Shell Eco-Marathon Motor Controller

Kevin Fitch, Olivia Janousek, Nick Oldenburg

Advisor: Noureen Sajid

Senior Design Project 2023

Bachelor of Electrical Engineering

Dunwoody Technical College

December 15th, 2023

[Acknowledgements 3](#_Toc153746486)

[Introduction 3](#_Toc153746487)

[Background 3](#_Toc153746488)

[Shell Eco-Car Background 3](#_Toc153746489)

[VESC Background 3](#_Toc153746490)

[Issues with Current Design 3](#_Toc153746491)

[Capacitors 3](#_Toc153746492)

[Connectors 4](#_Toc153746493)

[Research 4](#_Toc153746494)

[Connector Replacement 4](#_Toc153746495)

[Capacitor Failure Mode 5](#_Toc153746496)

[Connector Test Board 6](#_Toc153746497)

[Motor Controller Design 7](#_Toc153746498)

[Conclusion 11](#_Toc153746499)

[Future Work 11](#_Toc153746500)

[Bibliography 12](#_Toc153746501)

# Acknowledgements

We would like to thank Noureen Sajid, Adrian Durand, and the entire Dunwoody Daredevils Shell Eco-marathon team. Additionally, this project would never have been realized without the efforts of the Cheap FOCer and VESC open-source communities.

# Introduction

Our goal for this project was to construct a Printed Circuit Board Assembly (PCBA) manufacturing file for a motor controller that is to be used in Dunwoody’s Shell Eco-marathon car. To accomplish this the design used in last year’s competition was revised to eliminate specific issues experienced by last year’s team, as well as to add a few improvements based on team feedback. The key to generating a new motor controller was understanding both the existing board and its shortcomings. Our end-product is motor controller manufacturing files for use in the spring 2024 Shell Eco-Marathon Competition.

# Background

## Shell Eco-Car Background

The Shell Eco-Marathon is an annual competition where teams from high schools and universities compete on who can build the most efficient vehicle of various fuel types. We joined Dunwoody’s Electric Prototype Vehicle team, the Daredevils, in improving their existing motor controller. The competition has certain requirements for all aspects of the vehicle, including the motor controller. Most of these requirements are focused on safety. The most important rule regarding the motor controller is that it cannot be pre-purchased. It can be made of purchased pieces, or, if a PCB is used, the silkscreen mask needs to include the letters SEM. The motor controller must also be fully documented and work to drive the motor.

## VESC Background

VESC, or Vedder Electronic Speed Controller, was created by Benjamin Vedder. One important thing to understand about VESC is that it is not our actual speed controller. It is actually the firmware that runs our electronic speed controller, the Cheap FOCer. There are many controllers out there that run VESC, and they are all considered “VESC based”. As an open-source project, VESC is accessible to anyone and can be customized and configured to any need. On top of that, because of it being an active project, it has gained a large community that all works together to better understand and iterate this product.

## Issues with Current Design

### Capacitors

The most obvious issue we were asked to investigate was the failure of a capacitor. On one of the boards, one of the capacitors exploded, resulting in a useless board. There were a few proposed ways that this capacitor could have exploded, and one of our jobs was to determine what caused the capacitor to fail and what we could do to fix it. One considered issue was the level of stress increasing from that of the intended application of the base board. As the original motor controller was used to drive lightweight electric skateboards, we thought perhaps using the same board to drive a much heavier vehicle that needed much more torque to start caused issues with the capacitors.

### Connectors

A less obvious issue was with the connectors. These were again the suggested connectors used in the original application of the board, which would not have nearly as much room to move and fall out as well as be under considerably less vibration. The connectors had a tendency to come loose while driving, so had to be replaced. This was a simpler issue to consider than the capacitors, as the automotive industry typically has stringent connection requirements that we could use as a guideline for selecting what connectors to use.

# Research

## Connector Replacement

There are hundreds of thousands of connectors available on Digikey. To slim down the selection, a series of requirements had to be established. The easiest one to determine was the number of positions available on a module. There are many sizes of single row connectors on the board already, and ideally the Shell Eco-marathon team would be able to use the same series of connectors from one brand for most of these sockets. This means the type of connector we use must be available in multiple form factors, limiting the options to larger companies that have such a selection. Since we are aiming to fix the previous issues native to the original design, the connector must also be latching in some way. A latching connector, be it through screw terminals or crimped connections, will significantly reduce the effect of vibrations. We tested connectors of both types, and ultimately decided to use a clamping crimped mating connector. This type of connector would allow the team to change boards at a moment’s notice without having to rewire the entire car, without sacrificing the robustness needed. The last thing to keep in mind was that the connector module must be made with the intent of being used in the automotive industry. While not an official requirement, something designed for use in vehicles would give the team additional peace of mind against accidental disconnect. The MOLEX MicroClamp connector was ultimately the one that was chosen as it fit the stated requirements. It had the added benefit of having the same pitch as the original connectors, meaning the new selection could even be swapped into the old boards. New connectors can be seen in Figure 1.

#### Figure 1 – MOLEX Through-Hole Connector with Clasp

A white plastic connector with holes

Description automatically generated A white electrical connector with metal tips

Description automatically generated with medium confidence

## Capacitor Failure Mode

There are multiple reasons for an electrolytic capacitor to fail during operation. The most common of these failure modes are an inrush of current, overvoltage, abrupt temperature rise, or reverse polarity. For our motor controller’s situation, the problematic capacitors were the three large 63v 330 µf electrolytic capacitors on the bottom of the board. We knew that the capacitor in our situation only failed in one run during normal operation. This led us to believe that it was less likely to be from too much current or too much voltage, as our battery capacity was about 48 volts. A rise in temperature is possible, as a long duration of usage could raise the temperature of the board and the capacitors. This is also less likely due to it only happening once and the number of precautions that were taken to avoid overheating.

We decided it would be necessary to conduct some tests to eliminate some of these possibilities. For our tests, we chose capacitors of a lower voltage rating than the capacitors currently used in the motor controller for safety reasons. Smaller capacitors were used additionally because of the voltage supplies we had available; none of these were of a high enough voltage to overload the high-rated capacitors used on the board. The two main ways we tested the capacitors were through overvoltage and through reverse polarity. While we got destructive results with reverse polarity in all cases, no test of overvoltage resulted in a destroyed capacitor. When replicating this, we got the same outcome that was seen on the broken board. Once the capacitor is charged while being used with reverse polarity it explodes, causing it to vent.

Though the explosive nature of this failure mode was like the one seen on the board, this is not the only way for a capacitor to end up in that state. While conducting additional research on the specific board we were modifying, we came across some information that leads us to believe this result was obtained in a different way. The parasitic inductance on the MOSFETs prevents an instantaneous current change on the half bridge driver, meaning that as soon as the gate is closed, the kickback created by said inductance would create a voltage spike potentially exceeding the voltage rating of the capacitors. Over time, the leakage current lost to this phenomenon would effectively boil the electrolytic filler of the capacitor, eventually causing it to explode or vent, ruining the capacitor. This prosses can be seen in Figure 2. Resistance of MOSFET driving circuit was increased to reduce potential overvoltage states.

#### Figure 2 – Capacitor Overvoltage Due to Inductive Kickback

A screen shot of a computer

Description automatically generated

## Connector Test Board

Between the footprint of the components one hopes to incorporate to the actual stack-up copper and resin the manufacturing facility can produce, there is a lot that goes into PCB design. As this can be quite a daunting process, and our PCB design software was not yet selected, we decided to create a test board. Opting to kill two birds with one stone, the connector footprints where the ones chosen for prototyping. This allowed us to verify that the custom footprints would work when incorporated into the larger design, where mistakes could potentially be much more costly, while simultaneously learning the basics of PCB design.

Two PCB drafting programs were chosen as contenders. One, KiCad, which is a well-established open-source PCB Design Software, and the other, CircuitMaker, which is the free version of the industry standard Altium. The first footprints were made in CircuitMaker, which proved to be quite clunky and hard to navigate. Often a footprint would be created just for it to disappear the next time the project was opened. Between its complex project structure and oversimplified UI, CircuitMaker really did feel like the free version of a much more powerful program. This led us to KiCad. Fortunately, KiCad’s open-source nature in tandem with its wide userbase has resulted in a much more capable software. Many tools excluded from the aforementioned “lite” version of Altium were included natively with KiCad. This resulted in a much more streamlined development of the connector test board.

The ability to get familiarized with the PCB Drafting software before diving headfirst into the exceptionally complex motor controller would prove to be invaluable. Everything from file formats to organization of the project documents, to how to order a PCB from an actual manufacturer was clarified without interference from the already established KiCad Files of the Cheap FOCer. Having a physical PCB with which to test the prospective connectors would also prove to be a boon. After a software had been selected and our ability to edit/create rudimentary PCBs had been vetted, it was time to tackle the motor controller. The final connector board iteration can be seen in Figure 3.

#### Figure 3 – Connector Board PCB Ordering Pilot

A green rectangular object with white text

Description automatically generatedA green rectangular board with white text and yellow holes

Description automatically generated

# Motor Controller Design

Initially, the files available for the Cheap FOCer online left much to be desired. The first step was to decipher what was needed to start a new iteration, and what was a product of the bloat due to the open-source nature of the project. This required an extensive dive into the filing system of KiCad. After a clean project folder had been created with all the files needed to get the project to launch, the work began.

On first start-up, the PCB and schematic were riddled with errors and warnings (well over one thousand between the two). There were two ways we could fix this: the project could either be unraveled knot by knot, or we could start from a clean slate. The schematic had to be kept, as that served as the true foundation of the project. It was responsible for the source information of the PCB, bill of materials, and functioning principles of the system as a whole, so not much could be safely scrapped. The root schematic is a multilevel project file that contains the connections for every sub-schematic associated with the project. Each submodule requires different inputs and outputs, but when stitched together creates the template for the PCB. The root schematic can be seen in Figure 4.

#### Figure 4 – Original “Cheap FOCer” Root Schematic and Sub-Modules

A computer screen shot of a diagram

Description automatically generated

As there were many errors throughout the entire project file, we decided to cut what was not needed for the Shell Eco-marathon car. This would save on wasted effort, as we would not be correcting errors in systems that would never see use in our application. Several modules were cut. The first to go was the CAN module, used for communication between multiple motor controllers. This would not be used as the current car only utilizes a single BLDC motor. The next was the IMU, or Inertial Measurement Unit. As VESC-based boards such as the Cheap FOCer are commonly used in one-wheeled skateboard designs, this is normally an important piece for balance and control. While it is not currently used, this module could potentially be useful to the team in the future to measure distance and speed, so connections to add the module in were added, while the module itself was cut.

After the project had been pruned, the schematic was debugged to have a healthy transition to physical PCB. After diving into the schematic rule checker, many rule violations, or errors, displayed similar symptoms. Many custom components were missing their schematic symbol or footprint. In previous efforts to cull the projects file bloat, the custom libraries had been left behind. Once linkage had been reestablished and a schematic rebuild, there was a significant reduction in rule violations. However, the remaining errors, around 20% of what there was previously, had to be dealt with individually. This mostly consisted of unconnected pins, updating symbols to cohere to the updated KiCad 7’s rule set, creating custom missing symbols, reestablishing off-grid connections, and tying nets of different class names together. After all the previous work had been verified by KiCad’s internal check, it was time to start adding things desired by the Shell Eco-Marathon team, namely USB-C capability, and Bluetooth module connector; not to mention the new connectors. The new additions required modifications to the root schematic and introduction on multiple custom footprints. Once properly validated against the rule set, it was time to start physically placing the components onto a new PCB.

This was an intensive process, requiring the bulk of the project time. All this portion required was the connection of one pad to another pad of the same net, but in practice, many nets from opposite side of the board need to be connected in multiple spots with exceptionally limited area. To arbitrarily place anything would be a grave mistake, one made often over the course of this project. The sheer number of connections was overwhelming, so the decision was made to trace the PCB one schematic page at a time. The microcontroller was the first schematic page to be put to copper. Resistors and capacitors were placed close to pins to mitigate EMC and overall used space. The process was repeated module by module, save for the MOSFET drivers and the connectors, as the MOSFET drivers would require additional heatsink space and interconnectivity, and the connector placement was assumed to just fill up whatever space was left at the end. Once each room was laid out, they were positioned close to their interconnected module. This gave us a general idea of what the board would look like without the addition of power circuitry and interface circuitry. While leaving the logic and drivers/connectors separate aided in initial layout, this would prove to cause many problems for us down the road. First, the stitching of each preassembled module was tackled.

The two most connected submodules, the MCU and the MOSFET driving logic IC, were placed directly next to each other, as this would allow for the shortest paths between connection; however, it was immediately clear that laying out each module before bringing them together was a mistake. The capacitors and resistors required to support each module created an effective blockade between modules, with limited space remaining for interconnection. This problem only compounded with each additional submodule that was added. Much time was lost rearranging components countless times, just for a little more space to run a minuscule trace through. It is important to acknowledge that each length of copper and via had to be hand place, meaning that backtracking was an exceptionally expensive endeavor. Eventually, all submodules were stitched together in a compact manor. It was already abundantly clear that there was not much room left for the power circuitry, but the side of the MOSFET power driving IC was left open specifically for this purpose.

Once only the power circuitry and connectors remained, it was time to decide the placement of the power MOSFETs and their support circuitry. As each MOSFET required an abundance of space, it was decided to add one of each of the three phases to adjacent edges of the driver IC. This left each MOSFET with a healthy amount of space between them, and a clear route to connect them, or so it was thought. The first thing that was noticed once each phase of the power modules had been connected to their respective supporting circuitry was that the third current sense amplifier (used to control the third motor phase) and the IC pins used for the third phase were on opposite sides of the board. This was a quintessential example of the PCB design experience for this project. Two options remained: redesign most of the previous work, leaving us just shy of a total redesign, or move the third phase to be adjacent to the current sense circuitry, swapping it with the first phase and the weaving the control signals in such a way that they would not interfere with each other. We went with the latter option in the interest of time, but a valuable lesson in foresight was learned. From this hiccup on the layout was relatively simple, until it came to the final phase. When it came to finishing the connector placement, all the real estate around the MCU had been occupied, top and bottom. The design had to be optimized trace by trace in order to open more space. The endgame of the design had quickly become quite dense. While a dense board makes the most of its space, it actively diminishes thermal dissipation, especially on the higher current traces. To combat this issue, the board will have to be ordered with the thickest copper layer option from the manufacturer.

Once all nets had been connected and the components properly positioned, it was time to clean up the silkscreen and recut the board edges. This could have been a week all to itself, but unfortunately the time for optimization was over. We could not let perfect be the enemy of good. The final shape was set, the connector pins were labeled, and the board was complete. The final product can be seen in Figure 5.

#### Figure 5 – Final Board Design Drawing and 3D Rendering

A computer chip with a red and white background

Description automatically generated with medium confidence

A computer circuit board with many small circles and small holes

Description automatically generated with medium confidence

A green circuit board with many small components

Description automatically generated

A green circuit board with white and gold elements

Description automatically generated

# Conclusion

In the end, much was learned about PCB design, and even more about motor controllers. Trying to retroactively analyze the failure mode of a foreign PCB is a fantastic way to learn about its inner workings. Forming hypothesis after hypothesis only to disprove most of them also compounded our understanding. In the end, we can only hope that what we produced through this project will benefit the Shell Eco-Marathon team and the constituent schools for years to come. No matter what though, we will carry the time management and technical research aspects of this project with us throughout our professional careers.

# Future Work

The board will still have to be ordered and tested, but even before then there are many things that could have been further improved. From better labeling to heat sink add-on traces, time will tell where this new design falls short. Documentation is another large step that will be especially important to the Shell Eco-Marathon team. Cultivating a GitHub for the project, along with all the necessary information to utilize it, will likely be the most important final piece of the puzzle. All our work will be for naught if the information behind the project is lost come next semester.

# Bibliography

Analog Devices. (n.d.). *Field Oriented Control (FOC) as a Hardware Building Block*. Retrieved from Analog Devices: https://www.analog.com/en/products/landing-pages/001/field-oriented-control.html#:~:text=FOC%20uses%20orthogonal%20applied%20current,leading%20applications%20with%20synchronous%20drives.

Garage, T. B. (2022, Jul 19). *What is VESC? A Primer for Onewheel & Eskate People*. Retrieved from Youtube: https://www.youtube.com/watch?v=gaHIFx08fO8&ab\_channel=TheBoardGarage

Lewin, C. (n.d.). *Field Oriented Control (FOC) - A Deep Dive*. Retrieved from Performance Motion Devices: https://www.pmdcorp.com/resources/type/articles/get/field-oriented-control-foc-a-deep-dive-article

liveforphysics. (2010, Oct 15). *Updated Cap Choices for high power motor controllers Updated*. Retrieved from endless-sphere: https://endless-sphere.com/sphere/threads/updated-cap-choices-for-high-power-motor-controllers-updated.22194/

McAllister, W. (2021). *Inductor kickback*. Retrieved from Spinning Numbers: https://spinningnumbers.org/a/inductor-kickback.html

Reeves, M. (2019, Jul 19). *The GOD of Electric Skateboarding - Behind the scenes with VESC*. Retrieved from Youtube: https://www.youtube.com/watch?v=YHsPncLy5u8&ab\_channel=MojaReeves

Shaman. (2020). *Cheap FOCer 2 (Open-source, Low-cost, VESC 6 based ESC)*. Retrieved from VESC Project: https://vesc-project.com/node/1728

Shaman. (2022, jun). *Cheap FOCer 3 (open-source, Low cost, VESC 6 Based ESC)*. Retrieved from ESK8 News: https://forum.esk8.news/t/cheap-focer-3-open-source-low-cost-vesc-6-based-esc-in-development/67432/41?page=8

Shaman Systems. (2022). *Cheap-FOCer-2*. Retrieved from GitHub: https://github.com/shamansystems/Cheap-FOCer-2

Solutions, C. P. (n.d.). *The Causes of Electrolytic Capacitor Degradation*. Retrieved from resources.pcb.cadence: https://resources.pcb.cadence.com/blog/2022-the-causes-of-electrolytic-capacitor-degradation

Vedder, B. (2017). *The VESC Project*. Retrieved from VESC Project: https://vesc-project.com/

Vedder, B. (2023). *BLDC*. Retrieved from GitHub: https://github.com/vedderb/bldc