LiPo) battery cells because they offer the lowest internal resistance, but this is an unsafe choice because LiPo batteries are fragile and prone to catching fire at the slightest provocation.

For the DC-DC converters, we will implement the controller's cycle-by-cycle current limit feature (if available) to reduce the likelihood that a short circuit leads to overheating. We will follow the relevant standards on insulation, and the AC-DC converter will have a fuse and Metal Oxide Varistor (MOV) on the input to protect against voltage spikes. All conductors with greater than logic voltage will be insulated to prevent accidental contact, and most importantly, the components will be treated gently. These are prototypes, not consumer electronics products, so we will be careful to not subject them to undue stress or put them in harsh environments.

4.3.5 Manufacturability Constraints

Manufacturability constraints are important to this project because electrical components occupy physical space, so besides the circuit diagrams, we need to plan out part locations to ensure that there are no conflicts. Parts should be removable for debugging, possibly with screws, and there needs to be enough space above the screw for a screwdriver to fit. For example, to cite a common problem, there shouldn't be a heatsink in the way. There are many different ways to build things, and to meet cost and time constraints we need to find the cheapest and quickest method. Every PCB that goes on the car will need to be inside an enclosure and while machining these out of a block of aluminum would give the best performance, it's not practical. Bending and welding sheet metal would be a more manufacturable design, and these tradeoffs continue with every part.

In the realm of PCB design and assembly, there's a strong tradeoff between component density on the board and how easy it is to solder by hand. We will aim for a minimum of 4mm between components because this leads to much easier assembly than tighter clearances of 1-2 mm. Our group has no problems with soldering SMD components, which is fortunate because as the industry moves towards smaller and lighter packages, through hole components are becoming hard to find. For packages, we will use 0805 for most resistors and capacitors, not going any smaller than 0603, and the Integrated Circuits (ICs) will have a 1.27mm pin pitch whenever possible. Pin pitch is the center to center distance between pins, so distance between the edges is less, and if too much solder or solder paste is used it can bridge between pins via surface tension. ICs with 1.27mm pin pitch have a much wider margin of error for the amount of solder paste used and reduced chances of bridging compared to the other standard sizes of 0.95mm, 0.65mm, and 0.5mm pin pitch. To minimize the chances of needing to redesign because parts go out of stock, we will only order components with a reasonable stock level on major part distribution websites. Given our low volume application, our minimum stock threshold can be only 10,000 for common resistors and capacitors and 1,000 for FETs, diodes, and ICs. Boards will be connected by plugs instead of directly soldering wires to keep the system modular.

4.3.6 Environmental Constraints

The environmental constraints on this project are that the car should not generate waste and should instead create excitement for the coming transition to electric vehicles. Electrathon can be more sustainable than most existing racing series because the cars are smaller, so they take fewer resources to build, and they focus on efficiency, so they go through consumables more slowly. Racing series that focus on maximizing power and traction go through fuel, tires, and brake pads very quickly, not to mention the wear and tear on other parts of the car. For example, top fuel dragsters need a new transmission after every race!

Electrathon is more sustainable because the cars are much lighter and the driver minimizes acceleration and deceleration, so the tires wear out slowly. Teams commonly run the same set of tires across a whole season, which is 18 races. Brake pad wear is almost unheard of because the brakes represent wasted energy. The cars also do not burn fuel, and the 1kWh of electricity they consume has minimal environmental impact. However, it's important to note that for some kinds of batteries, such as Lithium Polymer (LiPo), performance drops rapidly with age, noticeably reducing after 5-10 cycles. To mitigate this, some racers replace their battery after 10 races, so the battery becomes a consumable and generates another waste stream.

To meet our environmental and sustainability goals for this project, we chose a longer lasting battery cell chemistry that maintains top performance over 500 to 1000 cycles. When the battery eventually wears out, we will bring it to an electronics recycling center. Lithium ion cells do not contain environmentally hazardous elements such as lead or cadmium [110], but battery recycling is important for recovering lithium from the electrodes. Due to surging demand from electric vehicles and grid storage batteries, the world is facing a lithium shortage that is expected to be strongly felt by 2025 [111]. Going forward, battery recycling will be important for ensuring an adequate supply of lithium and minimizing the amount of mining that needs to be done, which is important because mining is environmentally destructive.

The second category of environmental constraints is that the car should promote the adoption of electric vehicles (EVs) by creating excitement among the next generation of consumers and engineers, and by demonstrating advanced technologies that can improve EV efficiency. The first constraint can be met by demonstrating the car on campus and drawing sustainability focused art on the body panels, and the second constraint can be met by testing innovative solutions to technical problems. For example, our motor controller will use Gallium Nitride (GaN) power transistors, an advanced wide band gap semiconductor that is being investigated for use in EV motor controllers but likely won't see mass production for another 5-6 years [112].

5 Uses of ChatGPT

With the rapid growth of AI tools such as Generative Pre-Trained Transformers (GPTs) in recent years, it is worth analyzing how they may be applied to this project. Clearly, AI is not yet advanced enough to design, document, and build an electric car from start to finish. From a hardware standpoint in general, its capabilities are fairly limited, as will be discussed in this chapter. Still, GPTs can be useful in several aspects of project development, particularly research, documentation, and some cases of software prototyping.

ChatGPT can be useful for documentation drafting in some generic cases. Here, ChatGPT provides a list of pros, cons, and limitations of using AI software for engineering projects:

"*Pros:*

1. *Access to vast knowledge: AI tools like ChatGPT have access to extensive databases of information and can quickly provide answers to a wide range of questions related to electrical and computer engineering. This can be especially useful for research and learning.*
2. *24/7 availability: These tools are available round the clock, allowing engineers to seek assistance and information at any time, which can be crucial for meeting project deadlines.*
3. *Enhanced productivity: ChatGPT can help automate repetitive tasks, such as data analysis, code generation, and documentation, which can significantly improve productivity.*
4. *Quick problem-solving: ChatGPT can assist engineers in troubleshooting and problem-solving by suggesting potential solutions based on available information and best practices.*
5. *Educational resource: These tools can serve as educational resources, helping students and professionals learn about complex concepts, theories, and technologies in the field.*

Cons:

1. *Lack of context: ChatGPT may not fully understand the specific context of a project, which can lead to incorrect or inappropriate recommendations. It may not consider project-specific constraints or safety requirements.*
2. *Limited creativity: AI tools are primarily data-driven and lack creativity or innovation. They may not provide unique or out-of-the-box solutions to complex engineering problems.*
3. *Over-reliance on AI: Relying too heavily on AI tools can lead to a decrease in critical thinking skills and problem-solving abilities among engineers.*
4. *Privacy and security concerns: Sharing sensitive project details or data with AI tools can raise concerns about data privacy and security, especially when dealing with proprietary or confidential information.*

5. *Cost and accessibility: Access to advanced AI tools may require a subscription or payment, which can be a barrier for students or professionals with limited budgets.*

Limitations:

1. *Inaccuracies: AI tools can produce inaccurate information or suggestions, especially if they are not updated regularly or if the input data is flawed.*
2. *Inability to replace expertise: While AI can assist, it cannot replace the expertise of experienced engineers who have an in-depth understanding of complex systems and can make nuanced judgments.*
3. *Language barriers: AI tools may struggle with language barriers, and communication can be challenging if not conducted in a common language or if technical jargon is involved.*
4. *Dependence on available data: The quality and relevance of information provided by AI tools depend on the data they have been trained on. If certain topics or emerging technologies are not well-represented in the training data, the AI may not provide accurate information.*
5. *Ethical concerns: AI tools can inadvertently propagate biases present in their training data, leading to biased recommendations or outputs. This can have ethical implications in decision-making processes."*

While a human writer may take an hour or more to draft such a list, ChatGPT accomplishes the task in only a few seconds. This is a major asset to productivity when used appropriately. Still, ChatGPT is not a complete substitute for human expertise. More specific details of the project must be documented manually by the team. This includes engineering specifications, descriptions of hardware and software components used, and explanations of various decisions made throughout the development process.

A few of the key limitations of ChatGPT that apply to this project are lack of context, lack of creativity, and inability to replace expertise, as mentioned by ChatGPT itself. Contextual details of the project are often necessary to provide accurate assistance. If asked to suggest a "good" microcontroller with no other context, ChatGPT will return a generalized explanation of different types of microcontrollers, along with potential applications they may be suited for. Whether or not any of those microcontrollers would be "good" for the intended application will depend greatly on what that application is. In some cases, even if the project requirements are defined in great detail, ChatGPT is still only able to provide generalized guidance, like suggesting a family of microcontrollers rather than individual models. It does not know which models are currently available or what new products have been recently released.

Technical expertise and creativity are essential for complex engineering projects, and cannot yet be replaced by GPTs. Both hardware and software design and implementation require a great deal of background knowledge. ChatGPT can be one of many tools used to help with the process, but ultimately the overall solution is left to the human engineer. In the following sections, applications of ChatGPT relevant to this project are analyzed.

5.1 Research

GPTs can serve as a helpful starting point for research, like a custom-generated literature review. Rather than parsing through several sources individually, the GPT can accomplish this task for the user in just a few seconds (depending on the complexity of the inquiry) and return a summary of relevant information.

In this project, ChatGPT was used to assist with part selection by providing guidance on various types of microcontrollers. Early in the research process, the team was considering whether it would be better to use an Arduino or MSP430 to program the LCD, so ChatGPT was asked to compare the two microcontrollers:

"Arduino:
Benefits:

1. *Ease of Use: Arduino is known for its beginner-friendly development environment and a large community of users. It's an excellent choice for those new to embedded programming.*
2. *Abundance of Libraries: Arduino has a vast ecosystem of libraries and code examples readily available, making it easy to interface with various sensors and devices.*
   Hardware Abstraction: Arduino abstracts many hardware details, simplifying the development process. This can be an advantage for quick prototyping.
3. *Cross-Platform Support: Arduino IDE supports multiple operating systems, making it accessible to a broad audience.*
4. *Wide Hardware Variety: There are many Arduino board options, including those with built-in Wi-Fi (e.g., ESP8266, ESP32) and advanced features for various applications.*

Drawbacks:

1. *Limited Low-Power Features: Arduino boards are not typically optimized for ultra-low-power applications. While you can put them into sleep modes, they may not be the best choice for battery-powered devices that require extreme power efficiency.*
2. *Hardware Limitations: Arduino boards have fixed hardware, limiting your ability to select the best microcontroller for a specific application.*
3. *Cost: Arduino boards can be more expensive compared to MSP430 microcontrollers when considering the cost of additional hardware features you may not need.*

MSP430:
Benefits:

1. *Ultra-Low Power: The MSP430 family is designed for ultra-low-power applications, making it an excellent choice for battery-powered devices where power efficiency is critical.*
2. *Customization: You have more control over the hardware and can select specific MSP430 microcontrollers that match your project's requirements closely.*

3. *Hardware Flexibility: MSP430 microcontrollers often offer a wide range of peripherals, allowing you to tailor your design to the specific needs of your project.*
4. *Cost-Efficiency: Depending on your project's requirements, MSP430 microcontrollers can be more cost-effective compared to Arduino boards.*

Drawbacks:

1. *Steep Learning Curve: If you're new to microcontrollers and embedded programming, the MSP430 might have a steeper learning curve compared to Arduino.*
2. *Limited Community and Libraries: While the MSP430 has an active community, it's generally smaller than the Arduino community. You may have to write more of your code from scratch.*
3. *Development Environment: Setting up an MSP430 development environment might require more effort compared to the Arduino IDE, which is known for its simplicity.*

In summary, Arduino is a great choice for beginners and rapid prototyping due to its ease of use, extensive libraries, and community support. However, for projects where power efficiency is crucial and hardware customization is required, the MSP430 may be a better choice. It's essential to evaluate your project's specific needs, your familiarity with each platform, and your budget when making a decision between Arduino and MSP430."

ChatGPT gave a broad overview of each microcontroller brand, listing the benefits and drawbacks of each in a clear, concise manner. Further utilization of ChatGPT for research purposes is noted in Chapter 3 (Research) of this report.

Unfortunately, ChatGPT often provides inaccurate or incomplete information. While exploring microcontroller options, ChatGPT described Texas Instruments' TMS3202803x family as being part of the "Concerto" series when it is actually from the "Piccolo" series, also described in Chapter 3 of this report. Additionally, since ChatGPT only has knowledge up to 2021, it cannot provide information on the latest hardware or updated features that may have been released since then.

5.2 Development: Hardware

The team's electrical engineering majors are at a disadvantage in terms of technical support; ChatGPT is particularly unhelpful when it comes to hardware development.

Because ChatGPT is only trained on data up to 2021 and cannot access the internet in real-time, it is unable to return links to websites or youtube videos that may provide educational content/tutorials to the user. Comically, the most it can do is advise the user on how to use Youtube - when asked if ChatGPT can suggest Youtube videos, its response includes:

"1. Use YouTube's Search Bar: Type keywords related to your interests into YouTube's search bar. For example, if you're interested in learning how to cook a specific dish, you can type "how to cook [dish name]" and see what videos come up."

ChatGPT also cannot generate images, which limits its ability to assist with hardware design and implementation. It can offer descriptive "guides" (though often vague), but many concepts are more clear when paired with visual aids. For example, constructing a physical circuit is typically easier when provided with a diagram of the board layout rather than a written list of which components are connected. When prompted to help design a printed circuit board for a motor controller, ChatGPT was able to generate a lengthy list of steps on the process: define requirements, make the schematic (it suggested using Eagle, KiCAD, or Altium Designer), choose components, route traces, run a Design Rule Check, generate gerber files, fabricate the PCB, and finally assemble and test the design. Theoretically, this could be a starting point for a student or hobbyist with no prior knowledge of circuit design, but the user would most likely have to move on to a more robust tutorial to better understand each individual step.

ChatGPT notes that "designing a PCB is a complex task that requires knowledge of electronics, PCB design software, and specific details about your motor controller project". The project's level of customization and the necessity of advanced technical knowledge greatly limit ChatGPT's ability to be of assistance.

The team found that prior knowledge and researching the "old-fashioned way" (via search engines such as Google, reading textbooks, or consulting with experts) was more effective for hardware research and development than currently available GPTs.

5.3 Development: Software

The team had a slightly more promising outlook on ChatGPT's software capabilities. One of ChatGPT's most well-known applications is rapid code prototyping, as it is able to generate code in most programming languages (C, Python, Java, etc.). It is highly effective for small, simple functions, such as swapping two elements in an array. In addition to the function requested, ChatGPT generally includes an example "main" function to demonstrate an application of the function.

For embedded programming, there is an added layer of complexity. While ChatGPT is still able to generate codes for embedded software in common languages such as C, the code generally will not run successfully simply by copying and pasting it into an IDE.

5.3.1 Demo 1: Countdown Timer for LCD Display

In this example, ChatGPT is prompted to generate code for an Arduino Mega 2560 that will count down a timer from 1 hour and display the output to the screen of a 3.5" TFT display in mm:ss format. Note that this demo was completed prior to the group's decision to develop a mobile app for the display rather than using an LCD. However, its sentiment remains relevant as a simple demonstration of ChatGPT's capabilities and limitations in regards to embedded programming. Observations from the demo were considered when comparing choices of display implementation.

First, ChatGPT points out that a TFT library compatible with the display will be needed, and suggests the "Adafruit GFX" and "Adafruit ILI9341" libraries, which can be easily installed in Arduino. In a separate prompt, ChatGPT provides a step-by-step tutorial on how to do this. ChatGPT provides an example setup using these libraries, but notes that the pin definitions and rotation will likely need to be adjusted for the specific setup.

In terms of algorithm alone, ChatGPT successfully produces code for a countdown timer that displays the output in the format specified. With several *non-trivial* modifications, the following output was achieved on HiLetGo's 3.5" TFT display for Arduino Mega 2560, where the text shown counts down as intended:

Figure 5.1 - LCD Output Based on Algorithm Provided by ChatGPT

The keyword here is "non-trivial" modifications - although the algorithm is correct, quite a bit of background knowledge and customization is required in order for the code to run on the actual display. The setup and initialization portions of ChatGPT's code were rewritten with <TFT_HX8357.h>, a library known to work with the display used in this project. The color definitions were changed to match the definitions from this library - for example, the TFT_HX8357 library uses "TFT_BLACK" for black rather than "ILI9341_BLACK". Other colors could be used by finding their definitions in the library documentation or defining a custom color with its appropriate hexadecimal value.

For the initial requested output, displaying a countdown timer on an LCD, the code generated by ChatGPT places the timer in a rather awkward position on the screen - the text is small and sits a few lines down. Consider the case where the user wants the timer to be displayed in the center of the screen in a much larger font, but doesn't know how to accomplish this. Can ChatGPT help?

ChatGPT keeps a log of each conversation, and in many cases can be guided to producing a more desirable output by adding more details to a previous prompt. Here, ChatGPT chose a default text size and location because it was not provided one. When asked to rewrite the code to meet the goals described above, ChatGPT produces a modified output that would theoretically accomplish the task for its example setup.

Unfortunately, implementing this solution is not so simple as copying and pasting the new code into the IDE. ChatGPT uses a function called "getTextBounds", which exists in the Adafruit library but not the TFT_HX8367 library. ChatGPT briefly explains that the portion of code shown below calculates the coordinates to center the text on the screen.

Now, since this code doesn't actually work with the TXT_HX8367 library, is ChatGPT able to modify the code again for an alternative library? Again, it is not quite that simple. When prompted, ChatGPT simply replaced instances of "ILI9341" in its code with "HX8367", while attempting to use functions and definitions that are named differently or not available in the requested library.

In this case, it may be easier to observe the modifications ChatGPT made to its example than to coerce ChatGPT into making its code work for the specific board and libraries used in the project. By reading the board and library documentation and using ChatGPT's code as *inspiration,* the desired output can be achieved:

Figure 5.2 - LCD Output After Manual Modifications

When asked how to determine a screen rotation value and pin definitions, ChatGPT recommends reading the datasheet for the display, which typically contains detailed information on pin connections, configuration options, and initialization commands. Its next suggestion is online resources (which, as previously mentioned, ChatGPT cannot access) or library documentation that may have examples of configurations for commonly used displays. Lastly, "experimentation" is offered as a possible method. ChatGPT cautions that the user risks accidentally damaging the hardware by doing this. Brute-force experimentation is likely to be both inefficient (with the exception of "lucky guesses") and dangerous, especially for users with little technical experience, who could damage their board by attempting to use incorrect pin connections.

Based on this example, it seems ChatGPT is best used for *algorithm* development, but less helpful for hardware implementation. That is, generic small functions (including the above examples) can be generated using ChatGPT, but additional work will be needed to incorporate these functions into the overall project. At a minimum, product datasheets will be necessary to properly configure each microcontroller used in the project. Online resources, such as tutorials and embedded software forums, will also be better suited for model-specific guidance.

5.3.2 Demo 2: Countdown Timer and Bluetooth for Flutter App

After transitioning to a display design using a mobile app with bluetooth connection rather than an LCD screen, a similar demo was completed to observe ChatGPT's potential to assist with the updated design. At this point, the choice of app development platform (Flutter, Thunkable, or React Native) has not yet been finalized, so Flutter was arbitrarily chosen for this demo, with the assumption that ChatGPT's capabilities would likely be comparable across platforms (the group did not have high hopes for the demo regardless, anticipating other reference materials would be more useful in the development process).

First, ChatGPT was asked if it could help read data from an ESP32 to a Flutter app. It returned a high-level overview of the process consisting of the following steps:

1. ESP32 Setup
2. Choosing a communication protocol (WiFi/HTTP, Bluetooth, or other wireless methods, as the prompt did not specify)
3. Flutter app development
4. Connecting to ESP32
5. Receiving data from ESP32
6. Data processing
7. Displaying data
8. Handling errors and implementing security features

This list is not particularly useful, as these steps are fairly obvious and ChatGPT does not elaborate on how to complete them. It does provide an example Dart code for HTTP with Flutter. Since the group intends to use Bluetooth for the project, ChatGPT was prompted to rewrite its example code for a Bluetooth connection. ChatGPT notes that the 'flutter_blue' package must be added to the Flutter project by including it in the 'pubspec.yaml' file. The example uses default names such as 'esp32DeviceName', 'Your_ESP32_Device_Name' and 'your_characteristic_uuid', which must be replaced with the appropriate names associated with the specific device. What if the programmer does not know their device's name or UUID, and doesn't know where to find it? ChatGPT can help with that - it advises reviewing the code running on the ESP32, which typically sets these parameters.

Next, ChatGPT was asked to write a Dart code for a basic one-hour countdown timer, as in the previous demo. ChatGPT is aware of a Flutter package 'flutter_countdown_timer' and libraries for this purpose. Using these, ChatGPT presents a code for a simple Flutter widget to run and display the timer. In this case, the code generated by ChatGPT results in an app with '60:00' frozen on the screen rather than counting down. Notifying ChatGPT of the error, it apologizes for its "oversight" and claims to have fixed the code, but the code is not in fact fixed and the app still fails to run as intended.

Unfortunately, this example suggests seeking guidance from ChatGPT for app development may lead to more confusion and frustration than solutions. Luckily, there are many human-authored tutorials available for Flutter, which are likely to be more useful in this project.

5.4 Ethics

One question that remains is to what extent it is “ethical” to use ChatGPT in this project. There are two main considerations to address: one, the resulting “product” (the go-kart) is being used for competitive racing, and two, the documentation for the project is part of an academic assignment.

In regards to the Electrathon competition, there are no rules against the utilization of GPTs. Participants are allowed (and encouraged) to use any resources available to them when designing their vehicles. Since ChatGPT is publicly available and largely *unhelpful* for designing and especially building a go-kart, its use poses no threat to race fairness. The main use of ChatGPT in this project only applies to an optional addition to the go-kart - the LCD display - which is not a required feature. It is intended to provide assistance to the driver at a high level, but does not impact the mechanical performance of the vehicle.

In terms of academic integrity, the ethics of using ChatGPT are somewhat fuzzier. It has already been established that ChatGPT is mostly useless for hardware development and only moderately useful for software development. The main area of concern would be the research and documentation portions of the project. At the beginning of this chapter, a ChatGPT-generated list of pros, cons, and limitations was shown, pointing to the software’s productivity-enhancing potential for drafting documentation. Whether or not it is ethical to “copy” from ChatGPT in this case is a subjective matter. It is best to use ChatGPT-generated responses sparingly, and consult with professors for the course if there is any doubt as to what is allowed.

6 Hardware and Design

This chapter contains our design methods and results for the hardware elements of the Electrathon vehicle. We first discuss the design and configuration of the battery pack because it is the power source and origin point of all circuits in the vehicle. The motor controller is discussed next because it is the highest power onboard device, followed by the auxiliary converters that power the logic circuitry. We last discuss the AC-DC power supply that will rapidly charge the battery, which has by far the most complex design and over 200 components.

6.1 Battery Pack Design

This section covers the design for the car's battery pack, including the cell layout (6.1.1) and battery management system (6.1.2).

6.1.1 Cell Layout

In the Battery Selection section we outline why the Samsung 20S is the best battery cell to use in an Electrathon car. To design the power conversion circuits in this project we must first determine the voltage range of the battery pack that they will send or receive power from. The Battery Management System (BMS) will disconnect the battery if its voltage goes outside of the specified limits. The Samsung 20S datasheet specifies an end of discharge cut off voltage of 2.5V and a rated charge voltage of 4.2V. These are widely used values, but many sources recommend charging and discharging over a narrower range to minimize degradation and maximize lifetime. When using older and repurposed cells, ranges of 3V to 4V or even 3.5V to 4V are common [113]. To be competitive in Electrathon, we must approach the voltage limits as closely as possible, charging the battery to precisely 100% State of Charge (SoC) and discharging fully to 0%. Smartphones use a Li-Ion variation that can withstand higher voltages of 4.35V at the cost of lower lifetime, but other than these batteries Li-Ion should not be charged beyond 4.2V [114]. The batteries are rated to withstand charging beyond that level, but lifetime is reduced so greatly that it is not worthwhile, even for Electrathon. Charging slightly beyond 4.2V degrades the batteries but is not a safety issue, so 4.2V per cell is the commonly used charging cutoff voltage.

The question of when to terminate Li-Ion discharge is more complex. When Li-Ion batteries are allowed to reach low voltages, harmful and irreversible side reactions become thermodynamically favorable. In extreme cases, a battery that has been exposed to very low voltages of 2V or below will begin rapidly heating when it is recharged, and many E-bike battery fires have been caused by battery cells that were discharged too low and recharged by overriding the protection features of the BMS and charger [115]. In cases of less severe damage, battery capacity is decreased and the self discharge current is increased. Sources vary widely on the voltage threshold where these effects begin, and it is a topic of intense debate in the Do-It-Yourself (DIY) battery community. Some insist that Li-Ion cells become permanently damaged below 2.7V and so discharge should terminate at 3V, while others claim that cells below 2V can be safely recharged if the current is low [116] [117]. A complicating factor is that the internal resistance of the battery causes the terminal voltage to change independent of the battery's state of charge, and "internal resistance" is a combination of inductive and resistive impedance in the electrodes and charge transfer resistance with the electrolyte. The latter depends on ion concentrations, so a battery may be at one voltage under load, a slightly higher voltage just after removing the load, and recover to an even higher voltage after sitting for a few minutes while the concentrations equalize. Batteries near 0% SoC have much higher internal resistance than batteries near 100% SoC, so a battery that floats at 3V with no load may drop to 2.5V under a high load. E-Bikes and scooters cut off the battery when the voltage reaches 2.5V, even briefly, and this is acceptable because there is little energy between 3V and 2.5V open circuit. In the context of the power converter voltage ranges, we will assume a cell voltage between 2.5V and 4.2V. The average discharge voltage of Li-Ion is reported as either 3.6V or 3.7V, and those voltages apply to Lithium-Cobalt and Lithium-Manganese batteries respectively [118]. Li-Mn is more common, so 3.7V is more widely known.

High performance Li-Ion batteries are widely available to consumers in the 18650 and 21700 form factors, and we selected the former because it is an older standard, so there are more models to choose from. The names refer to the dimensions of the cylindrical battery cells, which are 18mm in diameter and 65mm long or 21mm in diameter and 70mm long respectively. These 18650 cells were originally used in laptop batteries, but they found a much wider market when Tesla connected 6,831 cells to form the battery pack in its first gen Roadster. The industry has moved towards larger cells, but even today the battery pack of the Model S is composed of over seven thousand 18650 batteries. The size is widely used in personal electronic and micro mobility applications such as flashlights, scooters, USB power banks, electric skateboards, and E-bikes, so it is a good choice for a small electric vehicle such as an Electrathon car.

When designing a Li-Ion battery pack for an Electrathon car, the main consideration is maximizing usable energy while keeping the weight of the cells below 15 lbs. 20S battery cells have a rated capacity of 2000mAh at 3.7V average discharge voltage, and the total rated energy of the pack is not allowed to exceed 1000 Wh, so the number of cells is limited to 1000/(2*3.7)= 135. These have a combined energy of 135*2*3.7= 999 Wh, which is ideal. 18650 cells weigh 45 to 49 grams, and a 135 cell pack is within the 15 lb weight limit at 135*0.049*2.2= 14.55 lbs. To minimize the number of separate voltage levels, cells are combined into a pack in a parallel-series configuration. They are first connected into parallel blocks, then those blocks are connected in series. The opposite approach of connecting cells into series strings and then putting those strings in parallel is not recommended because it creates more discrete voltage levels that must be monitored. Given that we must use 135 cells, the possible combinations of series and parallel groups are the integer factors of 135. These are 3, 5, 9, 15, 27, and 45. The number of cells in series is multiplied by 3.7V to find the average pack voltage, or it is multiplied by 4.2V for the maximum voltage. The motors that we identified in Motor Selection are intended for 12S battery packs, but since that voltage is not available, the next closest is 15S. It is acceptable to operate a motor controller at a higher voltage than the motor because the controller and motor inductance effectively form a buck converter.

Raising the voltage does not raise the magnetic saturation or RPM limits, so the motor should not be operated at full throttle at the higher voltage. Current is what determines motor heating and torque production, so the driver will monitor that value. The overall configuration is 15S x 9P for 55.5V average, 63V peak, and 18Ah capacity.

6.1.2 Battery Management System (BMS)

The BMS in a Li-Ion battery provides two related but distinct functions: Balancing and protection. When battery cells are connected in series the current that flows through them is the same, but each level of a pack will have slightly different capacity, internal resistance, and self discharge. These factors cause different levels of the pack to have different voltages, and if nothing is done it's possible for one level of the pack to have a voltage that is far too high or far too low while the overall voltage is normal. At a minimum, a BMS circuit measures the voltage of each level of the pack and terminates the charge or discharge if the voltage becomes too high or too low. When the process is terminated in this way, the effective capacity of the entire battery is bottlenecked by one imbalanced level. To improve performance, nearly all BMS circuits also include balancing, which is the charging or discharging of one level of the pack to bring its voltage closer to the average. Passive balancing passes current through a resistor to turn energy into heat, and active balancing recovers energy by moving it from one cell to another. Passive balancing is much more common because it is simpler and if cells are well balanced the energy loss is minor, so this is what we will use.

Commonly available BMS boards use a modular system of repeating subunits where the number of subunits equals the number of cells in series. For each cell, a DW01A IC monitors the cell voltage and is powered by it, triggering a communication bus when the voltage goes under 2.4V or over 4.3V [119]. The 0.1V of additional margin is to avoid spurious cutoff when the pack is being charged and discharged to 4.2V and 2.5V overall. Also for each cell, an HY2213 IC monitors and is powered by the cell voltage, and if it reaches 4.2V it pulls high the gate of an N-channel FET that is connected in parallel with the battery through a resistor [120]. This slowly discharges the cells that are a bit too high during the final constant voltage stage of the charge cycle. These operational characteristics are acceptable for our application, so we will use an off the shelf 15S BMS board. In E-bike applications all electronics are connected through the BMS to prevent the battery from being drained too low, but in Electrathon we cannot risk the motor cutting out in the middle of the race. Connecting the motor through the BMS would add resistive losses, and the highest rated 15S BMS is only 50A, which is not enough for the full power of the motor. Thus, the BMS will protect the battery while it is charging, and during a race the driver will reduce throttle input as the pack voltage approaches 2.5V per cell. It may be possible to design a “limp mode” that ensures the car can finish the race while protecting the battery.

6.2 Motor Controller Design

In this section, the motor controller design is explained. First, in 6.2.1, we describe how current sensing is used in the design. Next, we move to the overall layout of the motor controller in 6.2.2. 6.2.3 and 6.2.4 discuss the gate drivers and 5.5V regulator and 1.2 and 3.3V regulators, respectively.

6.2.1 Current Sensing

As explained in the Motor Comparison section, Brushless Direct Current (BLDC) motors differ from brushed motors because they do not have a commuter. Electronic switches in the motor controller must move the phase wires between VCC and ground as the rotor rotates to each new position. To change voltages at the optimal time, the motor must know the rotor's position, and in some motors this is accomplished with hall effect position sensors. The most efficient motor available did not have hall sensors, so the motor controller must use voltage and current measurements to estimate rotor position. On startup this is done by applying a small voltage pulse to each phase and measuring the current, which gives the inductance. The inductances of the phases will be different due to the orientation of the permanent magnets on the rotor, so from this information the rotor position and optimal starting point can be determined. To improve efficiency, we will use a Field Oriented Control (FOC) algorithm, and sensorless FOC is implemented with a sliding mode observer that uses a digital twin of the motor to estimate position. The known values are the voltage that is applied to each phase, the current that flows in each phase, and the inductance and resistance of the motor windings. In a permanent magnet motor the rotor is locked to the flux generated by its magnets, and the motion of this flux over the windings is what creates back-EMF. Thus, the controller uses motor parameters to estimate the fraction of the applied voltage that is canceling the back-EMF, and it integrates the back-EMF to find the flux according to Faraday's Law. Finally, the Flux Estimator block transforms the fluxes from the α and β cartesian reference frame to a polar reference frame that is referenced to the angle θ that was used for the earlier transformation to $\alpha\beta$. This is relevant to the hardware design of the motor controller because it means that unlike in some trapezoidal control implementations, we do not need to measure the phase voltages relative to a virtual grounding point, nor do we need comparators.

The four fundamental components of a motor controller are the half bridges, the DC bus capacitors, the voltage sensing circuit, and the current sensing circuit. Existing designs were examined for inspiration, and the voltage and current sensing circuits are similar to those used in the VESC 6.4 [122]. Phase voltages are measured with voltage dividers to ground and the ADC pins of the microcontroller, and phase currents are measured inline with the phase wires with the AD8417 differential shunt monitor IC. It amplifies the small voltage across a sense resistor by a fixed ratio of 60:1, and it has a much higher Common Mode Rejection Ratio (CMRR) than a discrete op amp implementation because the matching between resistors is better. It is powered by the same 3.3V level as the I/O on the microcontroller, and it can withstand up to 80V common mode, which is important because the sense resistor inline with the output will swing between ground and the 60V battery rail. It can measure current bidirectionally with 0A at 1.65V, which is conveniently in the middle of the ADC input range for a 3.3V microcontroller. Due to the capabilities of this device, the only other signal processing needed between each sense resistor and ADC input is a low pass filter formed with a 100 ohm resistor and 4.7nF capacitor. The schematic is shown in Figure 6.1 below.

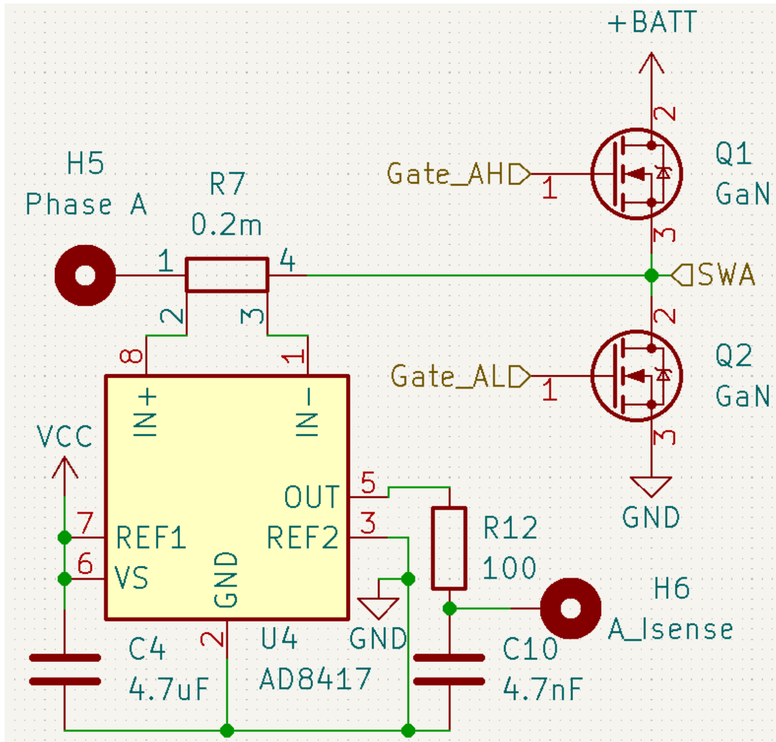


Figure 6.1 - Inline Current Sensing Schematic

6.2.2 Overall Layout

A "bridge" circuit is a configuration of four switches or diodes between two voltage rails with two parallel groups, each having two devices in series. An element that is connected between these devices can either have a positive or negative voltage applied to it up to the maximum difference between the voltage rails. This general concept is used in full bridge rectifiers (the most widely known), full bridge switching converters, Class D audio amplifiers, brushed motor controllers, and many other applications. Brushless motors are three terminal devices that need either a positive or negative voltage to be applied to each terminal, so they use a related circuit called a half bridge. A half bridge is two switches in parallel with the output connected to the node in between them, so it is half of a full bridge. The heart of a brushless motor controller is a triple half bridge circuit that can pull each motor phase to ground or VCC. The motor windings have a high inductance that acts like the output inductance of a buck converter, ensuring that the phase current is approximately constant as the half bridge switches pull the winding high and low in an alternating pattern. The average current is controlled by the Pulse Width Modulated (PWM) duty cycle in the same way as a buck converter. When the motor current is freewheeling no current is drawn from the input, so the input current alternates between the phase current and zero at a high frequency. To stabilize the bus voltage and prevent ripple current from damaging the power source and causing EMI, motor controllers also have DC link capacitors that must withstand this high ripple current. To minimize power loss, we selected two 260uF capacitors that keep voltage ripple to only $10V_{Pk-Pk}$ when 50A are drawn from the 55.5V bus.

6.2.3 Gate Drivers and 5.5V Regulator

The six power MOSFETs that form the triple half bridge must be turned on and off rapidly, and this is the job of the gate drivers. Silicon MOSFET gates can withstand +/-20V, and at 10V the switch is fully on such that the R_{DS-On} is nearly the same as when the gate is at 20V. This provides a lot of headroom, and the standard drive voltage is either 12V or 15V. We selected GaN FETs due to their lack of reverse recovery and consequently lower switching losses, and they require a different gate voltage. The maximum allowable V_{GS} on a GaN FET is only 6V, and at lower voltages they are not fully on, so the standard drive voltage is 5.5V, which is a mere 0.5V of headroom. To reduce the chances of damaging the gate, we selected Infineon 1EDN7116 drivers that are specifically designed for GaN, and we supplied them with 5.5V from a dedicated 15V to 5.5V regulator. Trying to go directly from the 55.5V battery pack to 5.5V would be an excessively high step down ratio for a buck converter, requiring a very low duty cycle and probably not achieving good regulation. It would also rule out using an integrated IC. Instead, the gate drivers and microcontroller are powered from an external 15V regulator. The 3.3V and 1.2V rails that power the microcontroller are not derived from the 5.5V source because this would increase the number of switches for the current to flow through and increase losses.

Half bridge drivers are FET driver ICs that control both the high and low FETs in a half bridge. Connecting both FETs to one driver allows the driver to have hardware protections against shoot through, the condition where both the high and low side FETs are on simultaneously. Shoot through is very bad because the FET channels act as a short circuit between the low impedance power rail and ground, discharging the energy of the DC bus capacitors into the FETs and likely destroying them. It is prevented by inserting deadtime, a period where both gate drives are pulled low, and discrete hardware circuits that add deadtime are very bulky and involve multiple inverters and schmitt triggers. Another complication of the half bridge configuration is that if the high side switch is an N-channel FET, and it should be to minimize R_{DS-On}, the gate must be charged 5-20V above the input power rail. More specifically, V_{GS} = 5-20V, and the source of the high side FET is connected to the switch node, so it varies between nearly VIN and nearly GND. The common solution in low and medium power applications is a bootstrap circuit where a capacitor and diode are connected between the switch node and a 5-12V power supply as a peak detector. When the switch node is low, a low voltage charges the capacitor, and when the node is high, the diode holds the capacitor voltage above everything else in the circuit. The only alternative to a bootstrap circuit is an isolated power supply that can tolerate high common mode voltages and rapid dv/dt swings, and each half bridge requires a separate secondary. These power supplies are expensive, bulky, and time consuming to design, so they are usually only used with very high power switches such as those in a 100+ kW inverter.

The clever bootstrap circuit has one problem when used with GaN. Due to the inductance of the output and the voltage drop across the low side switch, the switch node voltage is pulled slightly below ground as the current commutes from the high to low side FETs. Since this voltage transient is limited to the switch node and does not propagate to the power supply, the output voltage remains the same, and the voltage across the bootstrap capacitor increases. The transient is clamped by the reverse conduction of the low side FET before it turns on and clamps the voltage further, and as explained in Transistor Part Selection, GaN FETs do not have a body diode that guarantees a reverse conduction drop of less than 1V. Their reverse conduction mechanism results in a drop of several volts, and while silicon FET gates have 8V of headroom, but GaN FET gates only have 0.5V headroom, so the overcharged bootstrap capacitor can destroy the high side switch.

Much as this design would not be possible without the recently released GAN3R2-100C FET from Nexperia, it also would not be possible without the 1EDN7116 GaN gate driver from Infineon. The 1EDN7116 is not a half bridge GaN driver because those are not available, but it has features that make it easier to use in a half bridge. It has an active bootstrap clamp feature that prevents bootstrap capacitor overcharging by connecting the diode to a dedicated pin on the IC. This pin is high impedance when the driver output is off, preventing overcharging during reverse conduction, and it connects to VDD when the output is on. Although our 10kHz switch frequency is modest, GaN FETs are intended to be used at much higher switching frequencies with much faster transitions, so it is important to eliminate parasitic inductance from the gate drive path. To prevent dv/dt turnon due to Miller capacitance, FET gates must be pulled low with more current than they are pulled high. This is often done with a diode and a gate resistor in parallel, but the 1EDN7116 has separate source and sink outputs and is available with different drive impedances, so both the diode and gate resistor can be eliminated, reducing parasitic inductance. The 2A/5A version is a good fit for our FET's 12nC of gate capacitance, and in case of ringing, the first version of the PCB includes pads for mounting a gate resistor. These can be removed in a later version if they are not needed. Another challenge of using separate high and low drivers is that the ground referenced signal must be passed to the high side driver, and normally this is done with an optocoupler, but those add an unpredictable delay that makes the dead-time hard to synchronize. Due to its high reverse conduction loss, GaN benefits from very short deadtimes, and the 1EDN7116 implements the clever solution of a differential input with +/-200V common mode immunity. As long as the input pins are separated by 47k resistors, the driver can swing up to the VIN rail while the input signal remains at ground, and the amplifier inside the driver will only read the small difference in voltage. This allows the high and low side inputs to be cross coupled and driven together, and the differing drive strengths should provide enough deadtime given how fast GaN is. This will be tested at a low voltage. Figure 6.2 below shows the driver circuit for one half bridge where SWA is the switch node.

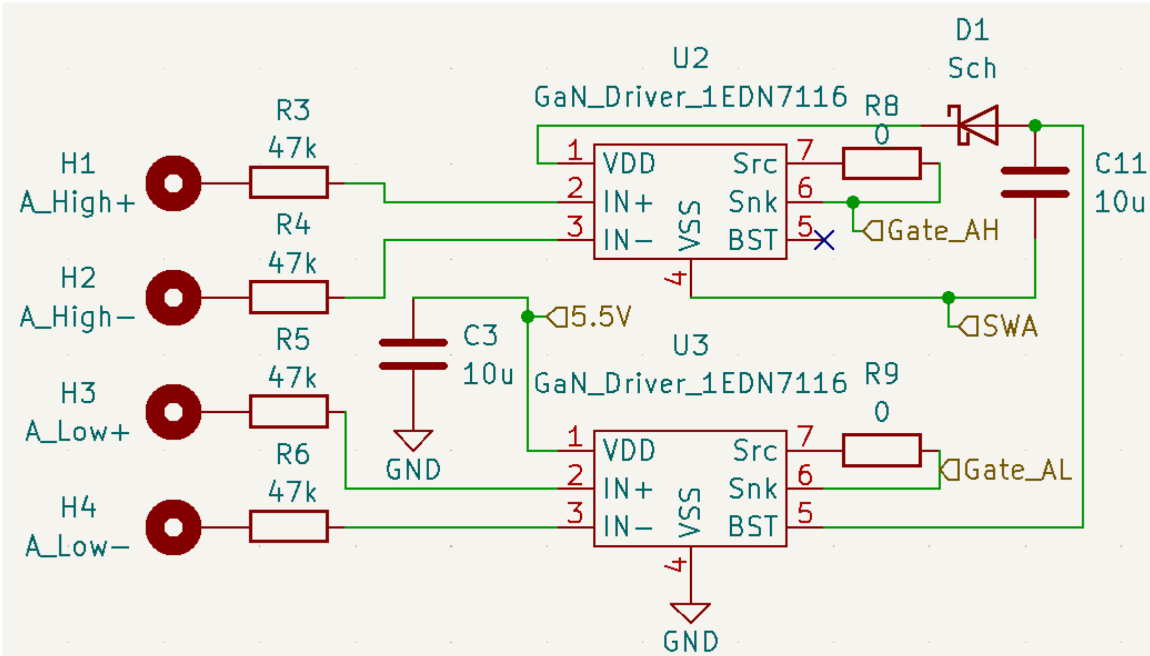


Figure 6.2 - Drivers for First Half Bridge in Motor Controller

The V_{DD} and bootstrap capacitors were selected based on calculations from the datasheet. The known inputs are V_{DD} = 5.5V, V_F = 0.3V, Q_G = 12nC (worst case), I_{VDD} = 1.8mA, $I_{D\text{-}Leakage}$ = 500nA, D_{Max} = 0.4, f_{SW} = 10k, V_{HBR} (UVLO threshold) = 3.85V, and $V_{Hysteresis}$ = 0.1V. The acceptable voltage ripple on the bootstrap capacitor is 1.25V, so the minimum required capacitances are C_{Boot} = 154nF and C_{VDD} = 163nF. Using a larger capacitance increases robustness and does not impact size, weight, or cost, so we specified 10uF 16V capacitors.

$$\Delta V_{BOOT,max} = V_{DD} - V_F - V_{BOOT,min}$$

ΔV_{Boot} = 5.5 - 0.3 - 3.95 = 1.25V

$$V_{BOOT,min} \geq V_{HBR} + V_{HBH}$$

V_{HBR} = High-side driver UVLO rising threshold
V_{HBH} = High-side driver UVLO threshold hysteresis

$$Q_{BOOT} \approx Q_G + \frac{I_{Vdd} + (I_{diode} \times D_{max})}{f_{sw}}$$

$$C_{BOOT} \gg \frac{Q_{BOOT}}{\Delta V_{BOOT,max}}$$

$$C_{Vdd} \gg \frac{Q_G + Q_{BOOT}}{\Delta V_{DD,max}}$$

- Q_{Boot} = 12*10^-9 + (0.0018+(0.5*10^-6*0.4))/10000 = 192nC.
- C_{Boot} = 192*10^-9/1.25 = 154nF

- C_{VDD} = (12+192)/1.25 = 163nF

To maximize the efficiency of the motor controller at low loads, it is important to reduce quiescent power consumption. Each driver has a maximum possible current consumption of 2.6mA, and the likely consumption is 12*10^-9*10000= 0.12 mA per driver, so the load current on the 15V to 5.5V regulator is tiny. After comparing the TPS62736, LMR43610, and LMQ61460 voltage regulators, we settled on the latter because it has the best efficiency at light load, 91% at 1mA and 77% at 0.1mA. Besides extremely low standby power consumption, it uses frequency foldback to reduce switching losses. For 400 kHz switching frequency and 5V output, the datasheet recommends a 4.7uH inductor and two 47uF output capacitors, so these were used along with a 10uF input capacitor. The internal voltage reference and thus regulation threshold is 0.45V, so a 5.5V output corresponds to R_{FBT} = 453k and R_{FBB} = 100k. The only downside is the package, which is so tiny that it approaches the limits of what our preferred PCB fabrication house can manufacture. JLCPCB has a minimum pad to pad clearance of 0.127mm, and this IC footprint has pads only 0.25mm wide with 0.25mm clearance between them. This can *probably* be soldered with solder paste and a hot air gun, and if not, the second version of the motor controller will use a different regulator. The circuit is shown in Figure 6.3.

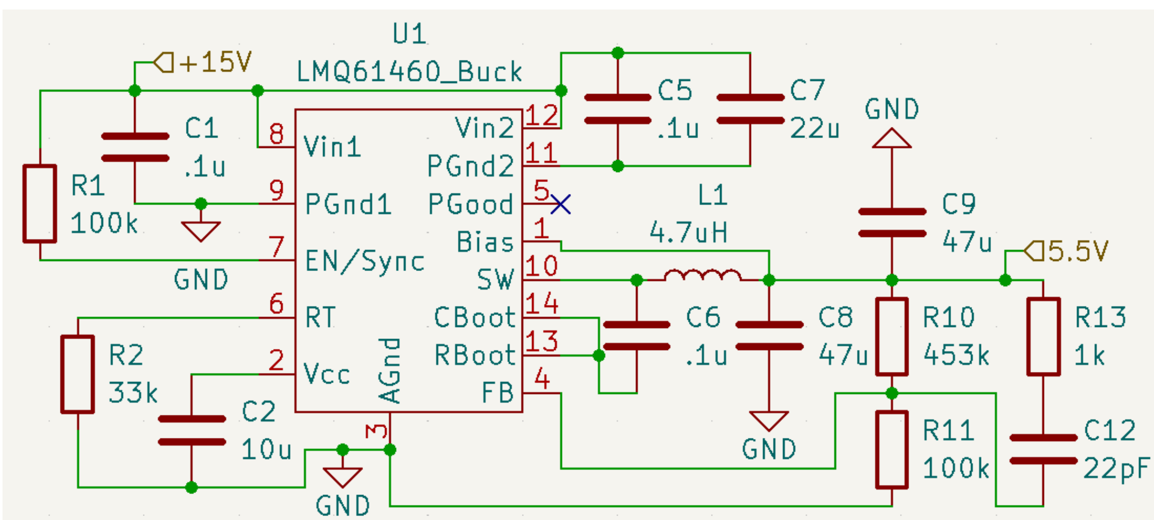


Figure 6.3 - 15V to 5.5V Regulator Schematic

6.2.4 3.3V and 1.2V Regulators

The F280049C microcontroller that governs the motor controller operates on two voltages. The I/O and ADC operate on the 3.3V level for greater compatibility, but the core operates on 1.2V to reduce power consumption. It must operate on 1.2V, but the microcontroller can run on a single supply because it has a built-in 3.3V to 1.2V regulator. VDDA pins power the analog circuitry and must be decoupled with at least a 2.2uF capacitor, so we use 10uF, VDDIO pins power the I/O and must be decoupled with at least a 0.1uF capacitor, so we use 1uF, and VDD pins power the core at 1.2V, so they need 20uF capacitors, and we use 22uF. Using an external 1.2V supply, the maximum current consumption on the 1.2V pin is 90mA, which scales to 90*1.2/3.3= 33mA at 3.3V at 100% efficiency, plus 75mA on the 3.3V pins for 108mA in total. Using the internal 3.3V to 1.2V regulator the 3.3V input current is 143mA under normal operation with no power saving modes engaged. For simplicity, we will use the built-in regulator.

The microcontroller is most commonly available in the VQFN-56 package, which has 14 pins on each side. Using the pin numbers of that package, pin 49 is the switch node that pulses between 3.3V and 0V according to the FETs in that small half bridge. The datasheet recommends a 2.2uH inductor between the switch node and a 20uF output capacitor, and we use 3.3uH and 22uF due to availability. Pin 51 provides output voltage feedback, and it must be connected externally. Pin 50 is the power ground, and pin 48 is the regulator input supply, so it requires at least a 20uF capacitor. This is essentially a buck converter where all parts except the capacitors and inductor are inside the microcontroller.

For the 15V to 3.3V regulator, TI Webench Power Designer was used to quickly produce a design. The 3.3V microcontroller current was rounded up to 200mA, and the other parameters were V_{IN} from 10V to 20V and a bias towards "high efficiency" designs. Most of the resulting designs used regulators that were out of stock, and as noted in the last section, high efficiency regulators tend to be very small. We selected the TPS54336 because it comes in an easy to solder SO-8 package with a positively gigantic 1.27mm pin pitch, shown in Figure 6.4 below.

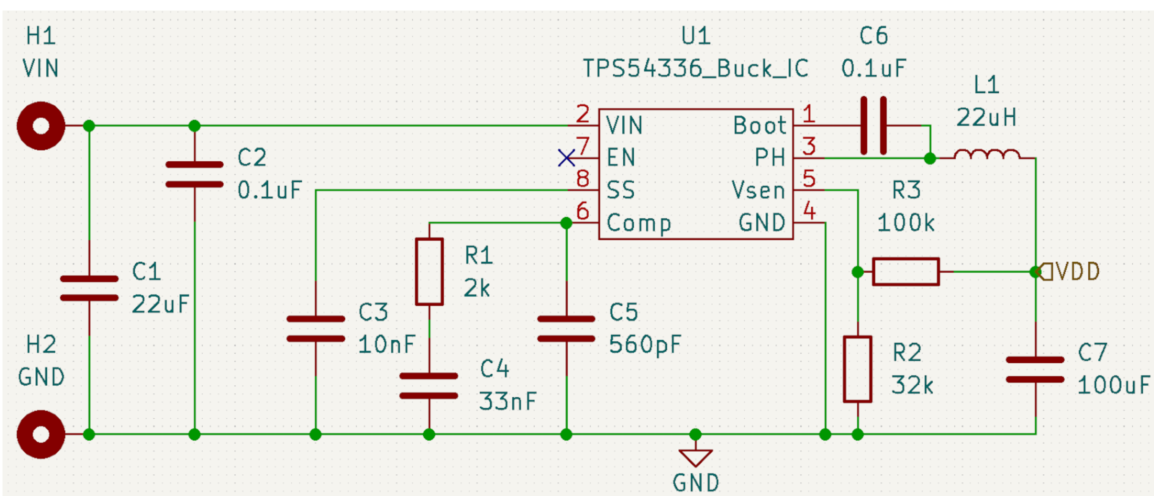


Figure 6.4 - 15V to 3.3V Regulator Schematic

The complete schematic for the motor controller is shown in Figure 6.5 below.

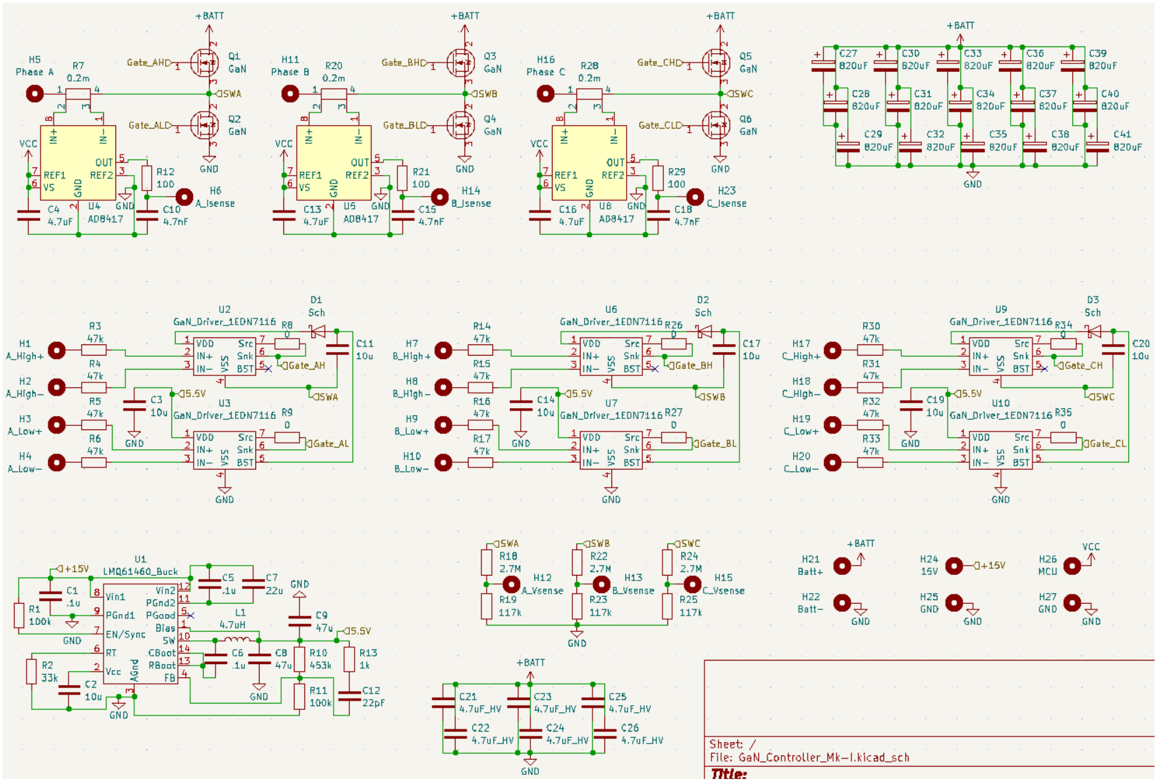


Figure 6.5 - Motor Controller Schematic

6.3 Auxiliary Converters

This section covers the design for the auxiliary converters: the +15V regulator is explained in 6.3.1, and the -15V regulator is described in 6.3.2.

6.3.1 +15V Regulator

The central microcontroller can either measure the current flowing to the motor controller through a resistive current shunt or a hall effect sensor. Current shunts are common because they are cheap and simple to implement, but hall effect sensors have higher efficiency and better noise immunity, so we will use that instead. Nearly all high power hall sensors require bipolar 15V and -15V power supplies, so the main purpose of the 15V and -15V regulators is to generate those rails. The 15V regulator also powers all other low voltage electronics in the car. Both converters use the LM5116 synchronous buck controller and were designed with TI Webench Power Designer. For the 15V regulator the given specifications were $V_{IN\text{-}Min}$ = 2.5*15= 37.5V, $V_{IN\text{-}Max}$ = 4.2*15 = 63V, V_{Out} = 15V, and I_{Out} = 2A, and the schematic is shown in Figure 6.6 below.

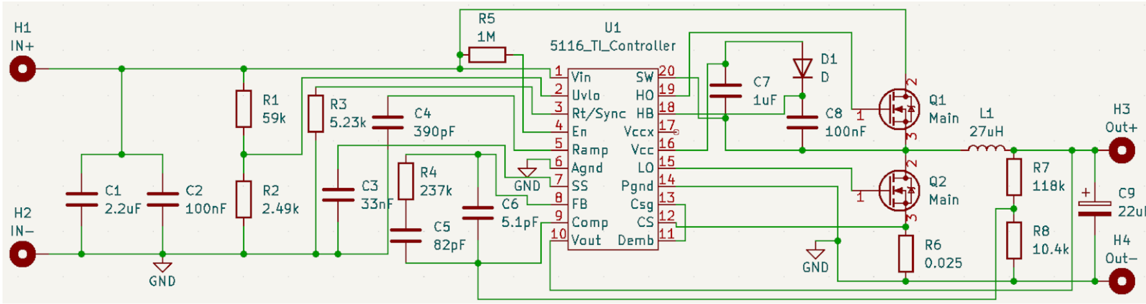


Figure 6.6 - Battery pack to 15V Regulator

6.3.2 -15V Regulator

The -15V regulator uses the LM5116 synchronous controller and the inverting buck boost topology. The given specifications were V_{IN} from 37.5V to 63V and V_{Out} = 15V at 0.5A because the low power hall effect sensor will be the only device connected to the -15V rail. The schematic is shown in Figure 6.7 below.

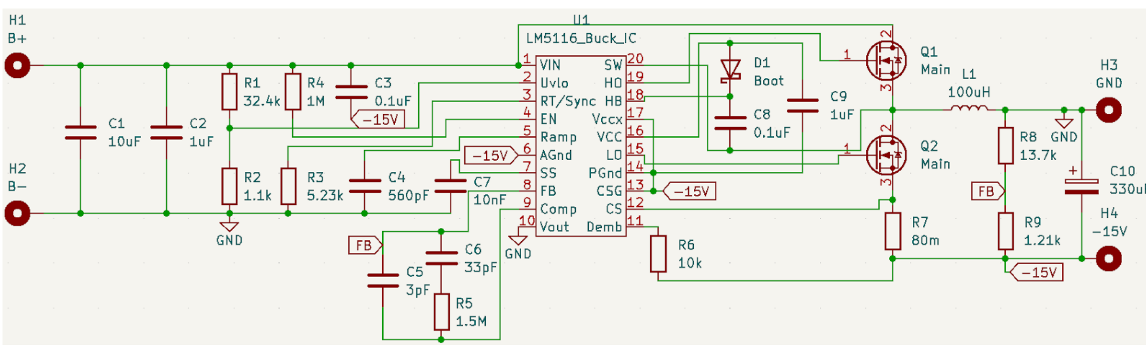


Figure 6.7 - Battery pack to -15V Regulator

6.4 AC-DC Battery Charger Design

Lithium ion batteries are charged according to the Constant Current (CC) Constant Voltage (CV) method known as CC/CV [122]. A good way to understand this is that Li-Ion batteries are damaged by overvoltage and overcurrent, but otherwise they don't have any restrictions, so a CC/CV profile maximizes the charging speed. The voltage limit is due to side reactions, and the current limit is due to intercalation and thermals. When a Li-Ion battery charges the lithium ions must intercalate into the graphite electrode, and if it's charged rapidly a higher percentage of those ions will deposit onto the surface of the electrode instead of going inside. These metallic lithium deposits degrade battery capacity and can possibly cause short circuits. There is also a thermal limitation on charging current since batteries have a series resistance, charging current causes heating, and batteries are strictly limited to 50C with longer lifetimes at lower temperatures.

Most power supplies operate in Constant Voltage mode only, and while they may have overcurrent protection, it is not designed to run continuously. If a power supply is connected directly to a battery, there will probably be a surge of current and the power supply will trip offline. Battery chargers are just smart CC/CV regulators that set their limits according to the battery that is connected. The Samsung 20S cell has a low internal resistance, so unlike other batteries, it is rated to charge at 2C continuously. For a 2000mAh cell, 2C is 4A, and we have 9 cells in parallel for 36A in total. The other perspective is that we have an 18Ah battery pack, and 18*2= 36A. Regardless, our custom charger has the specifications of $V_{Out\text{-}Max}$ = 63V, $I_{Out\text{-}Max}$ = 36A, V_{IN} = 120V, $I_{IN\text{-}Max}$ = 15A, and $P_{IN\text{-}Max}$ = 120*15 = 1800W. When the battery voltage is low the charger operates in CC mode, and once the terminal voltage rises to 63V it operates in CV mode. The battery reaches approximately 85% SoC at the transition point, and it can get there in only half an hour. We have two hours between races, and although the car will not get to the charger immediately, this should leave plenty of time for the last 15% to trickle in.

Switch Mode Power Supply (SMPS) regulators either use the error amplifier and voltage reference that are built into the controller IC, or they use an external error amplifier and reference. The error amplifier compares the scaled output voltage or current signal to the voltage reference and drives the PWM duty cycle higher or lower to reduce the error. For safety, off-line power supplies must be isolated as discussed in the Standards section, but this creates a problem with regulation. Negative feedback must be employed to keep the output voltage and/or current at the desired level, and it is very expensive and difficult to build an accurate isolated sensor that can pass an analog voltage value across the isolation barrier. Non-isolated power supplies do not have this problem, they simply connect a voltage divider across the output and connect it back to the inverting input of the error amplifier as shown in Figures 6.4, 6.6, and 6.7. Isolated power supplies must either use an external error amplifier and pass the error signal back through an optocoupler, or they must place the controller IC on the secondary side so there is no isolation barrier with the output. In this power supply we chose the latter solution because the duty cycle allowed us to use gate drive transformers, but in hindsight an optocoupler based design would have lowered the BOM cost by a few dollars.

To achieve CC/CV regulation, we used the circuit shown in Figure 6.8 below to apply either a voltage or current feedback signal depending on which was greater. The TL082 dual op amp can be powered by up to 40V, so 12V is acceptable and it doesn't need a lower voltage regulator, and it has a low offset voltage of only 1mV, which is better than the 7mV of the LM351 that is more commonly used in power supply regulation circuits. A low offset voltage is important for measuring the low voltage that is produced across the 1 milliohm sense resistor, but that is necessary to minimize power dissipation because 36^2*0.001= 1.3W. The circuit is simply a differential amplifier to multiply the voltage across the sense resistor up to 36*0.001*69.8= 2.51V at 36A, combined with an ideal diode connected to a voltage divider on the output. Either limit can pull the output high, regardless of whether the other limit is pulling low, and higher voltage on the FB pin reduces the duty cycle and output power.

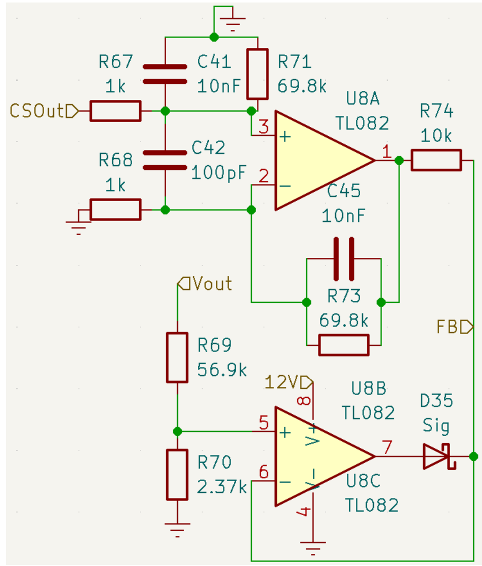


Figure 6.8 - CC/CV Feedback Circuit

6.4.1 Electromagnetic Interference (EMI) Filter

Starting from the input terminals, the first component is a 15A inline fuse in a 20x5mm holder, which is connected in series with the live terminal. This is followed by a 14mm Thermally Protected Metal Oxide Varistor (TMOV) that shorts and clamps the rail if there is a voltage spike, blowing the fuse. If there is a prolonged moderate overvoltage that does not increase the current enough to blow the fuse, the built in thermistor will prevent the TMOV from burning up. Another protection feature is that all the capacitors that connect to ground in the common mode filter are X2 safety capacitors, which means that they are much more likely to fail open than fail short. If a line-ground cap failed short, the fault current should blow the fuse, but that would only occur if there was a robust ground connection. If the ground was weak or disconnected, the output terminal and metal chassis would float up to 120V, potentially shocking the user. Live-to-ground faults should be avoided, and X2 capacitors help with that.

The job of the EMI filter is to reduce differential mode and common mode noise to the levels specified in EN55032 from 150 kHz to 30 MHz. The signals in power converters are mostly square waves at the switching frequencies, which in this case are 205kHz and 100kHz. Square waves carry most of their energy in the fundamental and first few harmonics, but hard switching converters, of which this is one, have much higher frequency oscillations after each switching event. Even a very detailed simulation would not capture the parasitic elements between components, so it is impossible to know what the EMI emissions of a circuit will be until it is built. The standard approach to EMI filter design is to first build the circuit, then measure the EMI, then calculate the corner frequency that will attenuate the EMI to the limit of 60 dBμV at 150 kHz. We want to get this right on the first design cycle, we will over specify the filter a bit to maximize the chances that it passes testing.

The standard design for an EMI filter is a common mode choke combined with capacitors to form a common mode LC filter and a differential mode PI filter [123]. Common mode chokes are two inductors coupled to the same core such that when a differential mode current flows, such as 15A in one direction and 15A in the other, the fluxes created by the two coils nearly cancel out. Very little energy is stored in the core, so it can be made quite small while the wires are large to pass a large current. When a common mode current tries to pass, it is “choked off” by the high inductance created by the high number of turns since the fluxes now add instead of canceling. There is still a small leakage inductance that applies in both cases. LC filters have two elements and a rolloff of -40 dB/decade, and PI filters have three elements and a rolloff of -60 dB/decade. An LC filter is sufficient for blocking common mode noise because the common mode inductance of the choke is very high, but a PI filter is required for differential mode because the inductance is lower.

A rolloff of -60 dB/decade means that for every 10x increase in frequency the attenuation increases by 10^(60/20)= 1000x, so a cutoff frequency of 75kHz would attenuate by a factor of (150/75)*1000/10= 200 at 150kHz [124]. The cheapest 20A common mode choke is the PG1265NL, which has 380µH common mode and a minimum of 1.2µH leakage inductance. Setting the differential mode cutoff frequency at 75kHz or lower requires a capacitor of 0.012uF or greater, which is easily met with a 0.1uF capacitor of the same cost. If the common mode cutoff frequency is the same, the required capacitor is much larger but still acceptable. The capacitor sizes can be modified according to the test results. The circuit design is shown in Figure 6.9 below.

- $f = 1/(2\pi*(LC)^{\wedge}0.5)$
- $C_{Differential} = 1/(L_{Diff}*(2\pi*f)^{\wedge}2) = 1/(380*10^{\wedge}-6*(2*3.14*75000)^{\wedge}2) = 0.012uF$
- $C_{Common} = 1/(L_{Com}*(2\pi*f)^{\wedge}2) = 1/(1.2*10^{\wedge}-6*(2*3.14*75000)^{\wedge}2) = 3.76uF$

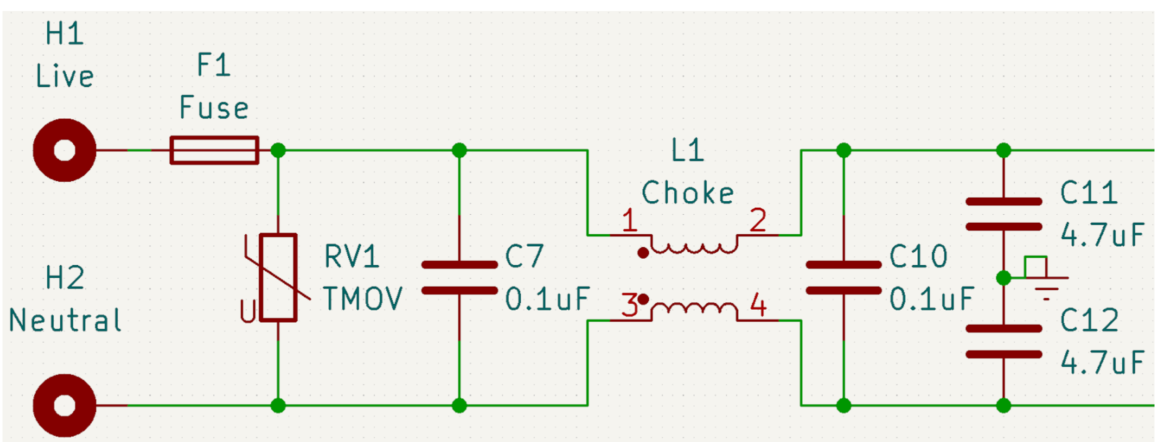


Figure 6.9 - EMI Filter Schematic

6.4.2 Power Factor Correction (PFC)

Low power or outdated power supplies simply connect a bridge rectifier and a capacitor to convert AC to DC. The problem with this approach is that the current drawn from the AC source is highly distorted, and this manifests as high frequency harmonics that are emitted from the AC-DC converter into the power grid. These harmonics increase heating in transformers and are undesirable. IEC 61000-3-2 sets limits on the maximum harmonic current, and these limits are best met by drawing a current that is as close to sinusoidal and perfectly in phase as possible. This can be accomplished with a Power Factor Correction (PFC) that is essentially a high frequency boost converter. A bridge rectifier folds the AC waveform so it has purely positive values, and from the perspective of the boost converter, the input voltage changes very slowly from zero to 120*2^0.5= 169V. The 205 kHz switching frequency of our boost converter design is so much higher than the 120 Hz input ripple that the input voltage is essentially constant. Dedicated controller ICs exist for the PFC application, and we selected the UCC28070 because it supports two interleaved phases.

Universal input power supplies must tolerate an AC RMS input from 85V to 265V so they can be used anywhere in the world. Our power supply will only be powered from 120V, but if we ever wanted to convert it to a universal supply, it would be helpful if the DC bus was above 265*2^0.5= 375V. There is an unavoidable 120 Hz power ripple on the DC bus between the PFC stage and the isolated power converter, and while it is smoothed with capacitors, it takes a lot of capacitance to minimize voltage ripple at 15A and only 120 Hz. Electrolytic capacitors are the only practical option, but they do not tolerate ripple current well as explained in Capacitor Selection. We chose to use a 400V DC bus to support future expansion and reduce the ripple current to approximately 1800/400= 4.5A, half the ripple that a 200V bus would have. To distribute the ripple current we selected 10 x 100uF capacitors, each rated to 0.68A ripple when they heat themselves to 85C. The combined 1000uF creates a voltage ripple of 37.5$V_{Pk\text{-}Pk}$, which is acceptable.

- R_{Load} = 400/4.5= 88.9 ohms
- V_{Ripple} = V_{Peak} / (f*R*C) = 400/(120*88.9*1000*10^-6) = 37.5V
- 400-37.5= 362.5V min, 437.5V max

In a power converter, interleaving two phases with a 180 phase shift between the gate signals is beneficial because the high frequency ripple current partially cancels and smaller inductors can be used. At $6 each, the inductors are still expensive, but together they cost far less than a single inductor would. High frequency ripple current is not a concern because 1uF 630V polypropylene film capacitors are $1 and have a very low dissipation factor of 0.1%, so the voltage ripple is minimal and the self heating is only 16mW.

- T = (1-0.715)/0.205 = 1.39µs
- ΔV = I*T/C = 7.5*1.39/1 = 10.4V
- ESR = DF/(2π*f*C) = 0.001/(2*3.14*0.205) = 0.777 mohms
- P_{Cap} = 4.5^2*0.000777 = 0.0157 W

TI Webench was given the specifications of V_{IN} = 114V to 126V, V_{Out} = 400V, I_{Out} = 4.5A, and it was used to implement the controller-specific details of the design, but everything was verified and the power stage calculations were done by hand. The entire PFC schematic is shown in Figure 6.10 below to illustrate the general structure of the circuit. The converter is not synchronous because at these voltages the forward diode drop is minimal, and SiC schottky diodes were used to minimize reverse recovery losses under hard switching. The converter operates in Continuous Conduction Mode (CCM), which reduces conduction losses but inevitably results in hard switching for this topology. A UCC2732 4A dual gate driver was specified by Webench, and that drive impedance turned out to be a good match for the FETs we selected.

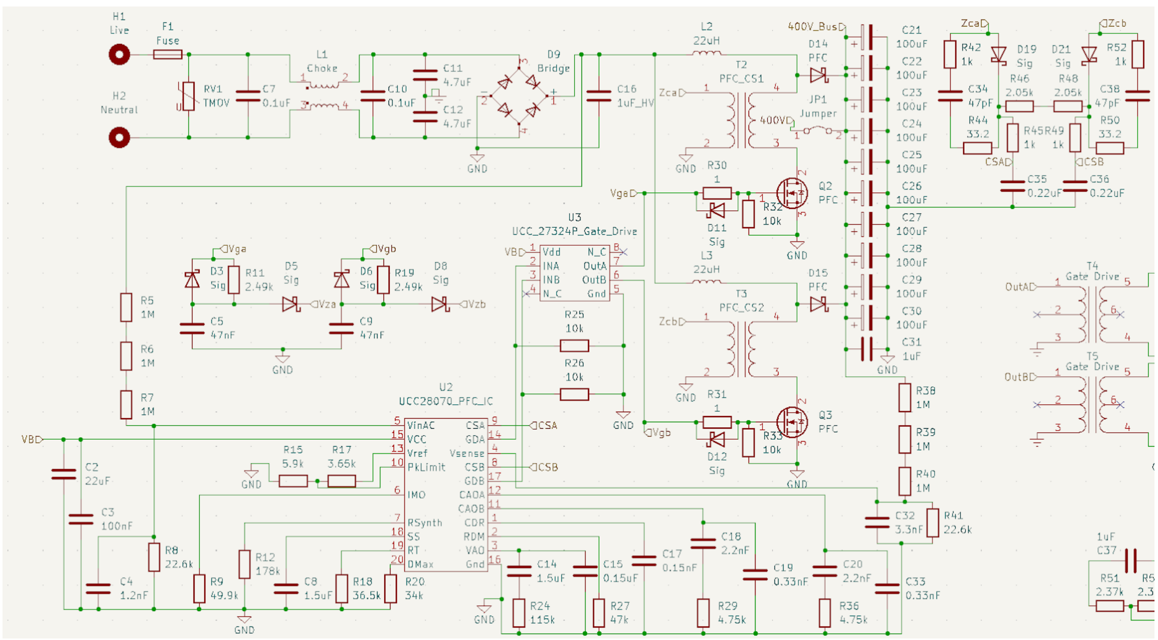


Figure 6.10 - PFC Schematic

The arrangements of diodes and resistors in the middle left and top right of Figure 6.10 were created by Webench, and their function was initially a mystery because the UCC28070 datasheet did not mention it. By searching for “ti webench pfc current sensing”, finding a forum post with a link to the SLUC114 design tool, opening the Excel sheet, and following a link to SLUA479, a reference design for a 300W PFC circuit with the UCC28070, we finally found the function: Improving noise immunity. The circuit uses gate signals to generate a synchronized ramp signal and combines that with the DC offset to make a higher threshold for triggering the start of the next waveform, avoiding false triggering from the noise of the current going discontinuous. The converter uses current mode control to balance current between the phases, and that can be vulnerable to noise.

The first major question was the method of current sense that would be used. Lower power PFC circuits simply have sense resistors, but this would create far too much power loss since the UCC28070 should have almost 4V current sense signal at peak current. A shunt and an amplifier would be too noisy, so we used a current sense transformer. This must be in series with the FETs so the core can be reset, and the controller uses downslope emulation to simulate the half of the inductor current waveform that is not measured. The Webench specifications for the current transformer were minimum primary inductance L_p = 4.85 mH and "Ns1toNp = 5 m." Since current transformers have one primary turn, that means 1/0.005= 200 turns on the secondary, and with 23A peak current and a 33.2 ohm sense resistor the output voltage is 33.2*23/200= 3.82V peak, which matches the slightly less than 4V that the datasheet recommends.

The second major question was which equations would be used to calculate RMS current and switching losses, and these values determine the selection of the inductors, FETs, and diodes. The results of the equations for generic boost converters were not anywhere near that produced by the equations in the datasheet, and on closer inspection we found that both sources were partially wrong, or at least overly conservative [125] [126]. The generic boost converter article took the peak diode current and assumed that it would conduct that current continuously when in fact this loss should be multiplied by (1 - D), and D_{Min} = 0.6, so it's never greater than 0.4. On the issue of MOSFET switching losses, if the voltage and current on the bus are constant, the area of the switching loss triangle is $\frac{1}{2}*V_{Bus}*I_{Bus}*t_S$ where t_S is the time that it takes for the transition to occur. The datasheet also considers the output capacitance, but otherwise both equations implement this simple model, the difference is that the article uses the peak switch current for both rising and falling while the datasheet uses the average. One transition will be peak and one will be minimum, so using the average is most appropriate.

Article Formula:

$$P_{SW} = 0.5 \times V_{O-MAX} \times I_{L-MAX} \times (t_R + t_F) \times f_{SW}$$

Datasheet Formula:

$$P_{M\_sw} = \frac{1}{2} \times f_{SW} \left(V_o \times \frac{I_{line\_max}}{2} \times (t_r + t_f) + C_{oss} \times V_o^{\,2} \right)$$

It is common knowledge that the RMS value of a square wave is the peak times the square root of the duty cycle, and if you assume that the inductor is infinite, the current through the switch in a boost converter has this waveform [127]. The peak of the square wave is the average inductor current, and the ohmic losses in the switch are equal to the resistance times the waveform squared, so if you apply that to the following equation you get the formula in the article, shown below.

$$rms = I\sqrt{D} \qquad\qquad P_{COND} = D_{MAX} \times \left(I_{\text{IN-MAX}}^{\;2} \times R_{DSON-MAX} \right)$$

P_{Con} = 0.715*7.9^2*2*0.055 = 4.91W

Datasheet Formula:

$$P_{M\_cond} = \left(\frac{0.5 \times P_o}{\sqrt{2} \times V_{IN(min)}} \times \sqrt{2 - \frac{16}{3\pi} \times \frac{\sqrt{2} \times V_{IN(min)}}{V_{OUT}}} \right)^2 \times R_{DS(on)}$$

The formula in the datasheet has the same structure where the term that is being squared is the RMS current and the term under the radical is the duty cycle, and to compare their validity we need to determine the limiting case. PFC circuits do not operate from a constant input voltage, obviously. Because we're interested in the highest current case, we consider a sine wave with V_{RMS} = 114V, 5% below nominal. The PFC circuit must draw current in a sine wave shape that is in phase with the voltage, 1800/114= 15.8A and 15.8*2^0.5= 22.3A peak. By necessity, the instantaneous power at the peak of the waveform will be greater than the average, and since 60 Hz is relatively fast from a heat flow perspective, we don't need to design the circuit for 22.3A continuous current. The strange thing about the datasheet formula is that it includes a √2 factor where it shouldn't, dividing the average power by the peak voltage to get something less than the RMS current. The 0.5 factor is because we're only interested in the average current in one leg of the converter. The duty cycle will range from a minimum of 0.6 at the peak of the waveform to the controller's maximum of 0.93 at low line, but there is not much current draw at low line. Considering the RMS voltage, the duty cycle is 0.715, and this is six times higher than the result the datasheet formula produces of 0.12. With these two factors, the datasheet formula gives a conduction loss 12x lower than is reasonable, so in this case the article is correct.

- $D = 1 - V_{In}/V_{Out}$ = 1-(114*2^0.5)/400 = 0.6
- D = 1-114/400 = 0.715
- (2-16/(3*3.14))*2^0.5*114/400) = 0.12

Having established our formulas, the next question is how much power loss each component can tolerate without overheating, expressed as a total loss budget. The article notes that the TO-252 SMD package can dissipate 50°C/W junction to ambient, but this heavily depends on how many vias and how much backside copper area the PCB has [128]. That is irrelevant if an SMD heatsink is used, and there are inexpensive ones that can deliver 25°C/W, although that is the limit. Standard industrial design procedure is to assume an ambient temperature of 40C = 104F, which is probably because power supplies are often run in cabinets or racks, but this will run outdoors where the temperature won't exceed 35C = 95F. Still, at T_J = 125C this only allows for (125-35)/25 = 3.6W of dissipation, which rules out surface mount parts. The PowerPak SO-8 footprint usually replaces the TO-252 in higher performance applications, and while it has lower leadframe inductance, the power dissipation is nearly the same at 3-4W depending on vias. All diodes and FETs will use the TO-220 package because this is the through hole package with the best selection of heatsinks. Commonly available heatsinks allow approximately 10-20W dissipation per device.

Starting with the bridge rectifier, it will see a maximum reverse voltage equal to the AC peak plus one diode drop, 126*2^0.5= 178V, round up to 200V. No schottky diodes are available in the bridge configuration, and at this voltage their advantage would be marginal. Most bridge rectifiers are intended for lower currents and have plastic packages with a high thermal resistance of 3 C/W, but to withstand 15 A_{RMS} on both sides of the bridge I selected one with 1.2 C/W. At 15A, V_F = 0.9, so the total dissipation is 2*0.9*15= 28W, and this leads to a temperature drop of 1.2*28= 33.6 C. The heat sink must be (150-35-33.6)/28= 2.9 C/W or better, and this can be accomplished with a 58 x 61mm extrusion (11.4mm fin height) and 300 ft/min of airflow for 2.3 C/W. Moving on to the half bridge diodes, their conduction loss is simply the other side of the square wave current that flows in the switch. The magnitude is 7.9A with V_F = 1.75V and D = 1-0.715 = 0.285, so P_D = 1.75*7.9*0.285 = 3.94W and ΔT = 1.2*3.94 = 4.7 C. The heat sink must be (125-35-4.7)/3.94= 21 C/W or better, which can be met with an inexpensive stamped sheet metal heatsink with 8.5 C/W at 300 ft/min of airflow.

For the MOSFETs, we have to consider three sources of loss: Conduction losses, overlap losses, and output capacitance (C_{OSS}) losses, although the latter two are grouped together into switching losses. C_{OSS} causes loss because the parasitic drain to source capacitance of the MOSFET is charged when the switch is off and discharges through the switch when it turns on. When a capacitor is charged and discharged through a resistance half the energy is dissipated in that resistance, so considering both the charge and discharge, the losses are C_{OSS}*V_O^2 for one cycle. This is a small source of loss. Conduction losses can be reduced by selecting a FET with a lower R_{DS-ON}, but the only way to reduce overlap losses is to turn the FET on and off faster, but this runs into dv/dt limits with the high side diode and FET itself.

SiC diode speed is limited by the voltage induced in the parasitic inductance of the TO-220 package, but this limit is not binding at 1260 V/ns [129]. These diodes have been subjected to repeated 400 V/ns pulses with no degradation [130], and the datasheet gives a rating of 100 V/ns. Most MOSFETs are rated to 50 V/ns, so this seems to be the limit. Webench recommended the UCC27324 dual gate driver IC, which has 3.7A peak and (12 - 5)/3.7 = 1.89 ohms when sourcing current and 4.28A and 5/4.28 = 1.17 ohms when sinking current. If we assume that there is 5nH of parasitic inductance in the loop between the gate, source, and driver (typical for 1” of PCB trace), and a gate capacitance of 4.1nF, the critical damping resistance is R = (4*5/4.1)^0.5 = 2.21 ohms [131]. Thus, the driver's current capability does not limit switching speed because we need to add a 1 ohm gate resistor to prevent ringing at turn-on, and a small diode will prevent ringing on turn-off. As an aside, we will not exceed the driver's power limit because P_D = 12*65*10^-9*200000 = 0.156W for each FET, which is acceptable given the SOIC-8 junction to ambient thermal resistance of 107.3°C/W, 35+2*0.156*107.3 = 68.5°C.

$$P = C \times V^2 \times f = V \times Q_g \times f$$

$$I_{line\_max} = \frac{P_o}{\eta V_{AC\_min}}$$

To calculate the switching loss, we first calculate what the datasheet defines to be I_{Line_Max}, which is the RMS input current at low line, 1800/114= 15.8A. We selected a FET with an excellent thermal resistance of 0.5 C/W and a good combination of 77nC gate charge, C_{OSS} = 87pF, and $R_{DS\text{-}ON}$ = 0.094 at 100C, so using the driver currents of 3.7A and 4.28A we have a switching loss of 14W and conduction loss of 8.4W for 22.4W in total. The rise and fall times are only 400/(67/3.7)= 22 V/ns and 400/(67/4.28)= 25.6 V/ns respectively, well within the limit, and ΔT = 0.5*22.4 = 11.2 C. The heat sink must be (125-35-11.2)/22.4= 3.5 C/W or better, which can be met with a stamped sheet metal heatsink with 2.4 C/W at 300 ft/min of airflow.

$$P_{M\_sw} = \frac{1}{2} \times f_{SW} \left(V_o \times \frac{I_{line\_max}}{2} \times (t_r + t_f) + C_{oss} \times V_o^{\,2} \right)$$

P_{SW} = 0.5*205000*(400*15.8/2*(77/3.7+77/4.28)*10^-9 + 87*10^-12*400^2) = 14W
P_{Con} = 0.715*7.9^2*2*0.094 = 8.4W

6.4.3 Bias Supply

The role of the bias supply is to be a power supply for the power supply. The regulators and gate drivers need a small amount of power, and this is best provided by a flyback converter. Flybacks are the best topology for low power levels due to their low component count, but they don't scale up due to high peak currents and voltages. The standard design uses a three winding transformer and places the controller IC on the input side, sometimes integrating the MOSFET into the controller itself. There is an output winding with a half wave rectifier diode and a capacitor, as well as an auxiliary winding to provide power to the flyback IC on the other side of the isolation barrier. It would be impractical to have a bias supply for the bias supply, so on startup a chain of high value resistors allows enough current to pass to turn on the FET for the first time. The schematic is shown in Figure 6.11 below.

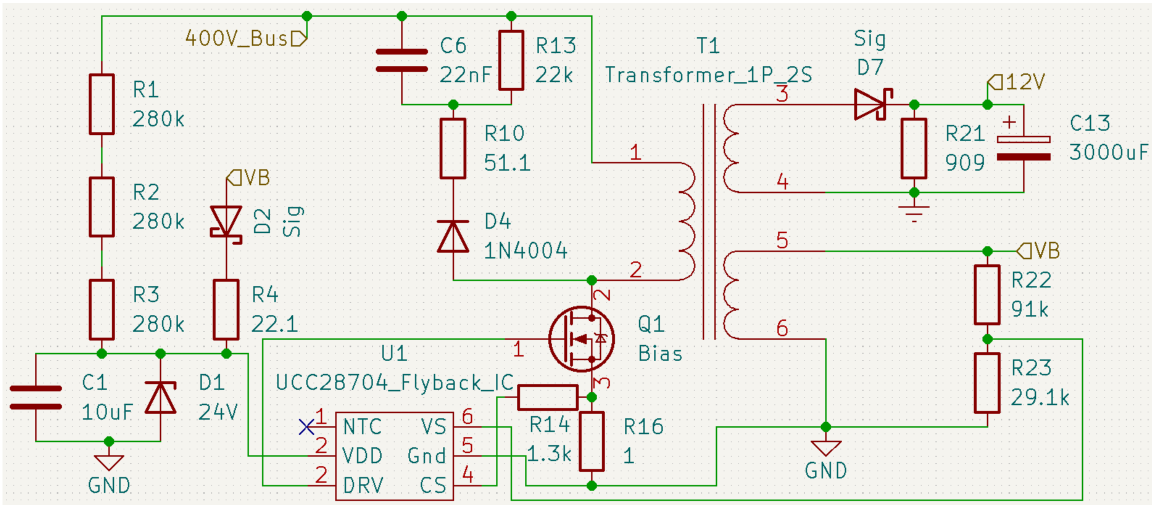


Figure 6.11 - Flyback Bias Supply Schematic

To save time, the circuit was designed with Webench and input parameters of V_{IN} = 114V to 126V, V_{Out} = 12V, and I_{Out} = 0.5A. That current was selected because it seems well in excess of what the full bridge controllers and drivers will need while still being a convenient value to find a transformer for. The cost difference between a 0.05A supply and a 0.5A supply is marginal, less than a few dollars. Webench produced a design using the UCC28730 flyback IC and a transformer with a 54:6:9 turns ratio, aka 9:1 to the secondary and 6:1 to the auxiliary winding. I found a transformer with 8:1 and 6:1 turns ratios with 500mA and 50mA ratings, which is close enough. Since the PFC controller is on the high voltage side of the isolation barrier and the full bridge controller is on the low side, we should need two bias supplies, but we can get away with only using one because the auxiliary winding can also power the PFC. The auxiliary winding produces a higher voltage, but the PFC controller is rated up to 22V (driver and FET rated up to 20V) and draws 2mA typical, maximum of 20mA. The flyback controller uses 2.3mA and the PFC FETs are 13.3mA each, so 2.3+2+13.3*2= 30.9mA total load on the auxiliary winding. The cooling fan uses 29mA, the rest of that circuit uses a little, the full bridge controller uses 5mA, and the FETs use 3.1mA each, so the load on the secondary is approximately 29+5+3.1*4= 46.4mA.

- For PFC FETs, 205000*65*10^-9= 13.3 mA each
- For bridge FETs, 100000*31*10^-9= 3.1 mA each

Resistors R_{S1} and R_{S2} form a voltage divider on the auxiliary winding that indirectly measures and regulates the voltage on the output. The regulation threshold is 4.06V, so the divider scales the auxiliary voltage to this. The auxiliary to secondary turns ratio is (aux-to-pri)/(sec-to-pri) = (pri-to-sec)/(pri-to-aux) = 8/6 = 1.33. A good turn-on voltage would be 85V RMS, so the resistors are 91k and 29.1k.

- R_{S1} = 85*2^0.5/(6*220*10^-6) = 91k
- R_{S2} = 91*4.06/(1.33*(12+0.6)-4.06) = 29.1k

$$R_{S1} = \frac{\sqrt{2} \times V_{IN(run)}}{N_{PA} \times I_{VSL(run)}} \approx \frac{V_{BULK(run)}}{N_{PA} \times I_{VSL(run)}}$$

where $I_{VSL(run)} = 220\mu A$

$$R_{S2} = \frac{R_{S1} \times V_{VSR}}{N_{AS} \times (V_{OCV} + V_F) - V_{VSR}}$$

6.4.4 Output Rectifier

There are two major decisions to make in designing the output rectifier, and the first is whether to use conventional diodes or MOSFETs controlled to provide synchronous rectification, which lowers the forward voltage. We chose not to use synchronous rectification because it introduces a lot of complexity and design difficulty and our output current isn't high enough to justify it. 55.5V average is a fairly high voltage, much greater than the 12V or 5V power supplies where synchronous rectification is commonly used. There can be a catastrophic failure if the FET is on at the wrong time, especially when a battery is connected, so it's not worth the complexity and risk.

Most SMPS converters use either a half wave rectifier or a center tapped transformer with a two diode full wave rectifier. This requires twice as many turns on the secondary as a non-center tapped transformer because the output voltage must be the same, but now only half of the output winding is being used at a time [132] [133]. If you think of it as a square wave with a duty cycle of 0.5, the RMS current in each half is 0.5^0.5= 0.707 times that of the non-center-tapped secondary. For the total losses to be the same, each half must have half the losses of the original, losses are I^2*R, and 0.707^2= 0.5, so this condition is only satisfied if the resistance remains the same [134]. This means that the output winding must have twice as many turns AND have twice the area, quadrupling the required window area. To check my math, let's say we have a secondary with 10 turns, 5A output, and 1 ohm coil resistance for 25W of loss. When we add a center tap each side now has 10 turns and carries 5*0.707 A_{RMS}, and for losses to be 12.5W, R = 12.5/(5*0.707)^2 = 12.5/(25*0.5) = 1 ohm. We simply don't have room for that much copper in the core, and it makes more sense to use the smallest possible ETD49 core with a four diode rectifier. Additionally, a full wave rectifier puts double the voltage stress on the diodes [135], so let's find representative examples, which must be Schottky diodes to minimize switching loss. The voltage thresholds are 65V and 130V with 36A rating, and we calculate the power loss per diode by multiplying V*I*0.5 for the 50% duty cycle. We find that the difference in conduction losses is only 43.2-27= 16.2W. That's a small efficiency gain for 4x the window area.

- 100V 40A 0.66V diode, SDT40A100CT

- At 18A, each leg dissipates 0.6*18*0.5= 5.4W, so 10.8W per diode
- 10.8*4= 43.2W total

- 150V 40A 1.36V diode, VP40PW15C
    - At 18A, each leg dissipates 0.75*18*0.5= 6.75W, so 13.5W per diode
    - 13.5*2= 27W total

For thermal considerations, the temperature drop is ΔT = 2*10.8= 21.6 C, so the heatsink must reach (125-35-21.6)/10.8= 6.3 C/W, and this is easily doable with a shorter version of the same heatsink used on the input FETs.

6.4.5 High Frequency Transformer and Output Inductor

The most difficult part of isolated power supply design is specifying and optimizing the transformer. Other than low power flyback transformers, it is impossible to buy SMPS transformers because the application requirements are so diverse. Designers need so many different voltages, turns ratios, saturation currents, frequencies, winding resistances, and winding configurations that no manufacturer could ever stock a reasonably sized catalog. Instead, transformers are grouped into certain families that share general characteristics and scale up and down. We will focus on the Economical Transformer Design (ETD) family because it is best for medium frequency power supplies. Bobbins that wire is wrapped around are sold separately from cores because different core materials are optimal at different frequencies and temperatures. Two core halves are held in the binding by two clips, and the winding is made of magnet wire and insulated with yellow polyester tape. We had originally planned to use TI Power Stage Designer and the ExcellentIT SMPS Transformer Calculator to make the design effort easier, but these tools were unhelpful, so everything was designed by hand.

Frequency and Skin Depth (δ)

Low frequency ferrites have similar loss curves, plots that show the hysteresis power loss in the core vs flux density swing ΔB at a constant switching frequency. The easily available alternative is 3F36, which is optimized for 1 MHz, but it requires those higher frequencies because the maximum ΔB is lower. There's a tradeoff between frequency and flux density where increasing the frequency does not actually increase power density by very much if losses must remain the same. Realizing that a frequency that high would be impractical, we eventually settled on the 3C94 material and a switching frequency of 100 kHz. The main consideration was the skin effect at higher frequencies and how that would increase the R_{AC}/R_{DC} ratio unless the conductors are extremely thin. Litz wire is expensive and copper tape has a better packing factor but is hard to find and causes high interwinding capacitance, which can cause leakage current problems.

Due to the alternating nature of the full bridge topology, a switching frequency of 100 kHz in each transistor is a frequency of 100 kHz in the transformer but a 200 kHz frequency in the controller IC oscillator and output filter due to the frequency doubling of a full wave rectifier. After checking that this low frequency would have an acceptably sized output filter (next section), we specified the winding gauge. A few years ago a group member performed a 2D FEMM simulation to find the resistances of wires of various diameters at various frequencies, organizing the results in Table 6.1 below. To make the trend more clear, they are expressed in R_{AC}/R_{DC} ratios that show the effective penalty of using that wire at that frequency.

$$\delta = \sqrt{\frac{2\rho}{\omega\mu}}$$

Where ρ = resistivity, μ = permeability, ω = angular frequency = 2πf

For copper, δ = (2*1.678*10^-8/(2*3.1416*f*1.256*10^-6))^0.5 = (0.004253/f)^0.5

Table 6.1 - Skin Depth in Conductors vs Gauge and Frequency				
Frequency	Skin Depth (μm)	R_{AC}/R_{DC} 16G	R_{AC}/R_{DC} 24G	R_{AC}/R_{DC} 30G
10 kHz	652	1.05	1.03	1.03
20 kHz	461	1.12	1.03	1.03
30 kHz	377	1.22	1.03	1.03
40 kHz	326	1.33	1.04	1.03
50 kHz	292	1.45	1.05	1.03
60 kHz	266	1.57	1.05	1.03
70 kHz	246	1.68	1.06	1.03
80 kHz	231	1.78	1.07	1.03
90 kHz	217	1.88	1.08	1.03
100 kHz	206	1.97	1.10	1.03
200 kHz	146	2.66	1.26	1.05
300 kHz	119	3.20	1.47	1.07
400 kHz	103	3.65	1.66	1.11
500 kHz	92	4.05	1.83	1.15
600 kHz	84	4.41	1.99	1.19

700 kHz	78	4.74	2.12	1.24
800 kHz	73	5.05	2.25	1.30
900 kHz	69	5.34	2.37	1.35
1 MHz	65	5.62	2.48	1.40

Given a frequency of 100 kHz, we selected 24 gauge wire because it has a moderate R_{AC}/R_{DC} of 1.1, approximately at the knee of the function. 24 gauge magnet wire has an external diameter 0.0221" = 0.561mm compared to the skin depth of (0.004253/10^5)^0.5 = 0.206mm, so the rule of thumb "radius should equal skin depth" is approximately correct. The other important transformer parameter is the turns ratio. The reduction ratio needs to be moderate enough for the converter to maintain full output voltage at its minimum input, and since it's drawing power from the 400V DC bus with 37.5V peak to peak ripple, the minimum input is 400 - 37.5/2 = 381V. The maximum output is 63+0.7*2= 64.4V, so the ratio could be at most 381/64.4= 5.9 to 1, round down to 5:1. In normal conditions the duty cycle is approximately $D = 0.5*n_1/n_2*V_O/V_{IN}$ = 0.5*5*55.5/400 = 0.347, and the maximum duty cycle is 0.5*5*64.4/381= 0.423.

Output Filter

The main cost of selecting a low switching frequency is that it requires a large output inductor, so before deciding on the frequency for sure I went and found that part. Conceptually speaking, isolated topologies are just like non-isolated ones except they have this magical device called a transformer that changes the voltage to current ratio of the pulses. Otherwise most buck converter design equations apply to buck derived topologies such as the full bridge, and it's helpful for understanding the dynamics of the circuit. Aside from a small leakage inductance, the transformer doesn't store energy, it just couples it to the output, and the output inductor is what actually maintains a constant output current and has varying current ripple as voltage is applied to it. The magnetizing inductance of the primary winding is much higher, so the current doesn't ramp up and down very much, it just switches between flowing across the rails (applying voltage) and circulating through one rail (freewheeling). Under light load the freewheeling current can be very small, which makes it impossible to achieve Zero Voltage Switching (ZVS) in the Phase Shifted Full Bridge (PSFB).

The equation for the critical inductance value is below, and while the rule of thumb is to use 10x the critical value, this only affects how soon the converter transitions to Discontinuous Conduction Mode (DCM) as the load is reduced. It will always go to

DCM eventually, and the converter is designed to tolerate it, especially because it has a pulse skipping mode. Not using synchronous rectification removes the considerable risk of the converter going DCM before the controller is ready. At full load R = 55.5/36 = 1.54 ohms, D = 0.347 (independent of load), T = 1/200000 = 5µs, and L_{Crit} = 1.26µH, so 12.6µH would be the standard value. For $6 we found a 6uH inductor with 62A sat and 2mohm, 36^2*0.002= 2.6W in losses, which is acceptable.

- L_{Crit} = (1 - D)/4 * R*T = (1-0.347)*1.54*5/4 = 1.26

Now that the inductance is known, we can find the amount of output capacitance that must be rated for high ripple current. It's not necessary to use polypropylene capacitors here because although X7R has a rather high 2.5% dissipation factor, that only amounts to 0.346W with 6.8uF of capacitance, the minimum amount. We will actually have almost double that with three 4.7uF 100V ceramic capacitors, and dissipation reduces with capacitance. There will also be a 330uF electrolytic capacitor for load transients and damping. The difference in dissipation factor is obvious because a capacitor with 330/(4.7*3)= 23 times more capacitance has a 38/1.38= 27 times higher ESR when it should be 23x lower.

- C = (1 - D)/(8*L*0.05*(2*f)^2) = (1 - 0.347)/(8*6*10^-6*0.05*(200000)^2) = 6.8 µF
- $\Delta I_{L\text{-}Typ}$ = (n_2/n_1*V_{IN} - V_{Out})*D*T / L = (400/5 - 55.5)*0.347*5/6 = 7.1 A typical
- $\Delta I_{L\text{-}Max}$ = (437.5/5 - 42)*0.347*5/6 = 13.2 A maximum
- $\Delta I_{L\text{-}Cont}$ = (400/5 - 42)*0.347*5/6 = 11 A max continuous
- ESR = 0.025/(2*3.14*0.205*4.7*3) = 1.38 mohms
- P_{Cap} = 11^2*0.00138 = 0.167W

Core Size Estimation

When designing a Switch Mode Power Supply (SMPS), maximum power is usually the first specification. Calculating it precisely involves drawing up an entire design that uses many variables to find the precise number of windings and adds up all the losses to make sure they're within the TDP limit. To get a sense of what size core the design should start with, the following formula gives an estimate, although it assumes passive air cooling, so the absolute limit is higher. Also note that ΔB is the peak to peak swing, so it's double the flux density shown on graphs.

$$AP = A_W A_E = \left(\frac{P_O}{K \, \Delta B \, f_T} \right)^{4/3} \quad cm^4$$

where:

P_O	= Power Output
ΔB	= Flux density swing, Tesla
f_T	= *Transformer* operating frequency
K	= .014 (Forward converter,PPCT)
	= .017 (Bridge, half bridge)

"This formula is based on current density of $420A/cm^2$ in the windings, and assumes a window utilization of 40% copper. At low frequencies, the flux swing is limited by saturation, but above 50kHz (ferrite), ΔB is usually limited by core losses. Use the ΔB value that results in a core loss of $100mW/cm^3$ (2 times the "flux density" given in the core loss curves)." [136]

Using the formula we considered two options, an ETD59 core with N87 (a cheaper ferrite) and an ETD49 core with 3C94. The former was larger than necessary, and the latter seems perfect. We may not be able to fit as much copper into the windings as the formula predicts, but with forced air cooling we have plenty of margin.

- For ETD59 with N87: A_E = 3.55, A_W = 5.19, ΔB = 200 mT at 100 kHz, 5.2W core loss
  - $P_O = K*\Delta B*f_T*(A_W*A_E)$^0.75 = 0.017*0.2*10^5*(5.19*3.55)^0.75 = 3,024 W

- For ETD49 with 3C94: A_E = 2.11, A_W = 3.74, ΔB = 240mT at 100 kHz and $100mW/cm^3$
  - P_O = 0.017*0.24*10^5*(3.74*2.11)^0.75 = 1,921W

Winding Design

Now that we've established the turns ratio, core geometry, and flux density, we can start designing the windings. The ExcellentIT SMPS calculator and other design methods start with the primary inductance, but that spec is not directly relevant because the output inductor determines the output ripple current. What matters is how much energy the transformer needs to deliver to the secondary during one switching cycle T_S because this determines how much flux the core needs to have. It's okay to round down to N_S = 12 and N_P = 60 because that will just slightly increase ΔB and slightly increase core losses. The primary winding will be on the inside because it is easier to insulate there.

- N_S = 64.4*10*10^-6/(0.24*2.11*10^-4) = 12.72 turns
- ρ = 1.72*10^-5 ohm x mm

$$N_{S1} = \frac{V_{O1}\, T_S}{\Delta B \times A_E}$$

Skin effect does not disappear as soon as the wire is divided into parallel strands. Paralleled conductors must rotate through all levels of the winding so that each conductor has the same induced voltage integrated along its length. This works because magnetic field must divide equally between the two winding portions in order to minimize the stored energy = ½*B*H^2*volume [136]. As long as the strands are twisted together and wrapped in a single layer, we can assume that the current in each strand of 24 gauge wire will be the same. The primary and secondary will each have half of the strands, they will just be in different series and parallel configurations. 24 gauge wire has a conductor diameter of 0.5105mm, so the area is 3.14*(0.5105/2)^2= 0.2046mm^2, and we will assume all current is at 100 kHz on average so R_{AC}/R_{DC} = 1.1. Assuming a packing factor of 40%, we can fit 374/0.5105= 732 strands, 366 in each coil, so on the secondary 366/12= 30 strands in parallel (rounded down). On the primary 366/60= 6 strands in parallel. The Mean Length per Turn (MLT) is 85.1mm, so this gives resistances of R_P = 0.0315 ohms and R_S = 0.00126 ohms and a total copper loss of 2.27W.

- R_P = ρ*L/A = 1.1*1.72*10^-5*85.1*60/(6*0.5105) = 0.0315 ohms
  - P_P = 4.5^2*0.0315 = 0.638 W
- R_S = 1.1*1.72*10^-5*85.1*12/(30*0.5105) = 0.00126 ohms
  - P_S = 36^2*0.00126 = 1.63 W, 2.27W total

Transformer Thermals

Much like MOSFETs, transformers are loss limited, and the estimates we make in this section will be a bit optimistic because the loss graphs are for sinusoidal drive, not square wave, and we're not considering eddy currents because they're almost impossible to calculate. Unlike with semiconductors, we can't simply read the internal thermal resistance of the transformer off a datasheet, and it's very difficult to calculate this internal resistance R_I. The external resistance R_E is also hard to know because we're not working with heatsinks that have data provided by manufacturers. For passive cooling the following formula is a good estimate, but it doesn't account for forced air.

$$R_E = \frac{800}{22 \bullet A_W \, \text{in} \, cm^2}$$

R_E = 800/(22*3.74) = 9.7 C/W

The internal thermal resistance of a transformer R_I is very difficult to calculate, but we will try to do it anyway. 3C94 has a minimum loss temperature of 100C, beyond that losses increase, and it shouldn't go beyond 140C. The standard procedure for windings is to assume a 120C maximum hotspot temperature. Wire insulation comes in various classes of 130°, 155°, 180°, 200°, and 250°C, and ours is rated at 155C. The resistance of ferrites is about 5 W/m*K, and each iron leg has a radius of (4.87 - 3.7)/4= 0.293 cm horizontally and 1.63/2= 0.815 cm vertically, (0.293+0.815)/2= 0.554 cm on average. For the core, R_{TH} = 0.00554/(0.00535*5) = 0.207 C/W, not a significant factor.

- Convection R_E = 800/(22*3.75) = 9.7°C/W for ETD49
- Convection loss limit = (120 - 35)/9.7= 8.8W

One large unknown is how well the windings will conduct heat to the airflow when they are covered in polyester tape. Copper is very conductive at 401 W/meter-Kelvin, and polyester tape is somewhat conductive at 0.155 W/m*K, but the air gaps could be a problem. Silicone is the best potting material [137], and there is an inexpensive compound that advertises 3 W/m*K. It costs $30 for a 28mL tube, and the total volume of the core is 3.74*8.51= 31.8 cm^2 = 31.8 mL, so we would only need a few mL with a cost of a few dollars. It's much better than air's thermal conductivity of 0.024 W/m*K.

$$k = \frac{(\dot{Q}_{hpv} V_{hp}) l}{(T_1 - T_2) A}.$$

$$k_e = k_p \frac{(1 + v_c) k_c + (1 - v_c) k_p}{(1 - v_c) k_c + (1 + v_c) k_p}$$

The above equations calculate the thermal conductivity of round conductors in a matrix of encapsulant [138]. V_C is the volumetric conductor ratio, aka the packing factor, which we assume to be 40%, and k_C and k_P are the conductor and encapsulant thermal conductivities, which are also known.

- k_1 = 3*(1.4*401+0.6*3)/(0.6*401+1.4*3) = 6.9 W/m*K
  - R_{TH} = 0.01035/(0.00372*6.9) = 0.40 C/W
- k_2 = 0.024*(1.4*401+0.6*0.024)/(0.6*401+1.4*0.024) = 0.056 W/m*K
  - R_{TH} = 0.01035/(0.00372*0.056) = 50 C/W
- R_{Tape} = 0.000117/(0.00372*0.155) = 0.203 C/W
  - This is for the two layers of 2.3 mil (0.0023") polyester tape, 0.0023*2*25.4= 0.117mm

Much like electrical resistivity and conductivity, to find the resistance from the conductivity we take R = L/(A*σ) since ρ = 1/σ. The winding depth is (3.7 - 1.63)/2= 1.035 cm, so if you assume that all heat is generated at the inner edge of the bobbin, the thermal conductivity is 50 C/W. If you assume that the heating is evenly distributed, the thermal resistance is half of that at 25 C/W. Either way that's quite a bit of resistance to multiply by at least 2.27W of loss in the windings, so it would be prudent to use the epoxy.

$$R_{th} = \frac{1}{\alpha_S S_T} \qquad \alpha_S = \frac{3.33 + 4.8v^{0.8}}{L^{0.288}}$$

- R_{th} = Thermal resistance
- S_T = surface area (m^2)
- v = air velocity (m/s)
- L = Characteristic length (m)

The other major unknown is what the external thermal resistance R_E will be under forced air cooling. The paper that used the equations above achieved 1.7 C/W with air cooling at an unspecified velocity [139]. The only unfamiliar term is "characteristic length" which is the cube root of the volume that the analysis takes place within. For an ETD49 core, the outer edge of the bobbin has a diameter of 3.7 cm and length of 3.2 cm, so let's assume that our area S_T = 0.00372 m^2, v = 300 ft/min = 1.52 m/s, and L = 0.04 m. As long as we have airflow and use the epoxy the transformer shouldn't overheat.

- S_T = 0.00372+0.0032+0.00215 = 0.00907 m^2
  - Cylinder area = 3.14*0.037*0.032= 0.00372 m^2
  - Core perimeter = 2*(4.87+4.94)*1.63*10^-4= 0.00320 m^2
  - Core top & bottom = 2*(4.87*4.94 - 3.6*3.7)*10^-4= 0.00215 m^2
- α_S = (3.33 + 4.8*v^0.8)/L^0.288 = (3.33 + 4.8*1.52^0.8)/0.05^0.288 = 23.79
- R_{TH} = 1/(23.79*0.00907) = 4.63 C/W for the overall transformer
  - R_{TH} = 1/(23.79*0.00372) = 11.3 C/W for the windings
  - R_{TH} = 1/(23.79*0.00535) = 7.86 C/W for the core

6.4.6 Phase Shifted Full Bridge (PSFB) Converter

Full Bridge Transistor

The transformer and output rectifier are finished, but we still need to design the rest of the converter. In a full bridge converter the four main transistors are subjected to the bus voltage and the input current, although technically they carry it at 50% duty cycle, so I_{RMS} = 4.5*(0.5)^0.5= 3.18A. The key question, of course, is how hard to drive them. We shouldn't use SiC because it's expensive and shouldn't be necessary, kind of a design crutch. PSFB converters usually operate with soft switching transitions that don't cause reverse recovery and dv/dt problems, but hard switching will happen at light load. It looks like the solution is just to reduce the body diode conduction current and time as much as possible with a schottky diode while accepting some Q_{RR} losses and being reasonable about the switching time [140]. Turnoff can still be very fast, but turn-on time should be limited by a gate resistor. It looks like di/dt is more important than dv/dt, and it should be limited to 500 A/µs, so a 4.5A ramp could take 4.5*1000/500= 9 ns, which is actually faster than the 16 ns transitions of the PFC stage. The pulldown path does not need a resistor because the diode should prevent ringing.

The limiting factor on drive speed is the critical damping resistance, and for 5nH and 1.27nF that is R = (4*5/1.27)^0.5 = 4 ohms. We round down to 3 ohms to account for resistance in the gate drive, and it'll be 0603 for compactness. The fall time is 31/4= 7.75 ns, and the rise time is harder to calculate because we have a voltage limited current source in series with 3 ohms and a capacitor. Actually, the fact that it's a current source doesn't matter because it hits its voltage limit immediately, and we know that the 31nC of charge will be delivered at the threshold voltage of 3V because that's where the Miller plateau is. The output voltage is Vcc - 1.4V, so the current is (12-1.6-3)/3= 2.47A, and the rise time is t_r = 31/2.47= 12.55 ns, 12.55+7.75= 20.3 ns in total.

We selected a FET with a good thermal resistance of 0.83 C/W and a good combination of 31nC gate charge, C_{OSS} = 80pF, $R_{DS\text{-}ON}$ = 0.19, and t_r+t_f = 20.3 ns, so the total losses are 4.39W. ΔT = 0.83*4.39 = 3.64C. The heat sink must be (125-35-3.64)/4.39= 19.7 C/W or better, which can be met with a stamped heatsink and passive cooling.

$$P_{M\_sw} = \frac{1}{2} \times f_{SW} \left(V_o \times \frac{I_{line\_max}}{2} \times (t_r + t_f) + C_{oss} \times V_o^{\,2} \right)$$

- P_{SW} = 0.5*100000*(400*4.5*20.3*10^-9 + 80*10^-12*400^2) = 2.47W
- P_{Con} = 3.18^2*0.19 = 1.92W

Gate Drive Transformer

A half bridge driver IC would be the obvious choice for this power level, except that we need isolation, and gate drive transformers are the simplest and most robust way to provide that. Some sources claim that gate drive transformers cannot go beyond 50%, but actually they can [141], and we only need to go very slightly beyond 50% because the idea of the PSFB is that the duty cycle is always close to that.

Aside from bulkiness and signal integrity, the biggest problem with Gate Drive Transformers (GDTs) is that the DC blocking capacitance can ring with the leakage inductance of the transformer. To minimize this, toroids are best because they have small leakage inductance, and using a larger DC blocking capacitor also helps to reduce the Q factor [142]. Another way to look at this is that it lowers the minimum resistance required to make the circuit critically damped. The gate resistor is on the primary side, and zener diodes on the output provide clamping to ensure that even with a boost from the turns ratio, the output voltage never goes too high. At 50% duty cycle half the voltage is dropped across the capacitor, which effectively provides a DC level shift from 0-12V to +/- 6V. The negative voltage is helpful for holding the FET off, but it doesn't help with turn-on, so I'll compensate by using a GDT with a 1:2 ratio. To determine the required volt-microsecond product, we look at the output and half the period, so 0.5/10^5= 5μs and V*T= 12*5 = 60 V*μs. While it may be possible to use only two transformers with dual cross coupled secondaries [141], one driver turning on slightly before the other could lead to a partial drive and a longer switching time. It's also hard to find a cost effective 1:2:2 transformer; only 1:2 are available. We found a transformer with 98 V*μs at the input, so it's overly large and expensive but definitely won't saturate. The leakage inductance is 3μH, we already calculated that we would want a 3 ohm gate resistor, and the primary and secondary resistances are 1 ohm and 1.9 ohms, so no resistor is needed. Looking just at the primary, C = 4*L/R^2 = 4*3/1 = 12uF or larger. We'll use ceramic capacitors for compactness and low power dissipation.

Controller IC Parts

We have specced out the power stage and gate drive, but we still need to configure the controller's parameters with various resistors and capacitors. In a transformer it's important for the volt-second balance to be constant across the switching cycle to make sure that you don't end up with a DC current that puts the transformer into saturation. Current is proportional to flux, and the flux needs to go to zero at the end of each cycle, so the current needs to go to zero. Peak Current Mode Control (PCMC) can accomplish this by turning off the FET whenever a given peak value is reached, ensuring that all the peaks are the same and there can be no meaningful difference in current, thus no DC current. PCMC normally uses a current sensor on the drain of the low side FET, but in the full bridge topology this would require two sensors, and separating the FET from ground makes the gate drive more difficult. Thus, the best solution is a current transformer on the input line, especially because we need isolation between the input and the controller IC.

For simplicity we will follow the TI design procedure for the current sense [143]. We somewhat arbitrarily select a CT ratio of 100 because this is commonly available, the typical output inductor current ripple is 11A, and we confirm that the maximum and average duty cycles are 0.434 and 0.393. With this information we calculate the peak current that the transformer must couple and the minimum magnetizing inductance to resolve the output current signal. We have 15.12/1.1= 13.7 times more inductance than required. The peak current threshold voltage is V_P = 2V, so the current sense resistor R_{CS} = 19.1 ohms.

$$CT_{RAT} = \frac{I_P}{I_S}$$

$CT_{Rat} = 100$, $\Delta I_L = 11A$, $D_{Max} = 0.5*63/362.5*5 = 0.434$

$$I_{P1} = \left(\frac{P_{OUT}}{V_{OUT} \times \eta} + \frac{\Delta I_{LOUT}}{2} \right) \frac{1}{a1} + \frac{V_{INMIN} \times D_{MAX}}{L_{MAG} \times 2 \times F_{SW}}$$

- $I_{P1} = (1800/55.5+11/2)*1/5 + 362.5*0.434/(0.001512*2*10^5) = 8.11$ A

$$L_{MAG} \geq \frac{V_{IN} \times (1 - D_{TYP})}{\frac{\Delta I_{LOUT} \times 0.5}{a1} \times 2 \times F_{SW}}$$

- $L_{Threshold} = 400*(1-0.393)/(11*10^5/5) = 1.104$ mH
- $A_L = 4200$ nH/turns^2
- $L_{Mag} = A_L/(N_P)^2 = 4200*60^2 = 15.12$mH

$$R_{CS} = \frac{V_P - 0.3V}{\frac{I_{P1}}{CT_{RAT}} \times 1.1}$$

$R_{CS} = (2 - 0.3)/(8.11*1.1/100) = 19.1$ ohm

$$P_{RCS} = \left(\frac{I_{PRMS1}}{CT_{RAT}} \right)^2 \times R_{CS}$$

$P_{RCS} = (5/100)^2*19.1 = 48$mW

$$R7 = 100 \times R_{CS}$$

R7 = 100*19.1 = 1,910 ohm

To find the maximum voltage that is applied across the diode that resets the current sense transformer, we have to find the resonant frequency of the ZVS transition. The leakage inductance will be approximately 151uH and $C_{OSS} = 80$pF, so the resonant frequency is approximately 1 MHz. D_{Clamp} is the maximum effective duty cycle where ZVS can still be achieved, and it results in 18V reverse voltage and 29.8mW of dissipation, so we can just use a signal schottky diode. Finally, we use R = 1k and C = 330pF to give the CS filter a cut off frequency of approximately 500 kHz.

- $L_{Leak} = 15.1/100 = 151$uH

$$f_R = \frac{1}{2\pi\sqrt{L_S \times (2 \times C_{OSS\_QA\_AVG})}}$$

- $f_R = 1/(2*3.14*(151*2*80*10^{-18})^{0.5}) = 1$ MHz

$$t_{DELAY} = \frac{2}{f_R \times 4}$$

$t_{Delay} = 2/(10^6*4) = 0.5\ \mu s$

$$D_{CLAMP} = \left(\frac{1}{2 \times f_{SW}} - t_{DELAY} \right) \times 2 \times f_{SW}$$

$D_{Clamp} = (1/200000 - 5*10^{-7})*2*100000 = 90\%$

$$V_{DA} = V_P \frac{D_{CLAMP}}{1 - D_{CLAMP}}$$

$V_{DA} = 2*0.9/(1 - 0.9) = 18V$

$$P_{DA} = \frac{P_{OUT} \times 0.6\,V}{V_{INMIN} \times \eta \times CT_{RAT}}$$

$P_{DA} = 1800*0.6/(362.5*100) = 29.8mW$

Finally, the complete AC-DC charger schematic is shown in Figure 6.12 below.

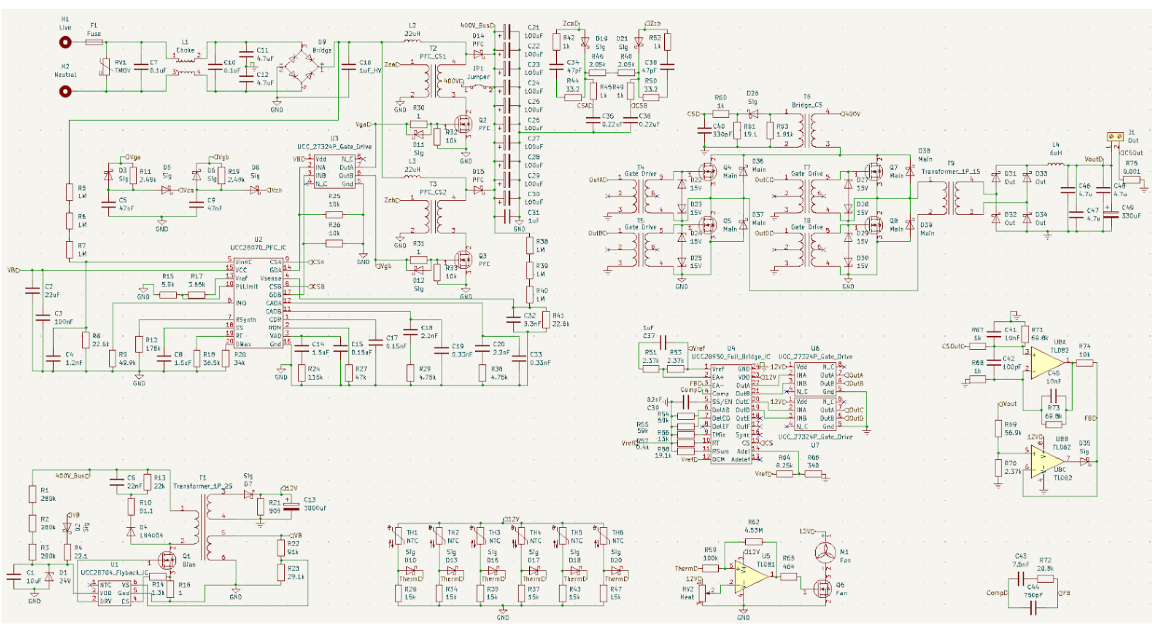


Figure 6.12 - Complete AC-DC Charger Schematic

7 Software Design

The project's software design consists of two main components: the motor controller and the display. Each has a dedicated microcontroller (TI C2000 for the motor controller, ESP32 for the display) and is connected to multiple sensors. The motor controller MCU will obviously be connected to *the motor,* and the display microcontroller will connect to a mobile app (the display) via built-in Bluetooth.The overall system design is shown in Figure 7.1, and the design for each component is further detailed in the sections that follow.

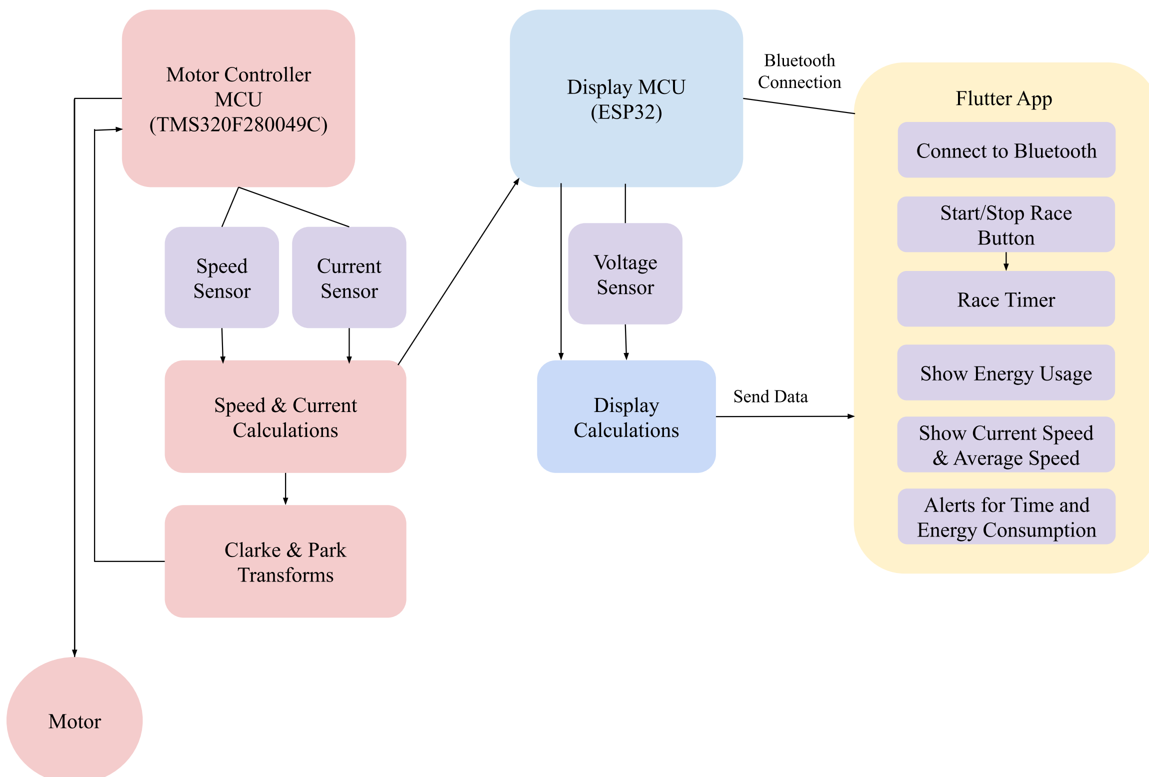


Figure 7.1 - Overall System Design (Software)

7.1 Motor Controller Design

The car's motor will be controlled using sensorless field oriented control, implemented via a dedicated microcontroller with connected sensors. The microcontroller chosen to accomplish this task is the TMS320F280049C from Texas Instruments, which boasts several features to aid field oriented control. These include the control law accelerator (CLA), trigonometric accelerator (TMU), and InstaSPIN-FOC - a tool included on select C2000 devices for expediting programming for FOC applications.

The programmers for the motor controller plan to take advantage of these included features, referring to Texas Instruments' relevant documentation on field oriented control. Though chosen for its superior efficiency to trapezoidal control, field oriented control is highly complex and will likely be the most difficult part of the project from a programming perspective. Taking measurements from speed and current sensors, Clarke and Park Transformations will be completed to implement the FOC algorithm.

7.2 Display Design

When the project was first being discussed, an initial display design was drawn for a 480x320 LCD display. The display was to include a countdown of the time remaining in the race, current speed, average speed (calculated based on a "speed history" from the past few minutes/laps), number of laps completed, and graphs depicting the "actual" power and energy usage compared to target values for effective energy budgeting. The hypothetical design is shown in Figure 7.2.

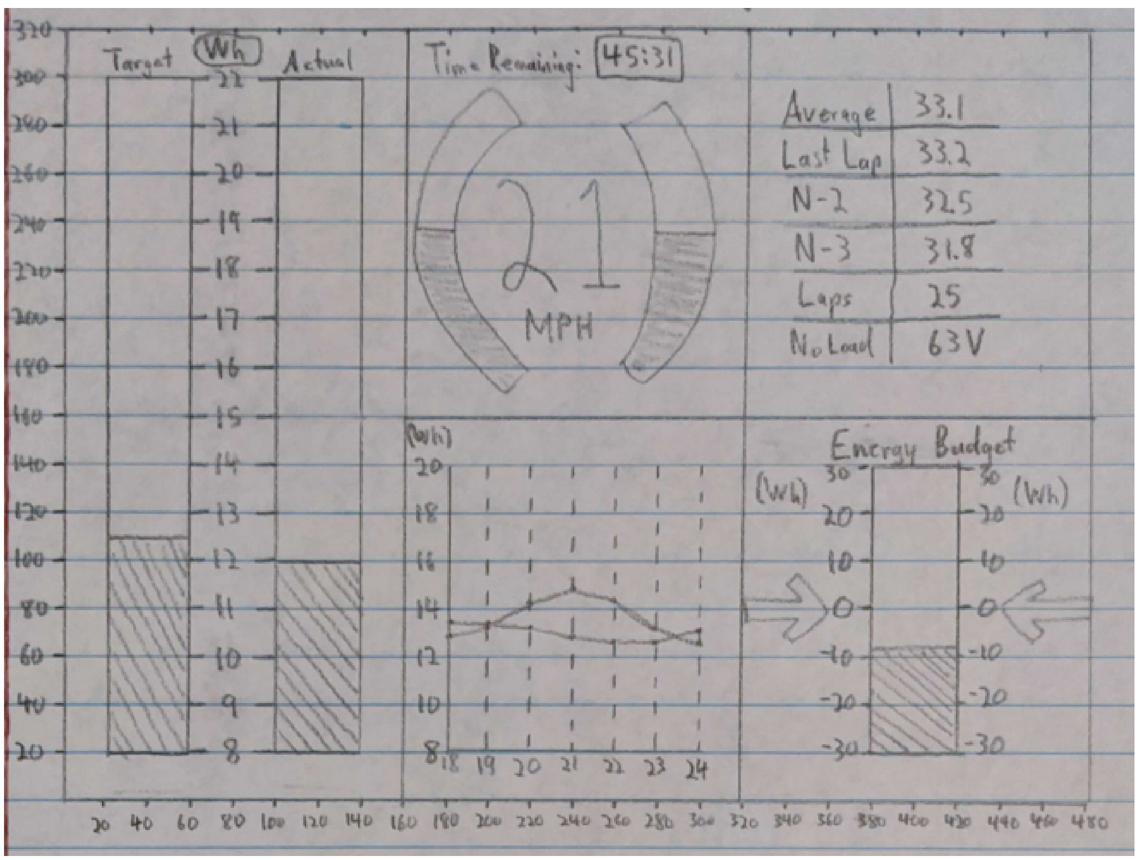


Figure 7.2 - Early Display Design

After further discussion, the group decided that a mobile app display would "look nicer" and be easier to program than an LCD display. Although the underlying goal of the display remained unchanged (showing the driver "useful" information to aid their race performance), the group's consensus on what information could reasonably be obtained and output to the display in real-time evolved.

A lap counter, for example, would be difficult to implement. Simply calculating the number of laps based on the car's speed and elapsed time would likely lead to inaccurate results (especially considering different race locations have tracks of varying lengths), and using an RFID scanning system may interfere with the tracker used by race administration. While it could still be done in theory, the group identified the lap counter as one of their "stretch" goals rather than a must-have for the display.

The "basic" display outputs the group agreed to be reasonable to strive for were a countdown timer, current and average speed, and energy budget. The countdown timer was so easy to implement that even ChatGPT could do it (and it did - see Chapter 9 for testing details), and the current/average speed and energy budget could be calculated from sensors the group planned to use (including speed, current, and voltage sensors).

The display should indicate when there are less than five minutes remaining in the race, most likely by changing the text color of the timer to red (from a more neutral color such as white). The driver should also be alerted if the car's energy consumption is determined to be "too high" to successfully complete the race based on the calculated energy budget, or if the Bluetooth connection to the ESP32 has been lost. In the case of a dropped Bluetooth connection, the speed and energy calculations should be paused, but the timer should continue until the end of the race unless the driver presses the "stop" button, as it is assumed the driver is still driving and would like to know how much timer remains in the race even if speed and energy information becomes unavailable.

When the display app is opened, the user will first see a friendly welcome page prompting them to connect to the microcontroller via Bluetooth. The user will need to enable Bluetooth on their device and pair with the microcontroller if they have not already. The microcontroller should be easy to identify from a list of available devices, having a characteristic name such as "My ESP32", which can be defined in the microcontroller's code. After establishing a Bluetooth connection, the app will show the main display page, which will tentatively consist of the items described above: a one-hour countdown timer in 'mm:ss' format, the car's current and average speed, an energy budget, and a start/stop button for the race. Due to its intended purpose, the start/stop button is the app's only necessary user interaction besides the initial Bluetooth connection. For added context, there will be an 'information' icon at the bottom right corner of the screen, which the user can press for an annotated version of the data displayed on the main page. This option will only be available before pressing the "Start Race" button, to avoid accidental interruptions to the race. Later, a settings page may also be added, as described in the Goals and Objectives (2.2) section of this report.

Figure 7.3 shows a tentative design for the app's startup screen, 7.4 shows the information screen, and 7.5 and 7.6 show examples of the main display both before starting the race and when there are less than five minutes remaining in the race, respectively. Figure 7.8 shows alert symbols that may appear if the Bluetooth connection has been lost or if the driver's energy usage exceeds the amount required to finish the race without running out of battery.

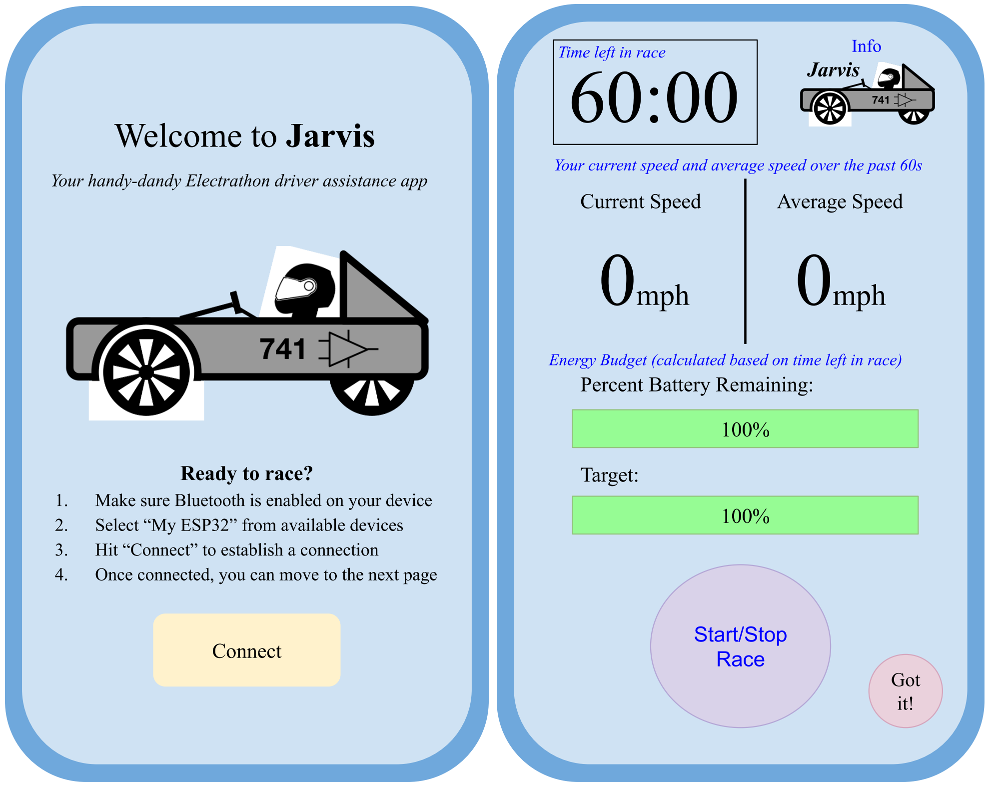


Figure 7.3 - Startup Page & Figure 7.4 - Information Page

Figure 7.5 - Before Starting Race & Figure 7.6 - Near End of Race

Figure 7.7 shows a few possible 'alert' icons that may be displayed. Similar to a typical car's dash lights, these icons would appear under certain conditions and do not impact the performance of the vehicle or the overall display output (aside from the small icon). The examples shown are a symbol for lost Bluetooth connection and a symbol for when the driver is using more energy than they should in order to complete the race.

As previously described, if the Bluetooth connection is lost, speed and energy calculations will pause (since the app will no longer be receiving data from the ESP32), but the countdown timer will continue until either the race ends or the driver presses the "stop timer" button.

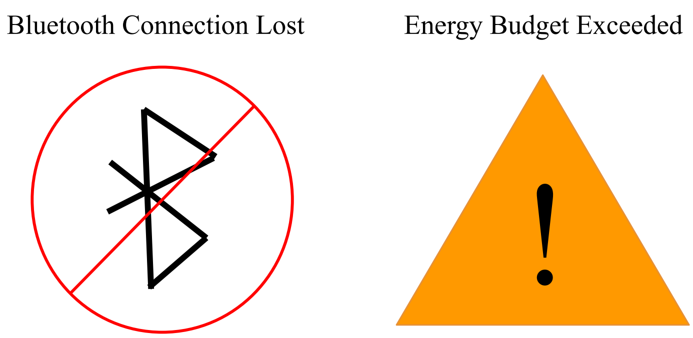


Figure 7.7 - Alert Symbols

Figure 7.8 shows a simple state diagram for the display app. As described, the app consists of three screens, with four main "states": the initial startup screen before the user connects to Bluetooth, the main page in "idle" state when the race has not yet started, the app information page, and the main page in "active" state after pressing the "Start Race" button. The app returns to idle state when the user presses the "Stop Race" button or when the race timer runs out.

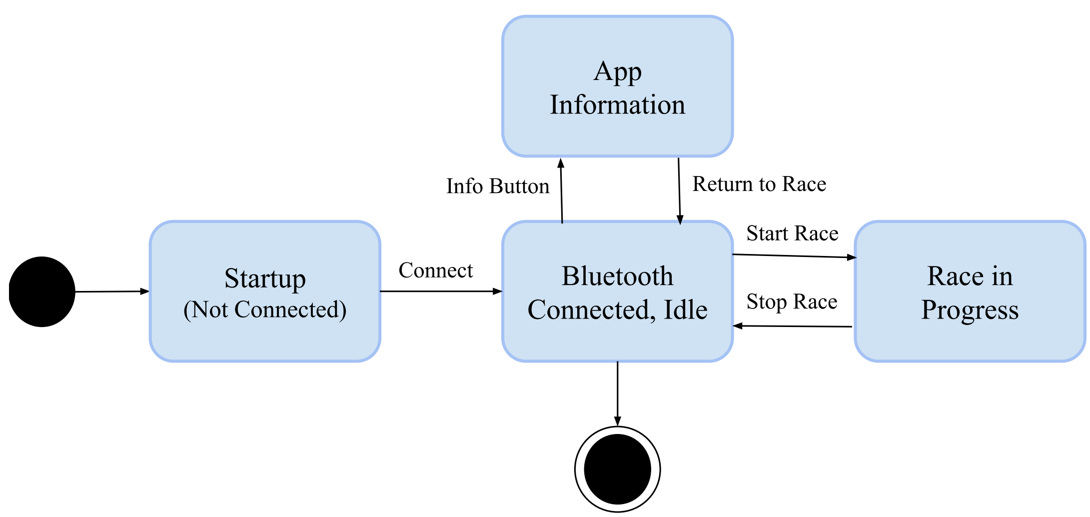


Figure 7.8 - State Diagram

8 System Fabrication/Prototype Construction

This section contains PCB layouts and details on the physical construction of the car's electrical system. The overall design is separated into individual circuits and PCBs by function as designed in Chapter 6, and the PCBs are linked by wires to form a cohesive whole.

8.1 Overall System Schematic and Component PCBs

First, Figure 8.1 below is an overall schematic of all the major circuits in this project.

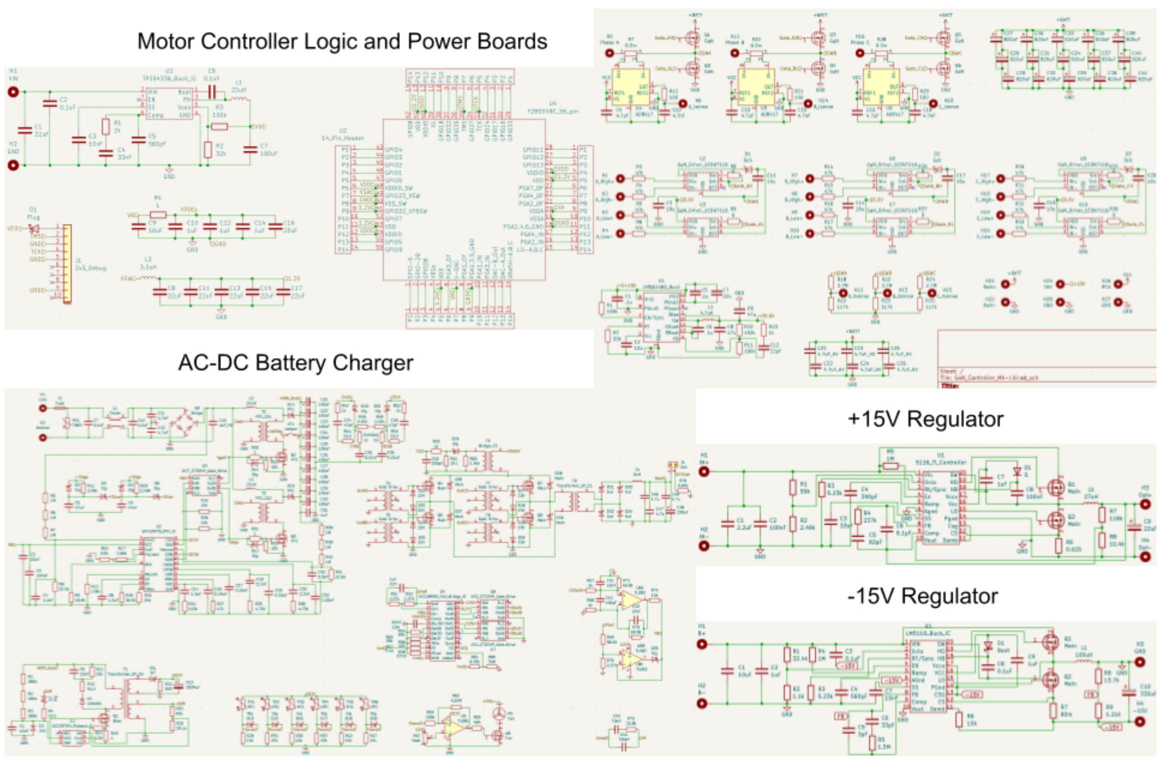


Figure 8.1 - Overall Schematic

After each schematics is created in Electronic Design Automation (EDA) software, footprints are assigned to each symbol, and the schematic is converted into a Printed Circuit Board (PCB). All major PCBs in this project are listed below. The motor controller power and data boards are separated for modularity because we may need to modify one but not the other. Figure 8.2 has the motor controller half bridges and gate drivers, Figure 8.3 has the microcontroller and logic, Figure 8.4 has the +15V regulator, and Figure 8.5 has the -15V regulator.

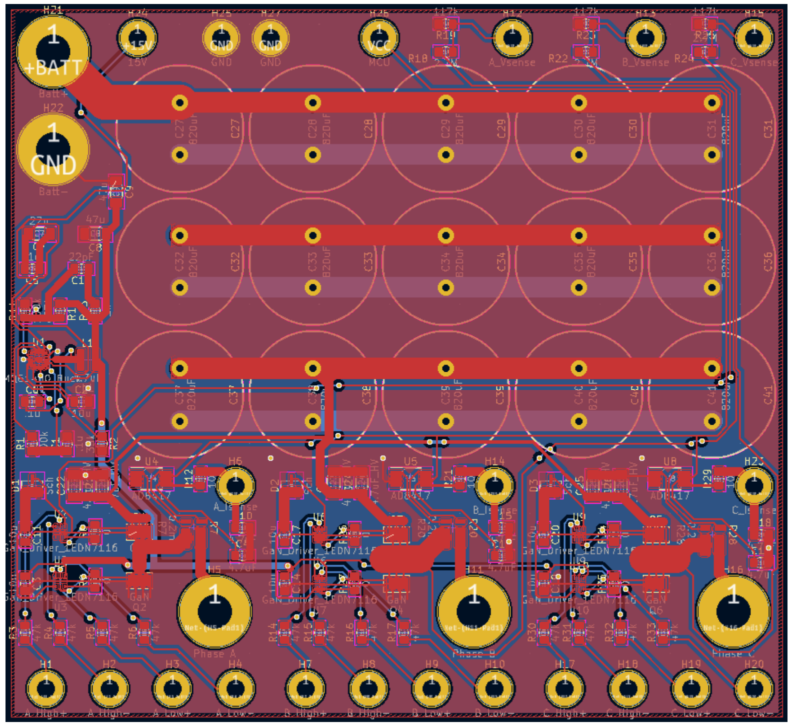


Figure 8.2 - Motor Controller Power PCB

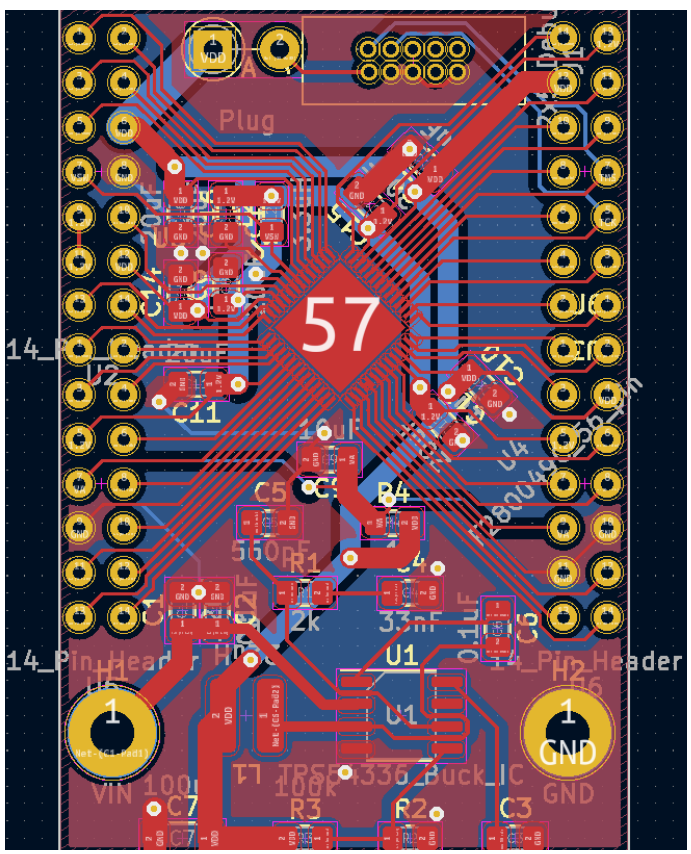


Figure 8.3 - Motor Controller Logic PCB

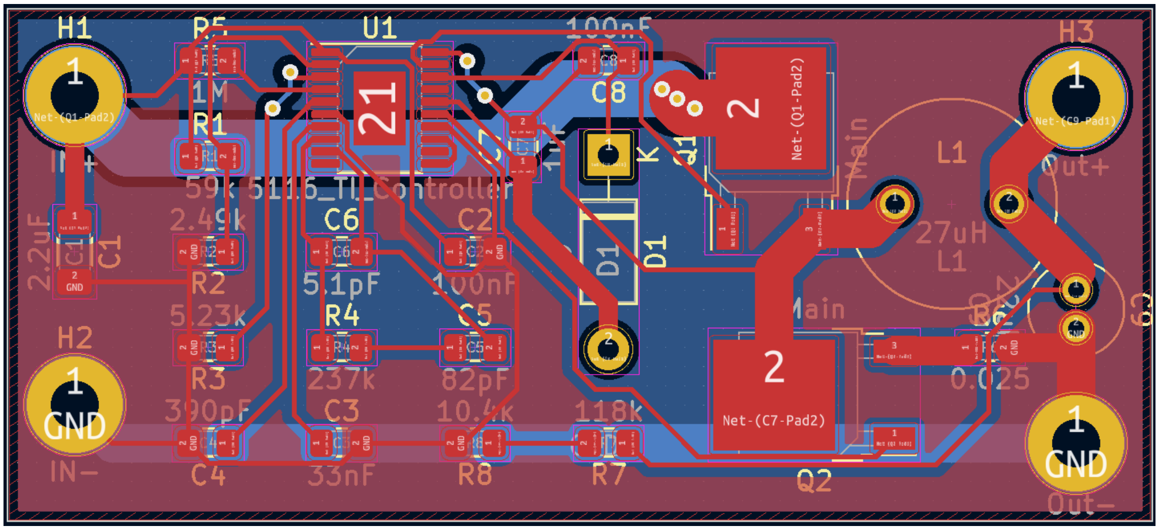


Figure 8.4 - Positive 15V Regulator

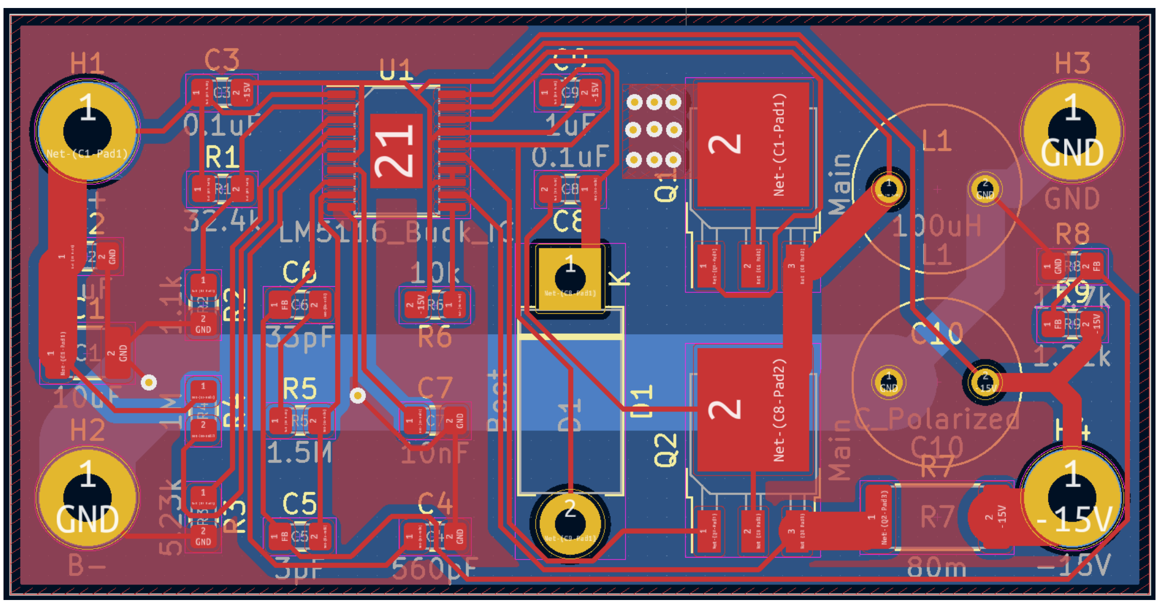


Figure 8.5 - Negative 15V Regulator

8.2 Battery Pack Construction

For reasons explained in Chapter 6, the battery pack is composed of 135 individual 18650 size cells. These are held together by injection molded plastic cell holders that interlock to form any desired shape. Each holder is a 20mm x 20mm square with an 18mm diameter hole in the center, and we need at least 135*2= 270 pieces. It is possible to assemble an 18650 battery pack by gluing the cells together, but it is more robust to use the cell holders, and they do not add much size and weight. Mechanical failure of the pack would result in a short circuit and possible fire, so it is important that the pack stays together.

To simplify the busbar configuration, the mechanical layout will match the electrical layout, which is 15 x 9. This leads to dimensions of 15*20/25.4= 11.8” and 9*20/25.4= 7.1”, so the battery pack will fit within a 12” x 12” x 4” steel junction box that is insulated with fish paper. This is easier than building a custom metal enclosure, and the junction box has pass throughs for wiring. The ends of the cells will be spot welded to bus bars made of nickel plated strip, and the BMS and main power wires will be soldered to the strip between cells. The power wires will run parallel to the edge of the block and be zip tied down for strain relief. The ends of the pack will be covered in fish paper that is taped down, and all other connections will be insulated with heat shrink. Fish paper is selected due to its insulation and wear resistance, and overall this is a very standard battery construction method.

8.3 Wires and Connectors

By far the most common size of bullet connectors is 4mm in diameter, and these are rated for 90A and can fit up to a 12 gauge wire. They are used on the motor phase wires since the current is distributed among three leads, but between the motor controller and battery the Electrathon rulebook requires 6 gauge wire because we have a 100A circuit breaker. A large breaker is necessary because the motor has a continuous current rating of 80A, and peaks could be a bit higher. The largest bullet connectors available are 6.5mm in diameter and rated for 200A. They are only designed for 8 gauge wire, but 6 gauge can probably fit with a bit of trimming.

On the PCBs, the 6 gauge motor controller wires will be soldered into 5 mm diameter holes because the conductor is 4.11 mm in diameter. The controller does not plug directly into the battery, technically it plugs into the car’s wiring harness and that plugs into the battery and many other things. The AC side of the charger will have a standard NEMA C13 port that connects to the 120V outlet through a NEMA 5-15P to C13 cable. The output will be a terminal block with 12.7mm pitch that is rated to 50A, much larger than the more common 5mm pitch terminals and 5x larger than the 0.1” pins of a breadboard. The 100A breaker also uses screw terminals, and the charger leads will be 4mm bullet connectors with dual 12 gauge wires.

9 System Testing

This chapter details testing of the hardware and software components used in the project. Section 9.1 describes hardware testing while 9.2 describes software testing. 9.3 discusses system integration, and 9.4 outlines the group's plan for senior design 2.

9.1 Hardware Testing Equipment and Facilities

Safely testing power circuits requires somewhat specialized equipment and facilities. Besides the usual multimeters and low voltage power supplies, it is helpful to have a higher voltage supply, and we have one that goes up to 60V 5A. To gradually apply AC input voltage to a power supply, we have a 120V 15A variable transformer. To probe the input side of an AC-DC converter, we have a differential isolated oscilloscope probe, and to test the output of various regulators we have a 200V 20A controllable load. To measure harmonic currents, we have an oscilloscope current probe. For programming and debugging the microcontroller, we have a TI Launchpad with the same F280049C model. In the Senior Design 2 lab we will have access to high quality oscilloscopes and a safe test environment.

9.2 Motor Controller Hardware Testing

In the motor controller, one reason we chose to power the gate drivers and microcontroller from a separate 15V supply is that it allows the main input voltage to be very low. In fact, the DC bus voltage could be below 15V, all the way down to 0V. For a constant resistance, power flow is proportional to voltage squared, so this will significantly reduce the energy of switching events and short circuits. Some experimentation will be necessary to confirm hardware parameters before any attempt can be made at spinning a motor.

9.2.1 5.5V, 3.3V, and 1.2V Regulators

Regulators are simple to characterize, at least at the level we are interested in. We are only interested in whether the output voltage is close to the specification at zero and maximum load current. The 5.5V regulator is integrated into the power PCB, but it can be tested first by simply assembling the regulator first and leaving the rest of the board unpopulated. It will be tested by connecting it to a variable power supply set to 15V and gradually increasing the load until the current reaches 10mA. The 3.3V regulator will be tested in a similar way by waiting to place the microcontroller on the board until it is confirmed that the regulator can output a stable voltage.

The 1.2V regulator can't be tested separately because it is built into the microcontroller. After the other regulators are tested the microcontroller will be placed onto the PCB and programmed with a default program, such as one that blinks an LED on a pin. The DC-DC regulator is enabled by default, and if the microcontroller runs with all peripherals enabled it shows that the regulator works.

9.2.2 Gate Drivers and FETs

The low side gate drivers can be tested without any voltage on the main power bus, and this is a good place to start because there are no consequences to turning a FET on slowly or at the wrong time. By connecting the input of the driver to a microcontroller or function generator the gate can be pulsed on and off, and oscilloscope probes on the input and output can measure the turn-on speed and check for ringing. The FET output impedance can be measured by connecting a voltage source and impedance between the switch node and ground. The performance of the gate drive can be tested at various frequencies, and we will only move on if the performance is satisfactory.

When a low voltage and series impedance are connected to the bus with a low impedance between the switch node and ground, the experiment gets much more interesting. We can drive the cross coupled inputs to see if the deadtime is satisfactory, and we can directly observe the voltage overshoot and reverse conduction of the half bridge if the load has some inductance. As performance improves we can gradually increase the voltage and load until the FETs are driving a dummy circuit of approximately the same power as the actual motor. Parts of the code could also be tested with this dummy circuit.

9.2.3 Current Sensing

The current sensors can be tested by soldering a wire to the switch node and another to the phase wire, allowing a Constant Current power supply to pass current through the hall effect sensor. The resulting voltage can be read by a multimeter and the microcontroller’s ADC to make sure everything is calibrated correctly.

9.3 AC-DC Battery Charger Testing

This section includes testing for the AC-DC Battery Charger. In 9.3.1, the PFC and Bias Supply are discussed, while 9.3.2 describes testing of the main converter.

9.3.1 PFC and Bias Supply

The PFC circuit has by far the most complex test procedure, or at least the most detailed. It must be tested alongside the bias supply, which is assumed to work because it is very simple. The full bridge converter is linked by a solder bridge jumper that will not be connected until after the PFC circuit is tested. The two tests that must be performed are for harmonic currents and EMC. The circuit will be slowly powered up with the variac, and at 85 V_{RMS} the bias supply is expected to turn on and power up the PFC circuit, which will initially have no load. The output voltage exceeds that of the controllable load, so a few spare parts will be used to make one. A 47 ohm power resistor will be connected in series with a 600V IGBT, both screwed to a large aluminum heatsink. An LM351 op amp will drive the IGBT gate with the inverting input connected to a several ohm power resistor between the IGBT emitter and ground, and the non-inverting input will connect to a low voltage power supply. This is the classic circuit for a constant current source.

Once it has been established that the PFC circuit can supply full power and the waveforms look normal, a current probe will be attached to the input and an oscilloscope will be placed into FFT mode to measure the harmonic currents. The standard method for measuring EMI involves isolating the device under test from the rest of the grid through a 50µH/50Ω V-type Line Impedance Stabilization Network (LISN) [123]. We have constructed a simple version that we will use for testing. The noise limit is 60 dBµV = 1mV, and an isolated probe would be a good idea but is not necessary since the output is referenced to ground. EMI will also be measured in FFT mode, although the window will be set to a much higher frequency.

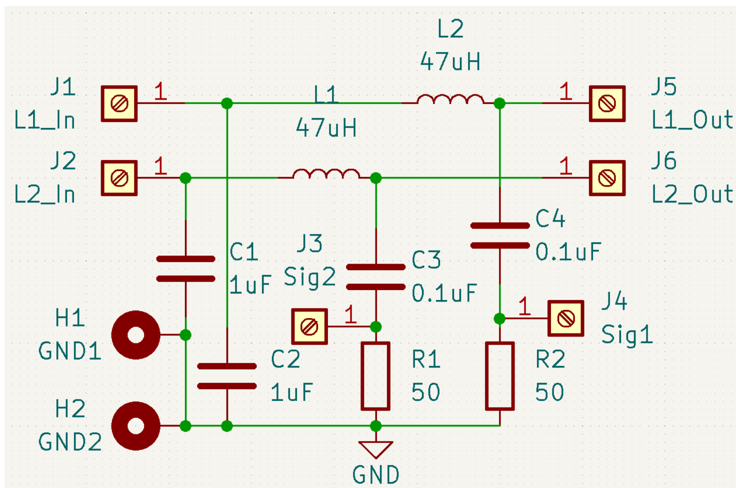


Figure 9.1 - L.I.S.N. Schematic

9.3.2 Main Converter

Once it has been established that the PFC circuit works, the solder jumper will be closed and the circuit will slowly be powered up with the variac. Many voltages will be probed on the gate drive transformers, current sense transformers, controller output pins, and output rectifiers. If everything seems normal a controllable load will be applied up to the 150W limit. At that point the charger will be connected to the battery through a 45A inline fuse, and the charging current will be monitored.

9.4 Software Testing

As described in Chapter 7 - Software Design, the software for this project mainly revolves around programming of the motor controller and in-car display, each having an associated microcontroller. The microcontrollers will be tested by running “demo” applications: one for the motor controller, as well as separate demo applications for both the display’s microcontroller and the display app itself.

9.4.1 Motor Controller MCU Testing

The microcontroller chosen for the motor controller was Texas Instruments’ TMS320F280049C. To help with prototyping, the group purchased the “TI LaunchPad with F28004x C2000 real-time MCU" for the motor controller MCU. Though actual programming of the microcontroller is expected to be complex and time-consuming, basic testing is fairly simple.

Code Composer Studio - Texas Instruments' development environment - was used, along with C2000Ware SDK, which includes examples, libraries, and low level drivers specific to TI's C2000 devices. Code Composer Studio's built-in "Resource Explorer" was consulted for guidance on testing the development board.

The demo project "led_ex1_blinky" was imported from C2000Ware's collection of examples for the F280049C. An LED blinking is an ideal starting example, as it is visually obvious whether or not the program is working as intended.There were a few points of confusion when setting up the board, as TI's documentation is not especially intuitive to the inexperienced programmer. First, there were issues connecting to the target device. In the project's properties, the USB connection had to be modified from the default "Texas Instruments XDS100v2 USB Debug Probe" to "Texas Instruments XDS100 USB Debug Probe", which can be checked by clicking the "verify" button on the General Properties page.

Now the program was able to run, but the state of the LED did not change. The issue was the build configuration, which is set to "CPU1-RAM" by default, but should be set to "CPU_LAUNCHXL_RAM" (or "CPU_LAUNCHXL_FLASH", depending on the application) when working with the development board.

After making this change, the red LED on the board started blinking as it should. A few lines of code were added to toggle the green LED next to it as well. The program ran without issue. A demonstration of this test is featured in the 3-minute component testing video that accompanies this report.

9.4.2 Display Testing

Before making a final decision on which app development platform to use, a few setup tutorials were attempted to get a hands-on view of which option described in the "part selection" section of this report - Thunkable, Flutter, or ReactNative - would be most preferable to use. The group wanted to minimize overhead challenges involved with configuration and learning of new tools to provide more time for skill-building relevant to project goals and industry needs.

The first tutorial followed was "ESP32 BLE + Android + Arduino IDE = AWESOME" by Tim Woo from Instructables.com, using the codeless Thunkable platform. This seemed like the most logical starting point, isolating the Bluetooth connectivity and microcontroller programming components before diving into more complex development code in an entirely new language.

Just when the display programmer was about to start working, she encountered an immediate two-day setback: although the group owned multiple ESP32 development boards, they did *not* have a compatible micro-USB cable. This was realized when the programmer opened the Arduino IDE, installed the appropriate package for the board ("esp32 by Espressif Systems"), and connected the board via USB. The LED on the board turned on, showing that it was receiving power, but no COM port was available for selection within the Arduino IDE. Online troubleshooting suggested that the USB cable the programmer was using (likely a cheap charging cable intended for cell phones) did not have dedicated data lines. The issue was resolved after the delivery of a micro USB from Amazon with a description specifically mentioning data lines and compatibility with the ESP32. Now the programmer could proceed with programming the microcontroller.

A demo code for the ESP32 was downloaded from the tutorial website and opened in Arduino IDE. The Thunkable Companion App, "Thunkable Live" was downloaded from the Google Play store, allowing for live testing on a mobile device. The tutorial instructs users to download a provided .aia file and upload it to Thunkable. Unfortunately, since the tutorial's publication, Thunkable has transitioned from "Thunkable Classic" to "Thunkable X", in which it is not clear how to upload files, if possible at all. Instead, the programmer attempted to follow the tutorial from scratch.

After parsing a collection of tutorials from the Thunkable website, using Thunkable seemed more complicated than worth pursuing. The programmer moved on to the next platform - Flutter.

Flutter uses the Dart language, which is supposedly similar to Java. Dart SDK for Windows (64-bit) was downloaded from https://dart.dev/get-dart/archive. To use Dart, a new path had to be added to the system's environmental variables:

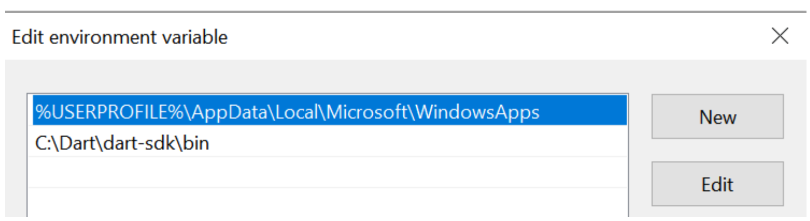


Figure 9.2 - Dart Path Setup on Windows 10

Next, Flutter for Windows was downloaded from and installed following a tutorial from Codemy.com. Installation of Flutter is more involved than typical application installations, as it requires the user to create a specific directory and define a path for it. In the Windows control panel, a new path was added for Flutter:

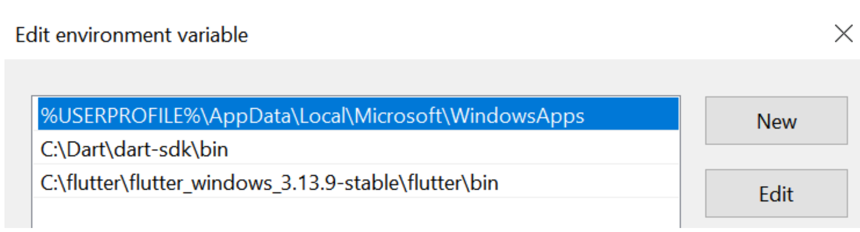


Figure 9.3 - Flutter Path Setup on Windows 10

Flutter and Dart extensions were added to Visual Studio Code. A series of frustrations were encountered while trying to configure Flutter, but finally a demo app was successfully set up and launched via Chrome web browser:

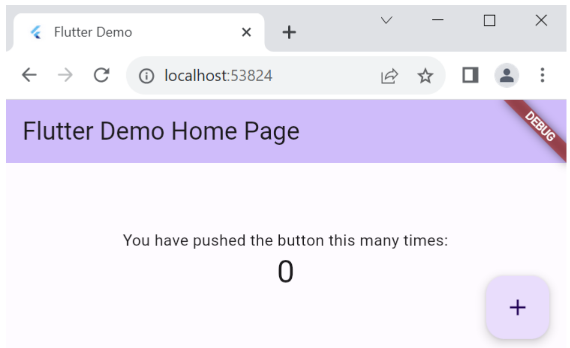


Figure 9.4 - Flutter's Equivalent of a "Hello World!" Program

At first, several tutorials and github projects were attempted for building a Bluetooth app with ESP32 and Flutter, but most of these only led to frustration as they produced errors that were difficult to debug. Instead of trying to do everything at once, the programmer decided to test each part individually. First, the programmer practiced working with Flutter by building a basic kitchen timer. Then, the ESP32 was isolated, testing its Bluetooth capabilities.

First, a simple Flutter timer app was programmed. A new project was created by running "flutter create timer" from the VS Code terminal, where "timer" is the name of the project. ChatGPT was asked to write a code for a flutter app that counts down from one hour in "mm:ss" format. An *unsuccessful* example of its attempt is described in Chapter 5 of this report. A few weeks later, the programmer asked again, and this time ChatGPT was mostly successful at accomplishing the desired task. It generated a code to create a basic timer with start/stop and reset buttons. There were a few bugs the programmer needed to resolve, but the program provided a nice starting point for further development.

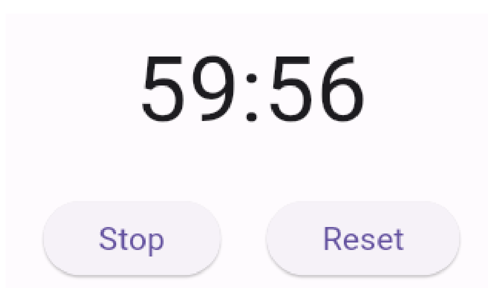


Figure 9.5 - ChatGPT's Basic Timer App

Each Flutter program has a "pubspec.yaml" file, containing publication, version, and dependency information. Dependencies are additional packages not included in the default application, similar to libraries or imports. For Bluetooth connections to work, the "FlutterBlue" package (flutter_blue: ^0.8.0) is needed.

Among several project folders included by default in a flutter project (.dart_tool, .idea, android, ios, lib, linux, macos, test, web, and windows), the main source code for an app exists in the "lib" folder, typically named "main.dart". As with other languages, more files can be added as desired to improve the program's readability, as long as each file is imported appropriately where its functions are needed in other files.

Unfamiliar with many of the configuration nuances involved in app development, the programmer encountered errors when trying to import packages. To successfully use non-default packages in a flutter project, a few commands need to be run from the VS Code terminal. First, "start ms-settings:developers:" opens Windows settings, where developer mode can be enabled.

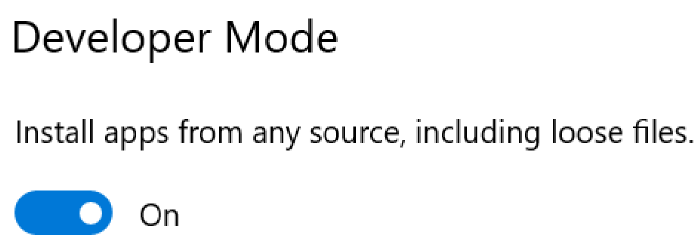


Figure 9.6 - Developer Mode on Windows

"Install apps from any source, including loose files" poses an obvious security risk, so developer mode should only be enabled when necessary.

Next, "flutter channel master" and "flutter pub get" or "flutter packages get" gets the required packages based on the dependencies defined in the pubspec.yaml file. "flutter pub get" must be run whenever new packages are added or when re-opening the project after closing VS Code.

To test Flutter applications on a mobile phone, developer mode also has to be enabled on the phone. For Android, this can be done by going to device settings, "about phone", and tapping the phone's "build number" repeatedly. Now, developer options are available. Clicking on Developer Options, USB debugging should be enabled. This is intended for development purposes only and should be turned off when not in use. Additionally, "USB Configuration" should be set to "MIDI" (the default is typically charging/file transfer only). Running "flutter devices" from the VS Code, the phone should now be included in a list of devices the Flutter application can be run from.

After establishing how to create and run apps in Flutter, the programmer returned to the ESP32 to test Bluetooth connectivity. When the libraries for ESP32 are installed in Arduino, a list of example projects using the board become available under "File>Examples>Examples for ESP32-WROOM DA Module". A tutorial by Rui Santos titled “ESP32 Bluetooth Classic with Arduino IDE – Getting Started” was referenced for testing the example “SerialToSerialBT”.

The code was opened in the Arduino IDE, compiled, and uploaded to the ESP32. “Tools>Serial Monitor” opens the serial port monitor, where the baud rate should be set to 115200. Pressing the ESP32 enable button on the development board will show the output to the terminal: “The device with name "My ESP32" is started. Now you can pair it with Bluetooth!” (Where “My ESP32” is the name of the ESP32 device, which can be changed). For testing purposes, the Smartphone app “Serial Bluetooth Terminal” was downloaded from the Google Play store. With Bluetooth enabled on the phone, the ESP32 was found on the list of available devices and paired.

A successful connection was verified by sending the message “Hello world!” from the Serial Bluetooth Terminal app to the ESP32.

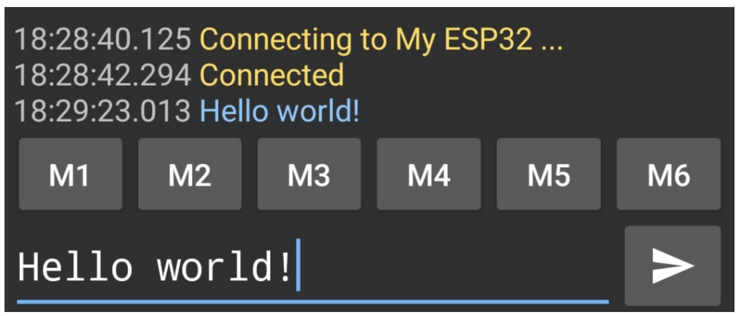


Figure 9.7 - Sending a Message

The message appears in the Serial terminal:

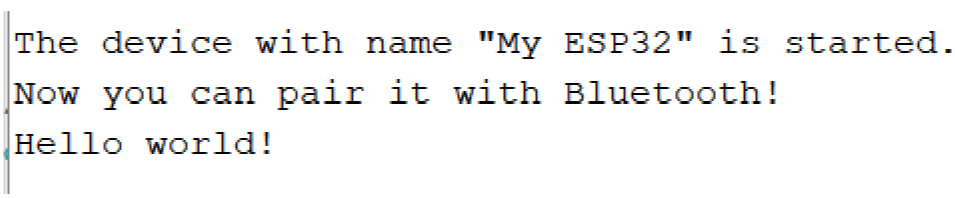


Figure 9.8 - Verification of Successful Message Delivery

The same message was sent from the serial terminal in Arduino, and was successfully received by the mobile phone. These results verify the ESP32’s functioning Bluetooth capability.

9.5 System Integration

Meeting the project’s goals requires the successful integration of hardware and software components. As shown in the overall system designs, the car’s motor and sensors will be connected to microcontrollers. The microcontrollers will read sensor data, perform calculations based on those readings, and respond accordingly to control the motor and display outputs.

9.6 Plan for Senior Design 2

With most projects, there will be inevitable setbacks that delay progress. For this reason, the group aims to take a proactive approach by working on the project as much as possible during the semester break and during the first few weeks of senior design 2. Electrathon races happen about once a month, which would theoretically give the team about 4 opportunities to “live test” the car between now and the end of senior design 2. Therefore, it is ideal to have a race-ready car as soon as possible so the team can test the car in a real race setting and make improvements based on the results.

The main priority will be accomplishing the project’s basic goals: get the car running, program the motor controller, and have a basic functioning display. From there, the team will focus on getting to Electrathon races for testing and work on accomplishing more advanced goals.

10 Administration

This section outlines the work breakdown, budget estimate, and milestones for the project. The work breakdown (10.1) shows the tasks each group member will be responsible for completing. The budget estimate (10.2) provides a list of materials needed, along with estimated quantity and cost of each component. Section 10.3 lists project milestones and when the group plans to complete them.

10.1 Work Breakdown

Julian will be responsible for the majority of the hardware design and physical work on the car, including spot welding cells for the battery pack and welding the car's frame. He will design the circuit and printed circuit boards for the motor controller and AC-DC charger in the main power system, as well as the bias supplies. For the report, he conducted extensive research for the hardware technology comparison and part selection (3.1 and 3.2), assisted with research for the software technology comparison and part selection (3.3 and 3.4), researched and documented engineering specifications and design constraints (Chapter 4), created figures for schematics and PCB layouts, and created a bill of materials for the project. Julian is also providing most of the funding for the project, as he intends to keep the car after the project is complete.

Abdullah will select sensors to gather information for the driver information system, such as accelerometer and GPS sensors, and he will integrate them with the control board. He will make a block diagram for this control board and perform the PCB layout in KiCad. For the battery pack, he will connect the cell holders together and solder leads, and in the AC-DC battery charger he will research and help to design the transformer and other magnetic elements. In the report he created the hardware block diagram, control board block diagram, figure of the work breakdown structure (figure 6.1), and the logic PCB layout. He researched and documented types of motors, batteries, transistors, and capacitors for the hardware technology comparison and selection sections of this report (3.1 and 3.2).

Grace will design and create a Bluetooth-connected mobile app for the driver display and program the central microcontroller it receives data from. Because "embedded is hard", Fouad and Grace plan to collaborate on programming for the central microcontroller and motor controller microcontroller, with Grace being primarily responsible for the central MCU and Fouad being primarily responsible for the motor controller MCU. For the report, Grace wrote most of the project description and created the software flowchart (Chapter 2), researched and documented choices of motor controller and display implementations (3.3), wrote about part selection for the motor controller's associated microcontroller (3.4.1), selection of an app development platform for the display (3.4.3), researched and ran demos related to the use of ChatGPT for the project (Chapter 5), and created test applications for the central microcontroller and display (Chapter 9). She also proofread the report, adding explanatory content as necessary, and created the list of project milestones.

Fouad will work with Grace to program the motor controller and central microcontrollers. For the report, he researched microcontroller options for the central MCU.

Figure 10.1 illustrates the list of tasks to be completed by each group member, with Julian and Abdullah (Electrical Engineering majors) focusing on the hardware design and implementation while Grace and Fouad (Computer Engineering majors) support the project's software needs.

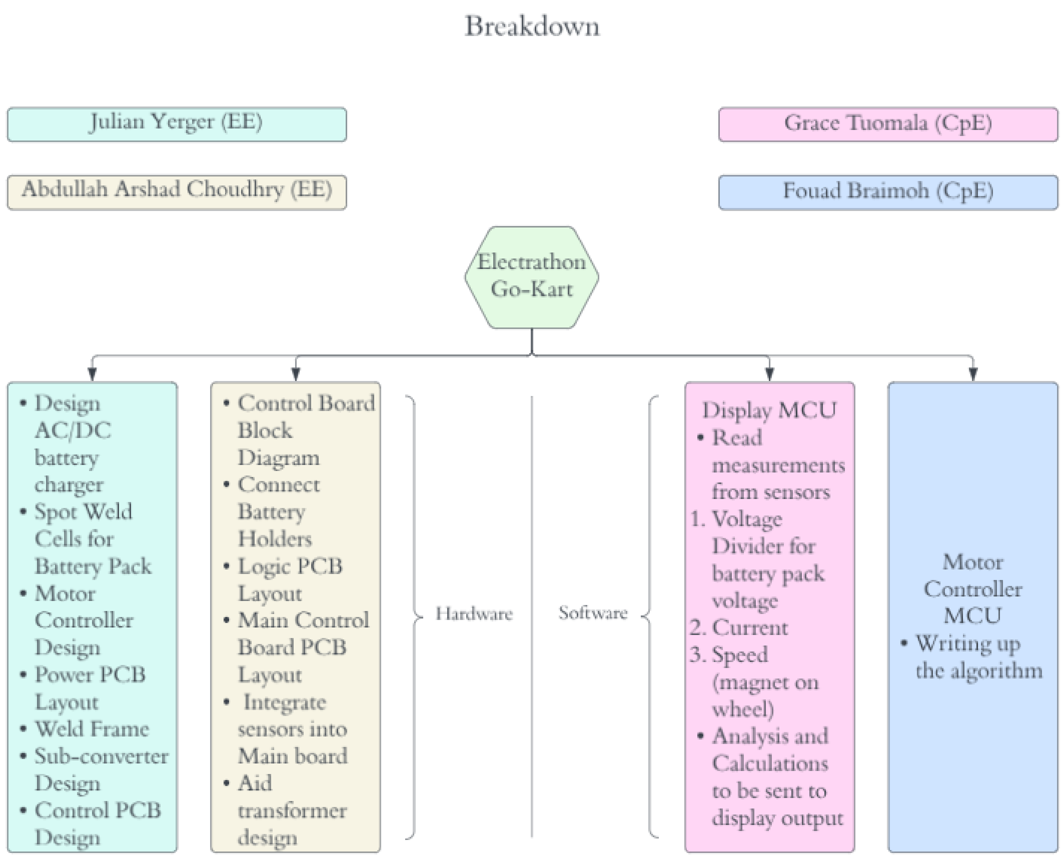


Figure 10.1 - Work Breakdown Structure

10.2 Budget Estimate

Table 10.1 provides a list of components needed for the project, along with the estimated quantities and cost. As previously mentioned, the project will be primarily funded by Julian.

Table 10.1 - Budget Estimate		
Item	**Quantity**	**Price Estimate**

Samsung 20S battery cell	140	$700
Busbars and cell holders	200	$30
Active cell balancer	1	$21
Hacker A60-18L motor	1	$272
Parts for motor controller	1	$300
Circuit breaker	1	$11
Wires and connectors	8	$50
Main microcontroller	1	$20
Sensors	3	$50
Parts for battery charger	1	$250
Total		$1,704

10.3 Project Milestones

Table 10.2 lists project milestones, along with when the group plans to accomplish them with respect to weeks in the semester.

Table 10.2 - Project Milestones

Task	**Planned Completion Week (SD1 and SD2)**
Complete Divide and Conquer/Project description	3-4
Install a basic power system in car	4-6
Select microcontroller and display to use	4-6
Begin programming display	4-6
Request approval for on-campus demo of car	4-6
Have a drivable vehicle for baseline performance testing	5-7
Test performance of various motors	5-7

Design and build custom motor controller	8-10
Working display showing current and average speed, average power consumption and time left in race	8-10
Complete report chapters 3, 4, 5	10-12
Race car in Electrathon competitions	Throughout
Component testing	12-14
Complete remaining report chapters	12-16
End-of-semester demo	14-16
Make improvements to electrical system	Throughout
Add features to display	Throughout
Final Testing	28-30

11 Project Summary

To conclude this report, we will be summing up all the inspiration for the project, the key points in the design and brainstorm aspect of the project, and our general plans moving forward into senior design 2 next semester. As mentioned earlier in the report, this project was decided upon because we wanted to create a new electric system for electric go karts that would help with battery consumption and general efficiency problems other electric go karts would face.

While our idea to create an electric power efficient go kart is not the first time it has been thought of or done in this field of engineering, we have been able to come up with new ideas that build on what others have done before and innovate new and different parts to increase the general efficiency of the electric go kart. We believe that what we envision to create next semester will be a great competitor in the electrathon races and our ideas will also be able to be implemented in commercial go kart tracks to create a better experience for both the producers and the consumers.

With the different components and ideas mentioned in the research paper, we have come up with solutions for many of the efficiency problems faced by electric go karts. From the microcontroller motor control system to the battery optimization system to the diver display system, all the major components will work in tandem to maximize the efficiency of the go karts power system. We have acquired and tested the functionality of most of the major parts to be put together for the project, over the winter break and early in the spring semester we plan to have a working demo of the display system to see how it fits in the whole project and what modifications, if any, will need to be made.

The process of designing and building a go kart, generally requires interdisciplinary engineers to work together and focus on what part of the go kart best suits each of their engineering disciplines. Our situation as displayed in the report is different in the fact that we are mainly working on the electrical and software side of the go kart design process, with the mechanical parts being previously done already. After researching the different components that we would need in the project namely the motors microcontroller, the display microcontroller, the motor, the battery pack, the display system in the car, the battery charger. We then compared the numerous versions of the individual products to find which one will provide both the most efficient results and be the best with the other parts of the project.

Throughout the research and design process, we came across a good amount of problems that required us to find alternative solutions and ideas to what we had originally planned to do. The fact that we did not design the go kart in its entirety also brought up some more problems that we had to work around and find a way to efficiently implement those solutions into the go karts system .

We have made some preparations in advance for senior design 2 with getting most of the parts and beginning to test them early. We have also started developing the application for the drivers display system and are working on having early prototypes, knowing that the main goal for the project as a whole is maximizing the efficiency of the go kart, we have agreed to prioritize that aspect in any decisions we make going forward in the project.

So far, this project has helped in improving our perspectives on working as a team and communicating better amongst peers to achieve success, we have also been allowed to not only put resources learned in class to good use, we have also learned the value of good researching etiquette, and using all the combined resources to accomplish our goals. We have also learned the importance of setting realistic goals and milestones and tracking our progress on the set goals and milestones to ensure we are aware of what needs to be done.

We would like to thank our advisory board for all the helpful feedback and constructive criticism that has been offered to us over the span of the project's developmental phase. We have heard their feedback and have integrated it into our project design and plan moving forward. To conclude this report, we have a good road map for senior design 2 and we hope to use the values and skills we learnt from senior design 1 to complete, and have a working demo of our project by the end of senior design 2.

This paragraph is written by Fouad Braimoh, I would like to apologize to my teammates for not doing my fair share of work in senior design 1, I acknowledge the facts that Grace and Julian did complete a majority of this 120 page paper and I have and a meeting with both them and the professor (Lei Wei) and I have promised to make senior design 2 my number 1 priority next semester and help my team succeed to the best if my ability.

This paragraph is written by Abdullah Arshad Choudhry. I acknowledge that the work that I put in for senior design 1 was insufficient and my group members had to cover up my shortcomings. I have met with both my team members and Dr.Lei Wei and have committed to doing better for senior design 2.

References

[1] "Home: America: The Official Electrathon America Site," Electrathon America. https://www.electrathonamerica.org/_files/ugd/032d61_b585e3874fcd4bb6a51e92c2bfb3aa5b.pdf (accessed Sep. 7, 2023).

[2] Wikimedia Foundation. (2023, August 10). *DC Motor*. Wikipedia. https://en.wikipedia.org/wiki/DC_motor

[3] Wikimedia Foundation. (2023c, August 15). *Brushed DC electric motor*. Wikipedia. https://en.wikipedia.org/wiki/Brushed_DC_electric_motor

[4] Wikimedia Foundation. (2023d, October 30). *Brushless DC electric motor*. Wikipedia. https://en.wikipedia.org/wiki/Brushless_DC_electric_motor

[5] Wikimedia Foundation. (2023d, September 15). *Alkaline Battery*. Wikipedia. https://en.wikipedia.org/wiki/Alkaline_battery

[6] Wikimedia Foundation. (2023h, November 3). *Nickel–metal hydride battery*. Wikipedia. https://en.wikipedia.org/wiki/Nickel%E2%80%93metal_hydride_battery

[7] Wikimedia Foundation. (2023h, November 1). *Lithium-Ion Battery*. Wikipedia. https://en.wikipedia.org/wiki/Lithium-ion_battery

[8] Gan vs sic: Silicon Carbide and gallium nitride compared. Arrow.com. (2020, January 22). https://www.arrow.com/en/research-and-events/articles/silicon-carbide-and-gallium-nitride-compared

[9] Tuite, D. (2009, February 19). Figure of merit: A refresher course | electronic design. Electronic Design. https://www.electronicdesign.com/technologies/power/article/21749890/figure-of-merit-a-refresher-course

[10] Infineon Technologies. (2023, June 20). Gan Systems releases highest power density 3kw LLC reference design for power applications. GaN Systems. https://gansystems.com/newsroom/highest-power-density-3kw-llc-reference-design/

[11] Walters, K., & Werner, B. (n.d.). Introduction to Schottky Rectifiers - Microsemi. Microsemi.com.

https://www.microsemi.com/document-portal/doc_view/14630-introduction-to-schottky-rectifiers

[12] Toshiba. (2014, September). SIC Schottky barrier diodes - Toshiba Electronic Devices & storage ... toshiba.semicon-storage.com/. https://toshiba.semicon-storage.com/info/catalog_en_20140912_ALQ00216.pdf?did=15510

[13] Styles, J. (2019, October). Common Misconceptions about the Body Diode. https://gansystems.com/. https://gansystems.com/wp-content/uploads/2020/01/Common-misconceptions-about-the-MOSFET-body-diode.pdf

[14] Microsemi PPG . (n.d.). Gallium Nitride (GaN) versus Silicon Carbide (SiC) In The High Frequency (RF) and Power Switching Applications. Digikey.com.

[15] Types: Capacitor guide. EEPower. (n.d.-h). https://eepower.com/capacitor-guide/types/#

[16] Ceramic capacitor: Types: Capacitor guide. EEPower. (n.d.-a). https://eepower.com/capacitor-guide/types/ceramic-capacitor/

[17] Electrolytic capacitor: Types: Capacitor guide. EEPower. (n.d.-b). https://eepower.com/capacitor-guide/types/electrolytic-capacitor/

[18] Polymer capacitor: Types: Capacitor guide. EEPower. (n.d.-e). https://eepower.com/capacitor-guide/types/polymer-capacitor/

[19] Tantalum capacitor: Types: Capacitor guide. EEPower. (n.d.-g). https://eepower.com/capacitor-guide/types/tantalum-capacitor/

[20] Mica capacitor: Types: Capacitor guide. EEPower. (n.d.-d). https://eepower.com/capacitor-guide/types/mica-capacitor/

[21] Film capacitor: Types: Capacitor guide. EEPower. (n.d.-c). https://eepower.com/capacitor-guide/types/film-capacitor/

[22] Pini, A. (2020, September 17). Fundamentals: Understand the characteristics of capacitor types to use them appropriately and safely. DigiKey. https://www.digikey.com/en/articles/fundamentals-understand-the-characteristics-of-capacitor-types

[23] Parsons, R. (2021, February 13). Basic BLDC (PMSM) efficiency and power loss estimations. Things in Motion. https://things-in-motion.blogspot.com/2019/03/basic-bldc-pmsm-efficiency-and-power.html

[24] toolman2, & circuit. (2012, September 5). Motor comparisons [ca120 et al..]. Endless Sphere DIY EV Forum. https://www.endless-sphere.com/sphere/threads/motor-comparisons-ca120-et-al.43306/page-2

[25] Battery Test-Review 18650 summary. (n.d.). https://lygte-info.dk/review/batteries2012/Common18650Summary%20UK.html

[26] *Samsung 20s - high drain 18650 li-ion cell's capacity test*. Samsung 20S - high drain 18650 Li-ion cell's capacity test. (n.d.). https://www.thunderheartreviews.com/2019/01/samsung-20s-high-drain-18650-li-ion.html

[27] *Headway LiFePO4 Battery*. evlithium limited. (n.d.). https://www.evlithium.com/Headway_Battery/

[28] Nurtriartono, A., Mukhlisin, A., Yuniarto, M. N., & Rijanto, E. (2019). Performance comparison of BLDC motor controllers designed based on trapezoidal commutation and Foc. AIP Conference Proceedings, 2187(1). https://doi.org/10.1063/1.5138363

[29] Lewin, C, "Field Oriented Control (FOC) - A Deep Dive", Performance Motion Devices Inc, https://www.pmdcorp.com/resources/type/articles/get/field-oriented-control-foc-a-deep-dive-article Accessed: Nov. 7, 2023.

[30] G. Tuomala and C. Ventura, "Product Guidance from a Texas Instruments Sales Representative," Sep. 20, 2023

[31] G. Tuomala and J. Krasnow, "Microcontrollers in Industry Applications," Sep. 19, 2023

[32] OpenAI, "GPT-3.5: A Large Language Model," OpenAI, [Online]. Available: https://www.openai.com/gpt-3.5. Accessed: Sep. 23, 2023.

[33] "Implementation of field-oriented control of induction motor using MATLAB and STM32F4," Wordpress, https://akpc806a.wordpress.com/2013/03/01/implementation-of-field-oriented-control-of-induction-motor-using-matlab-and-stm32f4/ Accessed: Sep. 24, 2023.

[34] B. C. Havens, "Motor Controller Microcontroller Performance Comparison," brianchavens.com, https://www.brianchavens.com/2018/09/20/motor-control-microcontroller-performance-comparison/ Accessed: Sep. 24, 2023.

[35] S. Colton, “Motor Controllers,” Shane Colton, https://scolton.blogspot.com/p/motor-controllers.html Accessed: Sep. 24, 2023.

[36] B. Akin and M. Bhardwaj, “Sensorless Field Oriented Control of 3-Phase Permanent Magnet Synchronous Motors,” Texas Instruments, https://www.ti.com/lit/an/sprabq3/sprabq3.pdf?ts=1697387570877&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252Fzh-tw%252FTMS320F28032 Accessed: Sep. 24, 2023.

[37] MathWorks, “Embedded Coder,” MathWorks Inc, https://www.mathworks.com/products/embedded-coder.html Accessed: Sep.25, 2023.

[38] MathWorks, “C2000 Microcontroller Blockset,” MathWorks Inc, https://www.mathworks.com/products/ti-c2000-microcontroller.html Accessed: Sep.25, 2023.

[39] MathWorks, “Motor Controller Blockset,” MathWorks Inc, https://www.mathworks.com/products/motor-control.html Accessed: Sep.25, 2023.

[40] Infineon, “PMSM FOC motor control software using XMC™,” Infineon Technologies, https://www.infineon.com/dgdl/Infineon-AP32370_PMSM_FOC_for_XMC1000_4000-ApplicationNotes-v01_06-EN.pdf?fileId=5546d4626bb628d7016be669c3eb7e60 Accessed Sep. 25, 2023.

[41] “Arduino Simple Field Oriented Control (FOC) Project,” SimpleFOCDocs, https://docs.simplefoc.com/ Accessed Sep. 27, 2023.

[42] Texas Instruments, "Field Oriented Control of Permanent Magnet Motors," Online video, YouTube, Aug. 29, 2012. [53:43]. Available: https://www.youtube.com/watch?v=cdiZUszYLiA. Accessed: Sep. 27, 2023.

[43] Texas Instruments, “TMS320F28335 data sheet, product information and support,” Texas Instruments, https://www.ti.com/product/TMS320F28335 Accessed Sep. 27, 2023.

[44] Texas Instruments, “TMS320F28334 data sheet, product information and support,” Texas Instruments, https://www.ti.com/product/TMS320F28034 Accessed Sep. 27, 2023.

[45] Texas Instruments, “TMS320F280049 data sheet, product information and support,” Texas Instruments, https://www.ti.com/product/TMS320F280049 Accessed Sep. 27, 2023.

[46] Mouser Electronics, “TMS320F28335ZJZS Texas Instruments,” Mouser Electronics, https://www.mouser.com/ProductDetail/Texas-Instruments/TMS320F28335ZJZS?qs=x1SoKP%252B681c0rWezKQfMHg%3D%3D Accessed Sep. 27, 2023.

[47] Mouser Electronics, “TMS320F28034PNTR Texas Instruments,” Mouser Electronics, https://www.mouser.com/ProductDetail/Texas-Instruments/TMS320F28034PNTR?qs=B6kkDfuK7%2FBqGXVBneIl4A%3D%3D Accessed Sep. 27, 2023.

[48] Mouser Electronics, “F280049CRSHSR Texas Instruments,” Mouser Electronics, https://www.mouser.com/ProductDetail/Texas-Instruments/F280049CRSHSR?qs=XeJtXLiO41TbBftqdZU8hQ%3D%3D Accessed Sep. 27, 2023.

[49] B. Akin, M. Bhardwaj, and J. Warriner, “Sensorless Field Oriented Control of 3-Phase Permanent Magnet Synchronous Motors,” Texas Instruments, https://www.ti.com/lit/an/spraby9/spraby9.pdf?ts=1695183129604 Accessed: Sep. 27, 2023.

[50] S. Lim, “Sensorless-FOC for PMSM With Single DC-Link Shunt,” Texas Instruments, https://www.ti.com/lit/an/spract7/spract7.pdf?ts=1697566359122&ref_url=https%253A%252F%252Fwww.google.com%252F Accessed: Sep. 27, 2023.

[51] Texas Instruments, “InstaSPIN™ solutions for designing three-phase motor control applications,” Texas Instruments, https://www.ti.com/lit/eb/sway030/sway030.pdf?ts=1697637712751&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FTMS320F280049C%253FkeyMatch%253DTMS320F280049 Accessed Sep. 27, 2023

[52] Arduino Uno REV3 Arduino Online Shop. Available at: https://store-usa.arduino.cc/products/arduino-uno-rev3?selectedStore=us

[53] Arduino Mega 2560 REV3 Arduino Online Shop. Available at: https://store-usa.arduino.cc/products/arduino-mega-2560-rev3?selectedStore=us

[54] Arduino Due Arduino Online Shop. Available at: https://store-usa.arduino.cc/products/arduino-due?selectedStore=us

[55] Raspberry Pi Buy A raspberry pi 5, Raspberry Pi. Available at: https://www.raspberrypi.com/products/raspberry-pi-5/

[56] ESP32 ESP32 Wi-Fi & Bluetooth MCU I Espressif Systems. Available at: https://www.espressif.com/en/products/socs/esp32

[57] STM32 32-bit ARM cortex mcus STMicroelectronics. Available at: https://www.st.com/en/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus.html

[58] PIC® mcus | microchip technology. Available at: https://www.microchip.com/en-us/products/microcontrollers-and-microprocessors/8-bit-mcus/pic-mcus

[59] 8-bit PIC® and AVR® microcontrollers | microchip technology. Available at: https://www.microchip.com/en-us/products/microcontrollers-and-microprocessors/8-bit-mcus

[60] Teensy® USB Development Board PJRC. Available at: https://www.pjrc.com/teensy/

[61] Beaglebone® Black BeagleBoard. Available at: https://www.beagleboard.org/boards/beaglebone-black

[62] Processors and microcontrollers NXP Semiconductors. Available at: https://www.nxp.com/products/processors-and-microcontrollers:MICROCONTROLLERS-AND-PROCESSORS#/

[63] MSP430 microcontrollers TI.com. Available at: https://www.ti.com/microcontrollers-mcus-processors/msp430-microcontrollers/overview.html

[64] Intel® Edison Compute Module (IOT) - product specifications (no date) Intel. Available at: https://www.intel.com/content/www/us/en/products/sku/84572/intel-edison-compute-module-iot/specifications.html

[65] Intel® Galileo Board - Product Specifications (no date) Intel. Available at: https://www.intel.com/content/www/us/en/products/sku/78919/intel-galileo-board/specifications.html

[66] Home odroid. Available at: https://wiki.odroid.com/odroid-c1/odroid-c1

[67] Photon Particle Photon. Available at: https://docs.particle.io/photon/

[68] Industries, A. Adafruit Feather M0 Basic Proto - ATSAMD21 cortex M0, adafruit industries blog RSS. Available at: https://www.adafruit.com/product/2772

[69] Eboka, P. (2022) Java vs JavaScript, Lighthouse Labs. Available at: https://www.lighthouselabs.ca/en/blog/java-vs-javascript#:~:text=While%20Java%20is%20an%20object,devices%20and%20back%2Dend%20applications.

[70] Rust. Rust Programming Language. (n.d.-a). https://www.rust-lang.org/

[71] Real Python. (2023, January 25). MicroPython: An intro to programming hardware in Python. https://realpython.com/micropython/

[72] Team, T. A. (n.d.). Software. Arduino. https://www.arduino.cc/en/software

[73] Get started. ESP. (n.d.). https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/

[74] J. Teel, "How to Develop a Mobile App That Communicates With Your Product Using Bluetooth," Predictable Designs, https://predictabledesigns.com/how-to-develop-a-mobile-app-that-communicates-with-your-product-using-bluetooth/ Accessed: Oct. 21, 2023

[75] “Difference Between Dart and Java,” Geeks for Geeks, https://www.geeksforgeeks.org/difference-between-dart-and-java/ Accessed Oct 21, 2023.

[76] T. Woo, “ESP32 BLE + Android + Arduino IDE = AWESOME,” Instructables, https://www.instructables.com/ESP32-BLE-Android-App-Arduino-IDE-AWESOME/ Accessed Oct 21, 2023.

[77] C. Drawbridge, “Building a live data feed BLE application with React Native and ESP32,” Medium, https://blog.octoco.ltd/building-a-live-data-feed-ble-application-with-react-native-and-esp32-6815b8c6dd43 Accessed Oct 21, 2023.

[78] Flutter Karma, “FLUTTER | ESP32 | BLE- Weather Station using DHT11 sensor,” Medium, https://medium.com/@flutterkarma/flutter-esp32-ble-weather-station-using-dht11-sensor-80ca013f6495 Accessed Oct 21, 2023.

[79] V. Karunarathna, “UART Communication between Flutter and ESP32 using BLE,” Medium, https://medium.com/@viduraerandika/uart-communication-between-flutter-and-esp32-using-ble-4f0e36623523 Accessed Oct 21, 2023.

[80] Bluepixel Technologies, “BLE Scanner (Connect & Notify),” Google Play, https://play.google.com/store/search?q=ble%20scanner&c=apps&hl=en_US&gl=US Accessed Oct. 21, 2023.

[81] Code Origin Apps, “Timer Stopwatch App - Sound,” Google Play, https://play.google.com/store/search?q=ble%20scanner&c=apps&hl=en_US&gl=US Accessed Oct. 21, 2023.

[82] State Farm Insurance, "Drive Safe & Save," Google Play, https://play.google.com/store/search?q=state+farm+drive+safe+and+save+app&c=apps&hl=en_US&gl=US Accessed Oct. 21, 2023.

[83] Driversnote, "Mileage Tracker," Google Play, https://play.google.com/store/apps/details?id=com.driversnote.driversnote&hl=en_US&gl=US Accessed Oct. 21, 2023.

[84] A. Spiess, "BLE with ESP32 tutorial part 1: the Server," Online video, YouTube, Jan. 29, 2023. [25:22]. Available: https://www.youtube.com/watch?v=0Yvd_k0hbVs Accessed: Oct 21, 2023.

[85] T. Galbraith, "ESP32 Step Tracker Tutorial React Native & Bluetooth," Online video, YouTube, Jul. 3, 2023. [38:16]. Available: https://www.youtube.com/watch?v=ssokvToiYU0 Accessed: Oct 21, 2023.

[86] Uniquelymade, "React Native with ESP32 Arduino | Bluetooth Low Energy BLE | Example Program," Online video, YouTube, Jul. 3, 2023. [1:59]. Available: https://www.youtube.com/watch?v=erWibryA_tE Accessed: Oct 21, 2023.

[87] Benjineer.io, "Easy IOT – Remotely Controlling ESP32 Using an Android App," Instructables, https://www.instructables.com/Easy-IOT-Remotely-Controlling-ESP32-Using-an-Andro-1/ Accessed: Oct 21, 2023.

[88] A. Spiess, "Create your own App! Control an ESP32 (Arduino) via Bluetooth - Part 1," Online video, YouTube, Apr. 26, 2021. [18:17]. Available: https://www.youtube.com/watch?v=aM2ktMKAunw Accessed: Oct 21, 2023.

[89] Landete, A. M. (2022, July 28). JTAG. Read The Docs. https://readthedocs.web.cern.ch/display/MTA/JTAG

[90] ARM Developer. (n.d.). Cortex Debug (10-pin). developer.arm.com. https://developer.arm.com/documentation/101416/0100/Hardware-Description/Target-Interfaces/Cortex-Debug--10-pin-

[91] Valdez, J., & Becker, J. (2015, June). Understanding the I2C Bus. Application Report. https://www.ti.com/lit/an/slva704/slva704.pdf

[92] Tanenbaum, A. S. (n.d.). Quote by Andrew S. Tanenbaum. Goodreads. https://www.goodreads.com/quotes/589703-the-good-thing-about-standards-is-that-there-are-so

[93] IC package types: DIP, SMD, QFP, BGA, SOP, Sot, SOIC. Electronics For You. (2022, December 13). https://www.electronicsforu.com/resources/dip-smd-qfp-bga-ic-packages

[94] Brateng, E. (2019, December). Standard package sizing for two-terminal smds. Tech Forum. https://forum.digikey.com/t/standard-package-sizing-for-two-terminal-smds/5104

[95] Power, S. (2018). EMI Standards Applications & Limits EN55011 • EN55022 • EN55015 • EN55032. Application Note. https://slpower.com/App-slpower/images/whitepapers/AN-G006_EN55XXX_AppNote.pdf

[96] Hegarty, T. (2018). An overview of conducted EMI specifications for power supplies. Texas Instruments. https://www.ti.com/lit/wp/slyy136/slyy136.pdf

[97] *Electromagnetic compatibility (EMC) – Part 3-2: Limits for harmonic current emissions*, IEC 61000-3-2, 2018

[98] *IEEE Recommended Practice for Electronic Power Subsystems: Parameters, Interfaces, Elements, and Performance*, IEEE 1573, 2021

[99] *Audio/video, information and communication technology equipment - Part 1: Safety requirements*, IEC 62368-1, 2023

[100] Buck, D. (2022, November). What is the difference between class I and class II power supplies? - TDK. Lambda.tdk. https://www.emea.lambda.tdk.com/il/KB/What-is-the-difference-between-Class-I-and-Class-II-power-supplies.pdf

[101] PCB Solutions, C. (2023, October 12). Designing to high voltage creepage and clearance standards in circuit board layouts. https://resources.pcb.cadence.com/blog/2022-designing-to-high-voltage-creepage-and-clearance-standards-in-circuit-board-layouts

[102] *National Electrical Code 2023 Handbook*, NEC, Crystal Lake, IL, 2023.

[103] The Scubersteve, “Intro to the Budget-Beater Electrathon car,” Online video, YouTube, Aug. 14, 2021. [7:41]. Available: https://www.youtube.com/watch?v=9K9XF5D23fc Accessed: Oct 25, 2023

[104] Archer, S. (2021, August 8). Cheapest possible Electrathon car. electrathonamerica.activeboard.com. https://electrathonamerica.activeboard.com/mobile.spark?p=topic&topic=67476766

[105] Texas Instruments. (2018, February). Power Stage Designer User’s Guide. TI.com. https://www.ti.com/lit/ug/slvubb4b/slvubb4b.pdf

[106] Dimonis, D. (2012, March 20). SMPS transformer design tool (Ver.4000). diyAudio. https://www.diyaudio.com/community/threads/smps-transformer-design-tool-ver-4000.209221/

[107] Texas Instruments. (n.d.). WEBENCH® Power Designer. webench.ti.com. https://webench.ti.com/power-designer/switching-regulator

[108] 1BitSquared. (n.d.). KiCad EDA: A Cross Platform and Open Source Electronics Design Automation Suite. kicad.org. https://www.kicad.org/

[109] Schweickhardt, J., & Hermanns, K. (2021, March). Tips & Tricks on Double Pulse Testing. Application Note. https://scdn.rohde-schwarz.com/ur/pws/dl_downloads/dl_application/application_notes/gfm347/GFM347_1e_Double_Pulse_Testing.pdf

[110] Samsung, S. (2014, November 27). Specification of product - imrbatteries.com. IMR Batteries. https://www.imrbatteries.com/content/samsung_20S.pdf

[111] Shine, I. (2022, July 20). The world needs 2 billion electric vehicles to get to net zero. but is there enough lithium to make all the batteries?. World Economic Forum. https://www.weforum.org/agenda/2022/07/electric-vehicles-world-enough-lithium-resources/

[112] Ahmad, M. (2023, April 2). GaN in an EV inverter? Here is a prototype. EDN. https://www.edn.com/gan-in-ev-inverter-here-is-a-prototype/

[113] Korishan. (2022, October 12). Different cutoff voltages. Second Life Storage & Solar. https://secondlifestorage.com/index.php?threads%2Fdifferent-cutoff-voltages.11967%2F

[114] Battery University. (2023, October 11). BU-808: How to Prolong Lithium-Based Batteries. https://batteryuniversity.com/article/bu-808-how-to-prolong-lithium-based-batteries

[115] Hu, W. (2023, June 21). How e-bike battery fires became a deadly crisis in new york city. New York Times. https://www.nytimes.com/2023/06/21/nyregion/e-bike-lithium-battery-fires-nyc.html

[116] Bhargava, C., JMS, & Andrea, D. (2016, February 25). Is draining a li-ion to 2.5 V harmful to a Li-ion cell?. Electrical Engineering Stack Exchange.

https://electronics.stackexchange.com/questions/219222/is-draining-a-li-ion-to-2-5-v-harmful-to-a-li-ion-cell

[117] matthewwl950, CrashOne, Enderman, & BlueSwordM. (2018, June 26). How critical is the 2.5V cut-off voltage for Li-ion battery life?. BudgetLightForum.com. https://budgetlightforum.com/t/how-critical-is-the-2-5v-cut-off-voltage-for-li-ion-battery-life/51929/3

[118] Battery University. (2022, March 3). BU-303: Confusion with voltages. https://batteryuniversity.com/article/bu-303-confusion-with-voltages

[119] DW01A Datasheet. H&M Semi. (n.d.). https://hmsemi.com/downfile/DW01A.PDF

[120] HY2213 Datasheet. HYCON Technology. (n.d.). https://www.hycontek.com/wp-content/uploads/DS-HY2213_EN.pdf

[121] Vedder, B. (2016, November 13). VESC 6 MK5. VESC Project. https://vesc-project.com/sites/default/files/Benjamin%20Posts/VESC_6_mk5.pdf

[122] Battery University. (2021, October 25). BU-409: Charging Lithium-Ion. https://batteryuniversity.com/article/bu-409-charging-lithium-ion

[123] Lu, B. (n.d.). Introduction to EMI in Power Supply Designs. Texas Instruments. https://www.ti.com/lit/pdf/SLYP757

[124] Cadence PCB Solutions. (2022, October 13). Passive PI Filter Design and Simulation. Cadence PCB Design and Analysis. https://resources.pcb.cadence.com/blog/2020-passive-pi-filter-design-and-simulation

[125] Jurgenh. (2018, April 27). The DC-DC boost converter, part 2 – power supply design tutorial section 5-2. Power Electronics News. https://www.powerelectronicsnews.com/the-dc-dc-boost-converter-part-2-power-supply-design-tutorial-section-5-2/

[126] UCC28070 Interleaving Continuous Conduction Mode PFC Controller. Texas Instruments. (2016, April). https://www.ti.com/lit/ds/symlink/ucc28070.pdf

[127] Boost Converter Loss Analysis. University of Tennessee Knoxville. (n.d.). https://web.eecs.utk.edu/~dcostine/ECE482/archivedRecordings/LossOverviewSlides.pdf

[128] Rohm Semiconductor. (2022, April). TO252 Package Thermal Resistance Information. Application Note.

https://fscdn.rohm.com/en/products/databook/applinote/ic/power/linear_regulator/to252_thermal_resistance_information_an-e.pdf

[129] Ivanov, P. A., & Levinshtein, M. E. (2021). The impact of parasitic inductance on the DV/DT ruggedness of 4H-sic Schottky diodes. Microelectronics Reliability, 122, 114159. https://doi.org/10.1016/j.microrel.2021.114159

[130] Wang, G., & Brunt, E. V. (2017). On Developing a dV/dt Rating for Commercial 650 V- and 1200 V-Rated SiC Schottky Diodes. PCIM Europe. https://ieeexplore.ieee.org/abstract/document/7990786

[131] Damped Oscillations in RLC Series Circuit. Physics Key. (n.d.). https://www.physicskey.com/rlc-circuit-damping

[132] Last Minute Engineers. (2022, May 15). The Full-Wave Rectifier. Last Minute Engineers. https://lastminuteengineers.com/the-full-wave-rectifier/

[133] Last Minute Engineers. (2022a, May 15). The full-wave bridge rectifier. Last Minute Engineers. https://lastminuteengineers.com/the-full-wave-bridge-rectifier/

[134] Math Keeps Me Busy. (2022, September 19). What technical issues need to be considered when choosing between full wave (two diodes) or bridge rectifier (four diodes)?. Electrical Engineering Stack Exchange. https://electronics.stackexchange.com/questions/635398/what-technical-issues-need-to-be-considered-when-choosing-between-full-wave-two

[135] Williams, T. (2022, August 20). Voltage/current rating for center-tapped rectifier vs. full-bridge rectifier. Electrical Engineering Stack Exchange. https://electronics.stackexchange.com/questions/631782/voltage-current-rating-for-center-tapped-rectifier-vs-full-bridge-rectifier

[136] Dixon, L. H. (2001). Magnetics Design Handbook. Texas Instruments. https://www.ti.com/seclit/ml/slup132/slup132.pdf

[137] Nategh, S., Krings, A., Wallmark, O., & Leksell, M. (2014). Evaluation of impregnation materials for thermal management of liquid-cooled electric machines. IEEE Transactions on Industrial Electronics, 61(11), 5956–5965. https://doi.org/10.1109/tie.2014.2308151

[138] Simpson, N., Wrobel, R., & Mellor, P. H. (2013). Estimation of equivalent thermal parameters of impregnated electrical windings. IEEE Transactions on Industry Applications, 49(6), 2505–2515. https://doi.org/10.1109/tia.2013.2263271

[139] Biela, J., & Kolar, J. W. (n.d.). Cooling Concepts for High Power Density Magnetic Devices. Power Electronic Systems Laboratory, ETH Zurich. https://www.pes-publications.ee.ethz.ch/uploads/tx_ethpublications/biela_PCC07.pdf

[140] Hard Commutation of Power MOSFET. Infineon Technologies. (2014, March). https://www.infineon.com/dgdl/Infineon-Power_MOSFET_OptiMOS_FD_200V-250V_hard_diode_commutation-AN-v01_00-EN.pdf?fileId=db3a304344ae06150144b1d2f8250165

[141] Ridley, R. (2006, December). Gate Drive Design Tips. imserv.org. https://imserv.org/MFKForum/Datasheets/gate%20drive/gate%20drive%20design%20tips.pdf

[142] Shepard, J. (2021, September 9). Gate Drive Transformer Specifications and Applications. Power Electronic Tips. https://www.powerelectronictips.com/gate-drive-transformer-specifications-and-applications-faq/

[143] UCC28950 Green Phase-Shifted Full-Bridge Controller with Synchronous Rectification. Texas Instruments. (2016b, November). https://www.ti.com/lit/ds/symlink/ucc28950.pdf

Copyrights

Samsung 20S - high drain 18650 Li-ion cell's capacity test Inbox ×

Abdullah Arshad Choudhry <abdullaharsh99@gmail.com>
to contact

Mon, Oct 30, 1:03 PM

Good Afternoon,

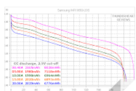
I wanted to get in touch with you in order to obtain permission to use one of your graphs from your reviews. Me and my team are students at the University of Central Florida and are writing our senior design report. In that report we were talking about different types of batteries we could use and we wanted to use a graph you created in your review. I have attached the image below and will also add a link to the review I am talking about.

Kindly let me know if you can grant us the permission to use it. Hoping to hear from you soon.

https://www.thunderheartreviews.com/2019/01/samsung-20s-high-drain-18650-li-ion.html

Thank you,
Abdullah Arshad Choudhry

One attachment • Scanned by Gmail



Narek Avetisyan <n.g.avetisyan@mail.ru>
to me

Sat, Nov 11, 4:11 AM

Hi Abdullah.
Feel free to use materials from my tests in educational purposes with mentioning the author and source.

Best regards
Narek Avetisyan aka Thunderheart

> Monday, 30 October 2023, 20:03 +03:00 from Abdullah Arshad Choudhry <abdullaharsh99@gmail.com>:
