# Integrated Brushless Motor Exploration Platform

ECE 445 Design Document

Project #17 Spring 2025

Alex Roberts and Jason Vasko

Professor: Yang Zhao
TA: Michael Gamota

# Contents

# 1 Introduction

## 1.1 Problem

As technology continues to develop, the electrification of mechanical loads continues to increase, such as the use of electric motors in robots and vehicles, where hydraulics and engines once were used. With the increased prevalence of electric motors in the coming years, there will be a growing field of study in motor controls. Currently, exploring topics in motor control requires at least a moderate knowledge of electronic hardware systems on top of the control theory being tested. Even when using commercial off-the-shelf motor drivers, system circuitry such as microcontrollers, power regulators, and power supplies still need to be properly chosen and connected together, which can be daunting.

There are individuals in math and controls heavy backgrounds, like aerospace engineers, which are likely lacking much of the electrical engineering background needed to get a motor even spinning, but they have the advanced knowledge of control systems to implement and test different algorithms for efficient motor control. There does not exist a simple solution for an all-in-one motor control platform designed for an educational use, as almost all commercial subsystems are optimized for application in products. This application focus removes all but the necessary circuitry for any subsystem to allow system designers to fit these modules in a wider array of products. This versatility however, places a problematic burden on a novice user to understand exactly how to connect every part of each subsystem, preventing people from exploring motor controls until they understand much of the electrical background behind them.

## 1.2 Solution

Our goal is to create a single PCB which combines as much circuitry as possible for the operation and advanced control of common electric motors. Specifically, we have focused on brushless DC motors in our solution, as they are incredibly common and are a prime target for control theory students with subjects such as field oriented control. The hardware platform would specifically combine the motor driver circuitry, microcontroller used for control, supplemental programming circuitry, and sensors required for the operation of the motor. The power supply used to power the phases of the motor will be external, but the board will have easy access points to use any common benchtop power supply. The main goal is to limit the number of physical connections required by the user, with only a USB for communication between the controller and a computer, a small wall plug for logic power, a benchtop power supply for motor drive power, and the motor phases themselves. As wall plugs and USBs are quite commonplace, ideally the user of our project would only need to connect two unfamiliar components, being the benchtop power supply, and motor phases. The motor controller would then communicate with an application on the computer, allowing users to modify and switch between the control algorithms used to spin the motor, as well as monitor real-time motor performance and PCB system health.

With a highly streamlined hardware platform, we aim to get more people interested in the field of electronic motor control. The motor driver circuitry would also be built out of discrete components where possible to encourage the natural development of hardware knowledge as the user explores motors. By breaking circuits out into individual components, we allow the user to see and understand each hardware block in the system, and eventually even experiment with changing hardware components as they become more advanced, such as changing FET technology or gate driver components. Ultimately, we aim to develop a hardware platform that lowers the barriers of entry to studying brushless motors, by allowing users to work backwards from a spinning motor and topics in motor control, back to fundamentals of hardware to allow them to begin designing motor systems independently.

## 1.3 Visual Aid

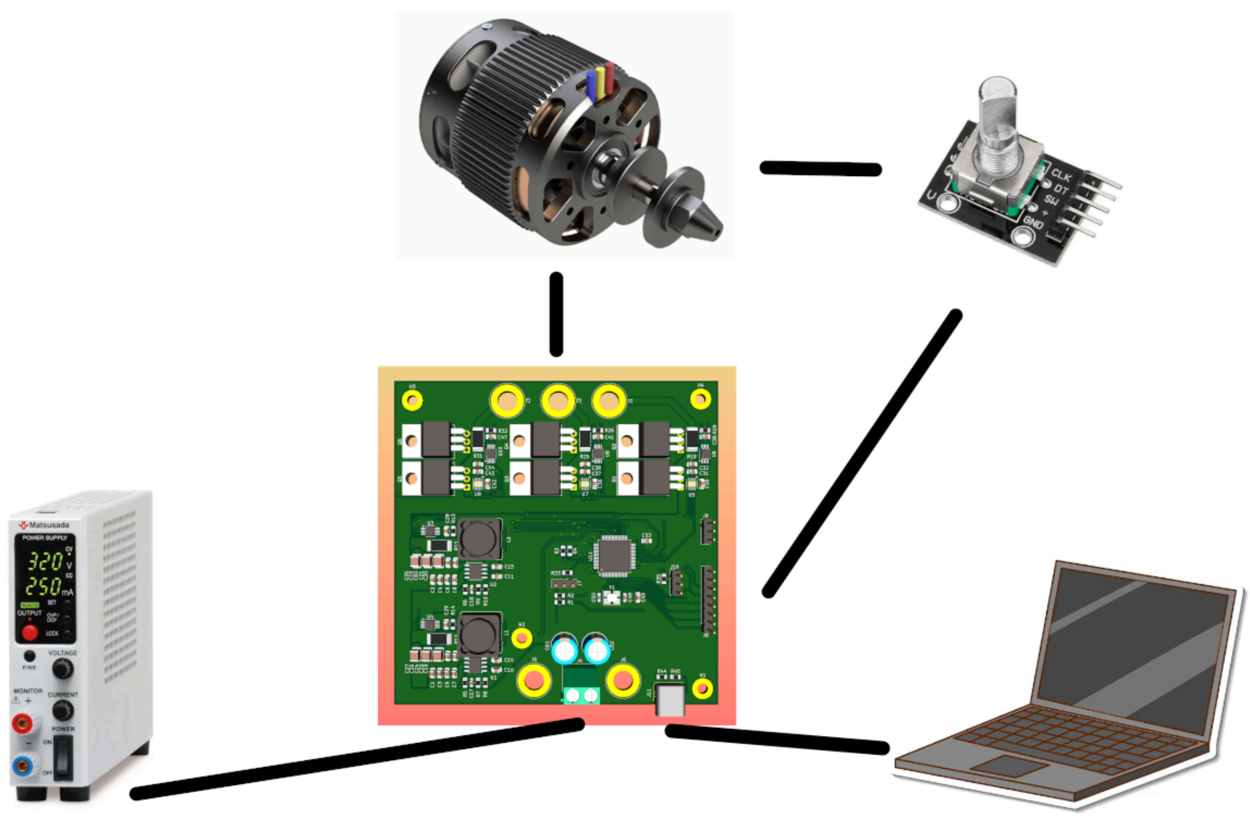


Figure 1 - System Application Visual Aid

Figure 1 depicts the use of our hardware platform in a system. Three to four external items are needed in addition to our PCB, which are a benchtop power supply, a laptop, and the motor itself. The laptop connects over USB to the PCB to control the system through a GUI application, which also allows for real-time data logging. The power supply directly supplies the motor bus voltage to the PCB, and the PCB has wide input voltage range regulators to generate logic voltage levels on the PCB from a single source for ease of use. Finally the motor connects to the three phase outputs from the motor drive subsystem. If the motor has an internal encoder, it can be directly connected to the PCB as well on the pin headers closest to the USB port. If the motor does not have an internal encoder and sensorless motor control is not desired, an external encoder can also be used and connected to the same pin headers.

## 1.4 High-level Requirements

The success of our project is governed by three primary requirements:

1. *Motor operation and control* - The user should be able to start, stop, and control the speed of a brushless DC motor at any operating voltage between 12V and 24V using a GUI program, and achieve at least 1000 rotations per minute at top speed.
2. *Configurability* - The user should be able to switch between at least two distinct motor control algorithms (such as sinusoidal and trapezoidal control) and tweak algorithm parameters such as PID coefficients, PWM frequency, control frequency, and more.
3. *Motor performance and system health monitoring* - The user should be able to view motor phase voltages, phase currents, motor speed, shaft position, and voltage rail power consumption on the PCB in real time (>1Hz rate, <500ms latency) with historical graphs.

# 2 Design

## 2.1 Block Diagram

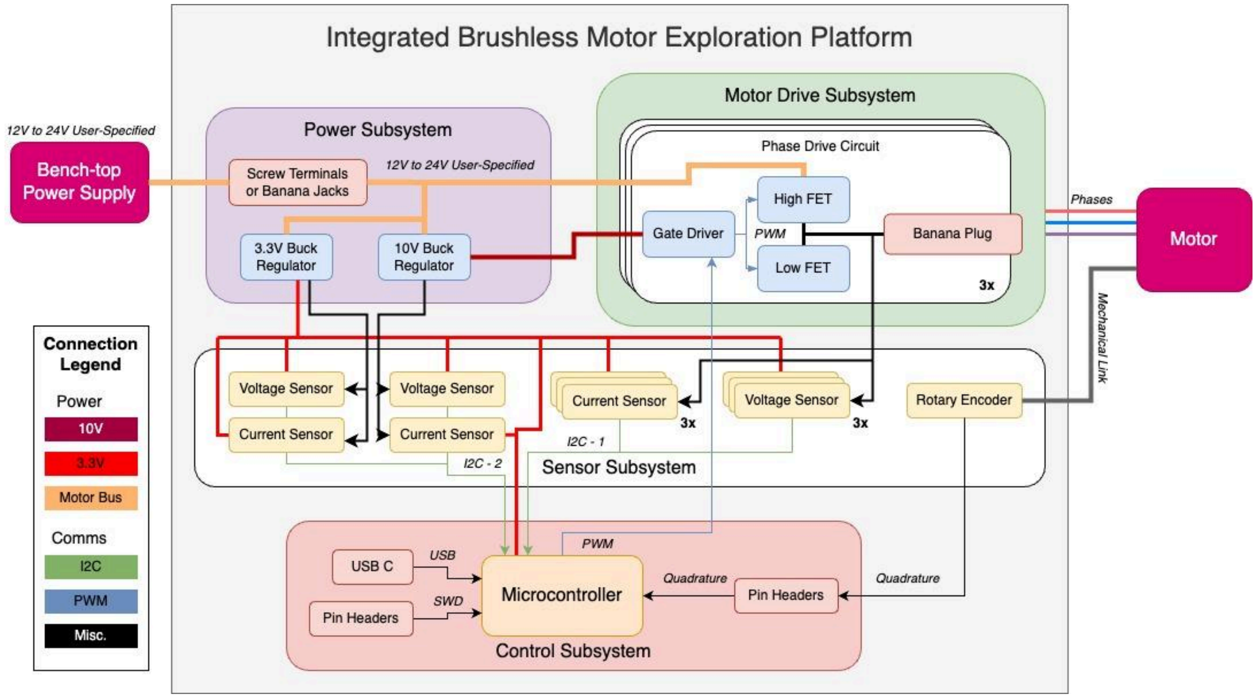


Figure 2 - System Block Diagram

Figure 2 shows the block diagram of our system, will all major power connections and communication interfaces between components highlighted. There are four major subsystems in the design. The power subsystem, which handles external power input and voltage regulation, the motor drive subsystem, which has the phase half-bridges and gate drivers, the sensor subsystem, which contains all the sensors on the PCB to measure output values and system health parameters, and finally the control subsystem, which has the microcontroller and several connectors to program the PCB and establish a data link to control the motor.

## 2.2 Power Subsystem

The power subsystem is responsible for generating the voltages needed for components on the board such as sensors, gate drivers, and the microcontroller. There are two possible connectors for input power from a benchtop power supply, either screw terminals or banana jacks. This input voltage is then passed to the motor driver MOSFETs directly as the motor bus voltage, as well as to two switching regulators to step the voltage down to 3.3V and 10V. 3.3V is fed to the microcontroller and sensors on the board, while 10V is fed to the gate driver IC.

This power architecture was chosen as it's the simplest for the user to interact with, only requiring a single connection to the PCB. The user can set the motor bus voltage to anything within the allowable range of 12V to 24V and the PCB will handle the rest, operating as intended. The part numbers of major components in this subsystem are shown below:

- Buck ICs: LMR33640ADDAR
- Screw terminal connector: 1715721
- Banana jack connector: CT2220

### 2.2.1 Requirements and Verifications

Below is table 1, which covers the subsystem requirements for the power system. These requirements are focused on ensuring the input voltage range does not affect the operation of the PCB, and that the core functionality of the subsystem, which is voltage regulation, operates within reasonable tolerances.

| Requirement | Verification |
|---|---|
| All voltages required on the PCB except the motor bus voltage shall be generated on the PCB | This is verified through design, as if the architecture is correct, the only external power connection will be the motor bus voltage. |
| All voltages generated on the PCB should have an accuracy of +/-5% around their set-point. | Usage of an oscilloscope to measure switching regulator voltage ripple under light and heavy loads for both the 3.3V and 10V rail at the extremes of the input voltage range will allow us to verify this requirement. |
| The PCB should function properly over an input voltage range of 12V to 24V. | As the input voltage only directly interfaces with the motor drive subsystem and power subsystem, if a motor can be spun at a speed of at least 1000 rpm and stopped, at both 12V and 24V input voltage, this requirement will be satisfied as the extremes of operation were validated. |

Table 1 - Power Subsystem Requirements and Verification

## 2.3 Motor Drive Subsystem

The motor drive subsystem is responsible for generating the AC waveforms supplied to each phase of the motor. This is done using half-bridge inverters for each phase, and given a specific PWM signal from the microcontroller a wide array of AC waveforms can be generated. Dedicated gate drive circuitry ensures the high-side MOSFETs can properly switch, and both high-side and low-side FETs switch quickly enough. Overall this subsystem is responsible for the driving of the motor, and takes as inputs the motor bus voltage, gate driver supply voltage of 3.3V, and six PWM inputs for the high and low MOSFET gate signals for each of the three phases. Each phase then outputs the generated AC signal to a banana jack on the PCB that the user will connect the motor phase to. The part numbers for major subsystem components are as follows:

- MOSFETs: IRFI1310NPBF
- Gate Drivers: DGD05473FN-7
- Banana jack connector: CT2220

### 2.3.1 Requirements and Verifications

Below is table 2, which covers the subsystem requirements for the motor driver system. These requirements are focused on ensuring the input voltage range does not affect the operation of the PCB, and that the core functionality of the motor drive subsystem, which is AC waveform generation, has proper safeguards and can hit important requirements, such as deadtimes and minimum frequencies.

| Requirement | Verification |
| --- | --- |
| The PCB should function properly over an input voltage range of 12V to 24V. (Duplicate with power subsystem) | As the input voltage only directly interfaces with the motor drive subsystem and power subsystem, if a motor can be spun at a speed of at least 1000 rpm and stopped, at both 12V and 24V input voltage, this requirement will be satisfied as the extremes of operation were validated. |
| The motor subsystem should be able to generate both trapezoidal and sinusoidal phase waveforms. | An oscilloscope can be used to view the waveform output across two of the phases in trapezoidal drive mode, and again in sinusoidal mode. The sinusoidal waveshape should be reasonably sinusoidal and the trapezoidal wave shape will go between motor bus voltage, zero volts, and the negative motor bus voltage. |
| The motor drive subsystem should have hardware restrictions on shoot through, and should have at least 2ns of time between FET transitions. | Use an oscilloscope to verify at least 2ns between one FET turning off and the other FET turning on for both possible half-bridge transitions. |
| Each half-bridge should be capable of 100kHz PWM frequency. | Using an oscilloscope, verify that a 100kHz, 50% duty cycle square wave can be generated on any of the three phases at a 24V input voltage. |

Table 2 - Motor Drive Subsystem Requirements and Verification

## 2.4 Sensor Subsystem

The sensor array is responsible for recording data related to the motor's operation and the overall health of the board. This subsystem will monitor both current and voltage of the three-phase signals driving the motor outputs and the voltage regulators to track power subsystem health. This subsystem will also accept quadrature inputs of motors with integrated encoders, or the connection of an external encoder to the motor for more accurate rotor position and speed measurements. Overall this system is simply 5 power monitoring ICs that measure voltage and current, spread across two I2C busses, and inputs for encoders if sensorless motor control is not desired. The part numbers of the critical subsystem components are listed below:

- Power monitor IC: INA230AIDGSR
- Shunt resistor: CRF2512-FZ-R005ELF

### 2.4.1 Requirements and Verifications

Below is table 3, which covers the subsystem requirements for the sensor system. These requirements are focused on ensuring the input voltage range does not affect the operation of the PCB, and that the core functionality of the motor drive subsystem, which is AC waveform generation, has proper safeguards and can hit important requirements, such as deadtimes and minimum frequencies.

| **Requirement** | **Verification** |
|---|---|
| All three phases should have voltage and current measured within 150mV and 100mA. | Using an oscilloscope, the actual voltage/current and reported voltage/current for each of the three phases can be measured and evaluated. |
| Any voltage rail generated on the PCB should have the voltage, current, and power reported. Measurements are for system health so an accuracy of 5% is acceptable. | Again an oscilloscope and electronic load can be used to measure the actual voltage, current, or power, and can be compared to the measured value from the sensor subsystem. |
| Motor speed should be collected either by an encoder or through measurement of back EMF with +/-10% accuracy. | A tachometer can be used to get a reference RPM of the motor under certain conditions, and either our back EMF calculation, or an external encoder connected to the motor connected to the quadrature inputs should report a speed within 10% of the tachometer's reading. |

Table 3 - Sensor Subsystem Requirements and Verification

## 2.5 Control Subsystem

The control subsystem is responsible for driving and monitoring all other subsystems. It will periodically read data from the sensor array, monitor the health of the power subsystem, and generate PWM signals for the motor drive subsystem. It will also communicate with the PC app, updating the GUI periodically and allowing the user to start and stop motor rotation, control speed setpoints, and change between at least two motor control algorithms, such as the trapezoidal and sinusoidal drive algorithms. The development process will use serial wire debug over a 100 mil pin header to program and debug code on the microcontroller, and during main use the USB C port will communicate with the platform as a virtual COM device. Main subsystem components include:

- Microcontroller: STM32F401RBT6
- USB C port: USB4085-GF-A
- 100mil headers for GPIO and programming

### 2.5.1 Requirements and Verifications

Below is table 4, which covers the subsystem requirements for the control system. These requirements are focused the ease of programming and development on the microcontroller, the speed of data reporting from the PCB to the computer GUI application, and the safety shutdown timing for when the USB link is broken between the computer and the PCB.

| **Requirement** | **Verification** |
|---|---|
| The microcontroller should be programmable and debuggable over serial wire debug (SWD) | Code to output a square wave on a GPIO can be programmed to the microcontroller. An oscilloscope can verify the code was delivered by looking at the GPIO, and the square wave should stop if a breakpoint is set and triggered. |
| The microcontroller should be programmable over USB C | Similarly, code to output a known signal on a GPIO pin can be uploaded over USB and then the GPIO can be measured to verify this requirement. |
| The microcontroller should report data in real time, at an update rate of greater than 1Hz and a latency of 500ms to the GUI application on the computer | The microcontroller can output a known waveform on the motor drive subsystem, and an oscilloscope can measure when a transition happens, and a timer can be started to measure the delay between the event occurring in real life and the arrival of the data on the computer. A 2Hz square wave output is a good waveform to test these quantitative limits. |
| Any failed connection between the computer and PCB should stop the motor from spinning within 2 seconds | Configure the platform to be spinning the motor. Unplug the USB link from the computer side and time how long it takes for the motor to stop spinning. If the data link is severed and the motor stops within two seconds, this requirement is verified. |

Table 4 - Control Subsystem Requirements and Verification

## 2.6 Tolerance Analysis

High accuracy sensing is incredibly important in any control loop, as if a control algorithm is changing system outputs based on inaccurate data, the system outputs are highly likely to be of poor quality and cause the control loop to fail to reach the setpoint. This issue is present in several areas of our system, one of the most important being the phase voltage sensing. Without accurate phase voltage sensing, it's not clear to the system what type of electrical signals are truly being supplied to the motor phases. If the phase waveforms are not controlled very carefully, the motor could fail to even spin, let alone achieve precise control. The metric we chose was less than 150mV of voltage error.

The sensor being used is the INA230AIDGSR. The phase voltage will be measured as the "bus voltage" in the sensor. The datasheet reports an LSB sensing value of 1.25mV and a gain error of 0.3% worst case for the bus voltage measurement [2]. The motor bus voltage range of our PCB is 12V to 24V. When the analog value is sampled in the ADC the smallest resolution is 1.25mV, so the worst case is to assume the quantization process introduces the full error possible of 1.25mV. Since the error metric is in mV, the highest operating voltage will introduce the most error since the percentage error will scale off a higher base value. Given an ideal 24V input, the ADC could quantize up or down by 1.25mV to give 24.00125V or 23.99875V, which can then be scaled by 100.3% or 99.7% respectively, to report 24.0733V or 23.9268V. If both errors are worst case overestimates, the error is:

$$24.0733[V] - 24[V] = 0.0733[V] = 73.3[mV]$$

If both errors are worst case underestimates, the error is:

$$24[V] - 23.9268[V] = 0.0732[V] = 73.2[mV]$$

Either way the sensor is accurate enough to stay within our goal of 150mV of error.

# 3 Cost and Schedule

## 3.1 Cost Analysis

The total cost of materials is $70.10 per device. We estimate the total cost of labor, including development time and assuming two prototype devices will be constructed, to be $26300. Thus, adding the cost of materials for two devices, we expect a total cost of $26440.20 for this project.

### 3.1.1 Materials Cost Calculation

Below is table 5, which is a bill of materials for the PCB and system. All components have their part number, item name, specifications, vendor (with link), price, quantity, and total cost listed. In total the entire cost of the bill of materials table is $65.80. Note that normally the INA230AIDGSR would be used for the power monitor IC, but due to stocking issues the more accurate INA226AIDGSR will be used as a replacement, though the original component is sufficient to meet all design requirements normally.

| **Item** | **Part Number** | **Specifications** | **Vendor** | **$ Per** | **QTY** | **Total $** |
|---|---|---|---|---|---|---|
| 10uF Cap | GRM21BC8YA106ME11L | CAP 0805 10uF 35V Ceramic | Digikey | $0.28 | 6 | $1.68 |
| 1uF Cap | CL21B105KBFNNNE | CAP 0805 1uF 50V Ceramic | Digikey | $0.08 | 4 | $0.32 |
| 100nF Cap | CC0805KRX7R9BB104 | CAP 0805 100nF 50V Ceramic | Digikey | $0.08 | 19 | $1.52 |
| 2.2uF Cap | C2012X7R1C225K125AB | CAP 0805 2.2uF 16V Ceramic | Digikey | $0.18 | 3 | $0.54 |
| 4.7 uF Cap | GRM21BR61H475KE51L | CAP 0805 4.7uF 50V Ceramic | Digikey | $0.19 | 5 | $0.95 |
| 220nF Cap | C0805C224K5RACTU | CAP 0805 220nF 50V Ceramic | Digikey | $0.10 | 2 | $0.20 |
| 100uF Cap | GRM32ER61A107ME20L | CAP 1210 100uF 10V Ceramic | Digikey | $0.84 | 6 | $5.04 |
| 56pF Cap | C0603C560J5GACTU | CAP 0603 56pF 50V Ceramic | Digikey | $0.10 | 1 | $0.10 |
| 22pF Cap | C0603C220J5GACTU | CAP 0603 22pF 50V Ceramic | Digikey | $0.12 | 3 | $0.36 |
| 100uF Cap | EEH-AZA1V101B | CAP TH 100uF 35V Alum Hybrid | Digikey | $1.42 | 2 | $2.84 |
| 5.1k Res | RC0805FR-075K1L | RES 5.1K OHM 1% 1/8W 0805 | Digikey | $0.10 | 6 | $0.60 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 10k Res | RMCF0805FT10K0 | RES 10K OHM 1% 1/8W 0805 | Digikey | $0.10 | 7 | $0.70 |
| 5m Res | CRF2512-FZ-R005ELF | RES 0.005 OHM 1% 2W 2512 | Digikey | $0.49 | 5 | $2.45 |
| 100k Res | RC0805FR-07100KL | RES 100K OHM 1% 1/8W 0805 | Digikey | $0.10 | 4 | $0.40 |
| 11k Res | CRG0805F11K | RES 11K OHM 1% 1/8W 0805 | Digikey | $0.10 | 1 | $0.10 |
| 43.2k Res | ERA-6AEB4322V | RES 43.2K OHM 0.1% 1/8W 0805 | Digikey | $0.10 | 1 | $0.10 |
| 0 Res | ERJ-6GEY0R00V | RES 0 OHM 1/8W 0805 | Digikey | $0.10 | 13 | $1.30 |
| 22uH Ind | SRR1260-220M | IND 22UH 4A 43mOHM SMD | Digikey | $0.96 | 2 | $1.92 |
| Banana Jack Plug | CT2220 | CONN BANANA JACK THRD | Digikey | $0.95 | 5 | $4.75 |
| 100mil Header | 61300811121 | PIN HEADER VERT 8POS 2.54MM | Digikey | $0.36 | 3 | $1.08 |
| Screw Terminals | 1715721 | TERM BLK 2P SIDE ENT 5.08MM | Digikey | $0.97 | 1 | $0.97 |
| USB C | USB4085-GF-A | CONN RCPT USB2.0 TYPE C 16+8POS | Digikey | $0.88 | 1 | $0.88 |
| MOSFETs | IRFI1310NPBF | MOSFET N-CH 100V 24A TO220AB FP | Digikey | $2.12 | 6 | $12.72 |
| 16MHz Crystal | ECS-2333-160-BN-TR | XTAL OSC XO 16MHZ HCMOS SMD | Digikey | $0.84 | 1 | $0.84 |
| 4A Adj Buck IC | LMR33640ADDAR | IC REG BUCK ADJ 4A 8SOPWR | Digikey | $1.92 | 2 | $3.84 |
| Half Bridge Gate Driver | DGD05473FN-7 | IC GATE DRV HALF-BRDG DFN3030-10 | Digikey | $1.31 | 3 | $3.93 |
| MCU | STM32F401RBT6 | IC MCU 32BIT 128KB FLASH 64LQFP | Digikey | $3.97 | 1 | $3.97 |
| Current and Voltage Sensor | INA226AIDGSR | IC CURRENT MONITOR 0.02% 10VSSOP | Digikey | $2.34 | 5 | $11.70 |

Table 5 - Bill of Materials

In addition to the cost of materials on the PCB, there is also the cost for the PCB itself, though this is quite cheap. Our PCB gerber files for the first revision were uploaded into the JLCPCB quote tool to estimate the pricing of our PCB. This quote is shown below in figure 3. Ignoring the promotional deal, it would be $4 for QTY 5 of our PCB, with $17.50 in shipping costs. Factoring in shipping, it costs $21.50 for 5 PCBs, or $4.30 per PCB, which can be added to the previous materials cost estimate.

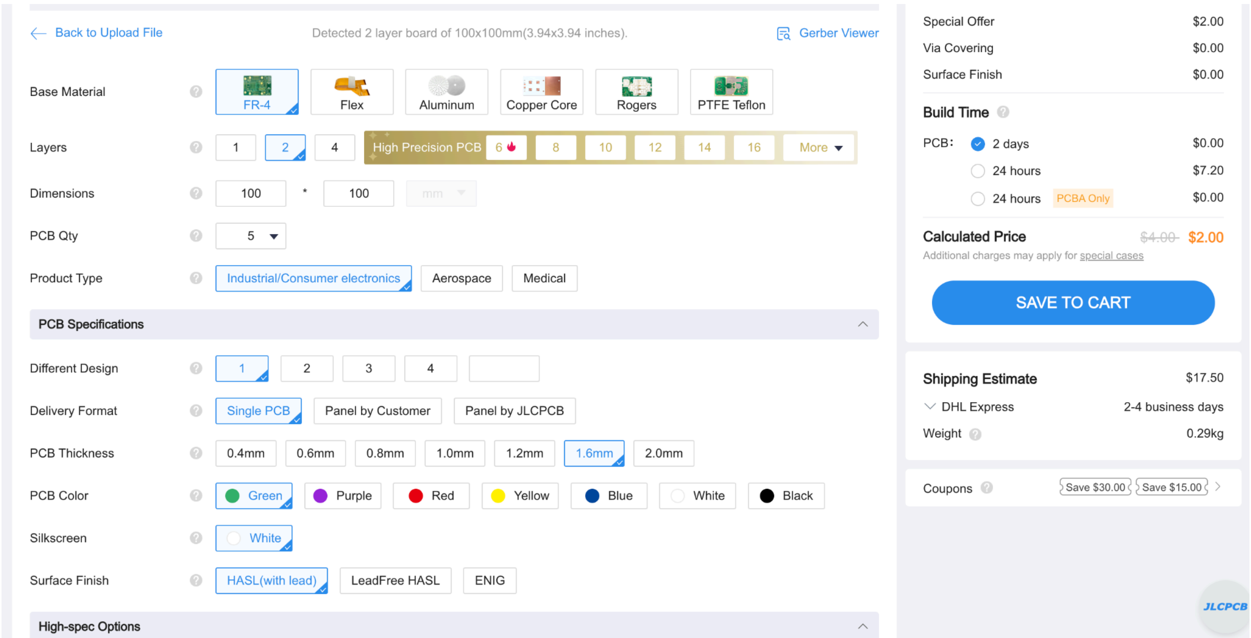


Figure 3 - JLCPCB Quote

In total then, with $65.80 in component costs and $4.30 in the PCB cost, the total cost for our hardware platform is $70.10 in materials.

### 3.1.2 Labor Cost Calculation

First, we will assume a reasonable salary of $40/hr. The labor cost can be divided into development time and device construction time. For device construction, we conservatively estimate 4 hours of labor for device soldering, programming, and validation per device. For development, we can further divide the labor duration into hardware and software development. For hardware development, we estimate 35 hours of work on the initial design and PCB, 50 hours for evaluation and verification of the first prototype, and 50 hours for fixing any issues and evaluating a second prototype. This yields a total hardware development time of 135 hours. For software development, we estimate 20 hours of development time for motor control functionality and sensor drivers and 20 hours of development time for the GUI app and the associated communication link. By convention, we will triple the expected software development time to account for unforeseen bugs, as fixing such bugs usually accounts for the majority of software development time, giving a total software development time of 120 hours. Therefore, we expect a total development time of 255 hours. Adding a total device construction time of 8 hours for two device prototypes, we estimate the total labor time of this project to be approximately 263 hours, and a total cost of:

$$263[hr] \times 40[\$/hr] \times 2.5 = 26300[\$]$$

Note that this estimate is likely to be inaccurate due to the unpredictability of development times.

## 3.2 Schedule

Below in table 6 is an estimated week-by-week schedule for the remaining weeks in the semester. Each of us has at least one general task per week, and we will always be available to help one another if needed throughout the week. The final three weeks are likely the most important, with the mock demo being the week of 4/21, the final demo and mock presentation the week of 4/28, and the final presentation and paper the last week of 5/5. Note that for any documentation, as a general rule, Alex will handle information primarily related to hardware while Jason will handle information primarily related to software, and general system information can be handled by either. Also note that if we finish tasks early we may progress to later tasks or begin incorporating more advanced motor control algorithms into our design earlier than specified on the schedule.

| **Week of** | **Alex's Tasks** | **Jason's Tasks** |
|---|---|---|
| 3/10 | - Create second hardware revision to try alternative layouts and fix small problems from first revision before 2nd order deadline<br>- Piecing together hardware setup for breadboard demo<br>- Working with Jason to integrate HW/SW and prep. for breadboard demo during this week | - Generate PWM waveforms on dev board<br>- Work with Alex to integrate HW/SW and prep. for breadboard demo during this week |
| 3/17 | SPRING BREAK | SPRING BREAK |
| 3/24 | - Assembling PCBs for first hardware revision<br>- Performing hardware bringup and evaluation of first assembled unit<br>- Handing unit off to Jason for his software testing<br>- Probably working on third hardware revision based on information gained | - Develop drivers to read data from sensors and test as necessary |
| 3/31 | - Finishing third hardware revision and ordering. Likely to just test alternate layout<br>- Assembling and evaluating a second revision PCB to test second layout<br>- Working with Jason for system integration | - Finish up driver development and testing for sensors<br>- Incorporate sensor readings into motor control algorithm with the goal of successfully driving a motor with hard coded parameters on the microcontroller |
| 4/7 | - Creating plan for mock demo<br>- Helping with any hardware issues in software evaluation | - Develop the GUI app's frontend and backend (specifically, establish a connection from the GUI app to the microcontroller, choose a message format, |

| | | |
|---|---|---|
| | - Researching more motor control algorithms to contribute to software side as hardware ends | add response logic on the microcontroller, and add a control interface and a performance viewer on the GUI app) and test as necessary |
| 4/14 | - Solidifying mock demo, practicing the demo, ironing out any issues<br>- Begin creating final presentation outline | - Finish GUI app frontend and backend development and testing<br>- Test the final product rigorously and fix any remaining bugs<br>- Incorporate advanced motor control algorithms, time permitting |
| 4/21 | - Performing the mock demo for TA<br>- Making plan for final demo based on mock demo feedback, fixing any issues<br>- Working on the final presentation<br>- Beginning on final paper | - Perform the mock demo for the TA<br>- Fix any remaining bugs and incorporate feedback from the mock demo<br>- Incorporate advanced motor control algorithms, time permitting |
| 4/28 | - Performing the final demo<br>- Wrapping up final presentation<br>- Giving mock presentation and incorporating feedback<br>- Working further on the final paper | - Perform the final demo<br>- Work on the final presentation<br>- Give the mock presentation and incorporate feedback as necessary<br>- Work on the final paper |
| 5/5 | - Giving final presentation<br>- Completing final paper | - Give the final presentation<br>- Finish the final paper |

Table 6 - Schedule

# 4 Ethics and Safety

## 4.1 Ethical Concerns

In considering the ethics surrounding the development and existence of this project, two main points in the IEEE code of ethics stand out in their relation to this project. The central focus of the project is to be an educational platform for the driving and control of brushless DC motors, an already highly utilized and increasingly important technology. The IEEE code of ethics mentions in point 2 the ethical need to improve the understanding of people in the capabilities of conventional and emerging technologies [1]. It also highlights the necessity of treating all persons equally and with respect, regardless of background in point 7 [1]. We believe the existence of our project is closely linked with these two points, as it will provide a way for a wider array of people to understand brushless DC motors. Additionally, as we aim to make the platform as accessible as possible, we lower the barrier of education which can be frequently apparent in hardware systems, where price of equipment can make learning less accessible. By making an open learning platform using common and cheap components, students from less wealthy backgrounds may be able to study a subject they were previously unable to gain hands-on experience with.

It's also very important to follow all aspects of the IEEE code of ethics during the development process of the project, including supporting one another as teammates as outlined in point 10 [1].

## 4.2 Safety Concerns

As our project involves a spinning object in the motor and interacts with unknown external components like the bench-top power supply, there are some safety concerns for the user. First, OSHA standard 1910.303(g)(2)(i) considers voltages of 50V DC or more to be hazardous [3]. Our board is designed to use 24V DC input, which is well below that limit, so users have no reason to work with high voltages. Bench-top power supplies are generally a safe option, but requiring an inexperienced user to control one is potentially dangerous. For example, if the user supplies power to the motor at an excessive bus voltage, this could cause damage to the system's components and harm to the user. As such, we have designed safety into our system using overvoltage protections. In addition, we will design our software to be as safe as possible, prompting the microcontroller to stop the motor if the computer connection is lost or if the system health sensors report any potential problems. Lastly, we will protect the user from accidentally starting the motor using confirmation messages in the GUI. However, we decided not to use a physical stop button on the PCB to avoid encouraging the user to reach for the PCB in a panic, which could cause further harm in the event of an errant motor.

# 5 Citations

[1] - IEEE, "IEEE Policies - Section 7-8 - IEEE Code of Ethics," [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 13-Feb-2025].

[2] - Texas Instruments, "INA230 36V, 16-Bit, I2C Output Current, Voltage and Power Monitor With Alert," [Online]. Available: https://www.ti.com/lit/ds/symlink/ina230.pdf. [Accessed: 13-Feb-2025].

[3] - Occupational Safety and Health Administration, "Guarding requirements for 50 volts or more DC.," [Online]. Available: https://www.osha.gov/laws-regs/standardinterpretations/2015-09-04. [Accessed: 7-Mar-2025].