VFD Motor Controller

Mackenzie Miller
Andrew Nguyen
Aidan Rader
Ryan Regan

CONCEPT OF OPERATIONS

REVISION – 1
28 April 2025

CONCEPT OF OPERATIONS
FOR
VFD Motor Controller

TEAM VFD MOTOR CONTROLLER

APPROVED BY:

____
Project Leader   Date

____
Prof. Lusher   Date

____
T/A   Date

Change Record

Rev.	Date	Originator	Approvals	Description
0	9/15/2024	VFD Motor Controller Team		Draft Release
1	4/28/2025	VFD Motor Controller Team		Final Report Update

Table of Contents

List of Figures

1. Executive Summary

The sponsor for this project is in need of a VFD motor controller for an AC induction motor. The VFD will control the speed and torque of the AC motor. This project will help to increase energy efficiency by ensuring that the motor isn't running at a higher speed than necessary. It also makes the system more reliable and sustainable by decreasing mechanical stress.

A Variable Frequency Drive (VFD) is a device that controls the speed of an electric motor by varying the frequency of the delivered voltage. VFDs are useful because they improve the energy efficiency, reliability, safety, stability, and controllability of a motor. The goal of this project is to make a VFD motor controller that will efficiently control the seed and torque of an electric motor.

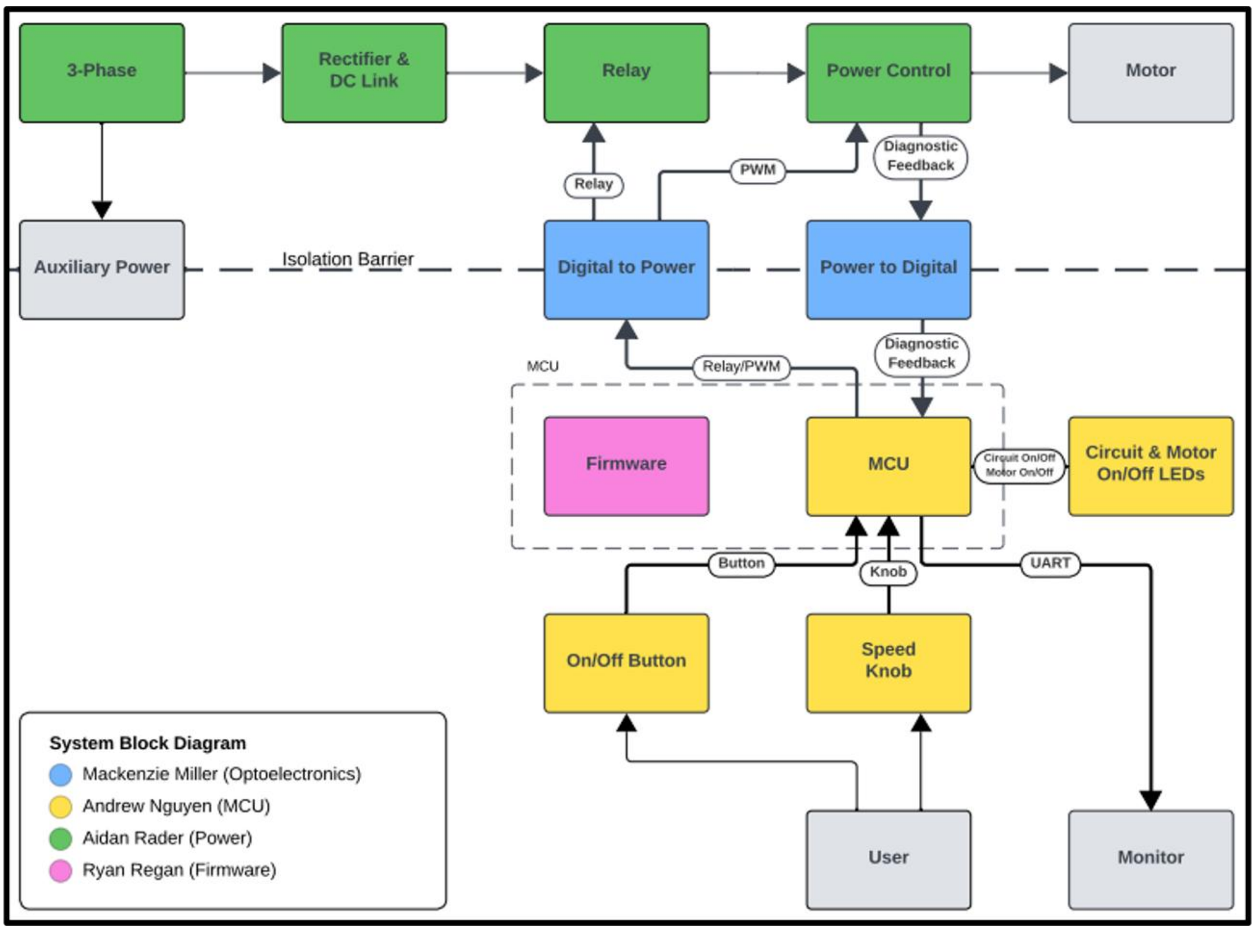

Figure 1: VFD Motor Controller Block Diagram

2. Introduction

2.1 Background

VFD motor controllers have several beneficial functionalities that can improve the quality of a system. A VFD motor controller regulates the speed and torque of electric motors by controlling the frequency and voltage supplied. The speed is controlled by varying the frequency of the supply of electricity. It also allows for more efficient systems. By running at certain speeds, the VFD can reduce the power consumed, allowing for energy to be saved. It also reduces stress on the motor by allowing for soft starting and stopping. The VFD also has a functionality allowing it to shut off the motor if something is shorted, overheating, or overloaded. The VFD will replace a fixed speed motor in a mechanical system.

2.2 Overview

While fixed speed motor controllers are still useful in some applications, more often than not a VFD motor controller is a better alternative. It offers all the functionality of a fixed speed motor controller while also being able to vary the speed and torque of the motor if needed. In this project, our team will work to create a Variable Frequency Drive to control motor speed. In addition, we will consider several different factors such as user interface for the drive, and implementation of the drive in a closed and open loop. This upgrade will improve the energy efficiency, reliability, safety, stability, and controllability of the motor.

2.3 Referenced Documents and Standards

"The Advantages of Frequency Drive Operation in Submersible Pumps." *Grundfos*, www.grundfos.com/us/learn/ecademy/all-courses/the-sp-submersible-pump/the-advantages-of-frequency-drive-operation-in-submersible-pumps. Accessed 15 Sept. 2024.

By, and Dosupply. "What Do VFDS DO in HVAC Systems?: Do Supply Tech Support." *Do Supply Tech Support | Information and Troubleshooting on Allen-Bradley, Eaton, GE Fanuc, and More*, 11 Aug. 2022, www.dosupply.com/tech/2022/08/11/what-do-vfds-do-in-hvac-systems/.

Hoffman, Theresa. "What Are the Benefits of a Variable Frequency Drive?" Wolf Automation,www.wolfautomation.com/blog/what-are-the-benefits-of-a-variable-frequency-drive/. Accessed 15 Sept. 2024.

"Variable Frequency Drive for Conveyor and Material Handling." *Variable Frequency Drive for Conveyor and Material Handling-Darwin Motion*, darwinmotion.com/blogs/variable-frequency-drive-for-conveyor-and-material-handling#:~:text=The%20use%20of%20a%20Variable,belt%20speed%2C%20reducing%20energy%20consumption. Accessed 15 Sept. 2024.

IEEE 519-2014 Standard for Harmonics

3. Operating Concept

3.1 Scope

The VFD motor controller is intended to control the speed and torque of an AC motor by varying the frequency and voltage supplied. This project will use three phase power as the input that will be converted to DC and then transmitted to a microcontroller via optoelectronics and used to power a motor. The microcontroller also will send signals to the DC link, again using optoelectronics, that contain the desired frequency specified by the user on the user interface. This VFD shall be implemented to increase efficiency and save energy.

3.2 Operational Description and Constraints

The VFD motor controller is designed to vary the frequency and voltage supplied to an AC motor to control its speed and torque. It converts a three-phase AC input into DC using a rectifier and transmitting the power using a microcontroller.

- Microcontroller Function
    - The microcontroller is responsible for generating pulse-width modulation (PWM) signals which are used to adjust the frequency and voltage supplied to the motor.
    - The user can set the desired motor speeds via the user interface or the code itself. The microcontroller then ensures that the motor is running according to its specifications.
- Optoelectronics
    - LED lights are used to communicate between the high-voltage power components and the low-voltage microcontroller. This ensures that the signals are safely transmitted.
- User Interface
    - The microcontroller will have a potentiometer that allows the user to change the frequency of the three-phase PWM signal that is being output to the motor.
    - While the PCB is plugged into a computer, opening the output terminal in MPLab will allow the user to view debug statements such as the potentiometer’s value, output duty cycles to phases A, B, and C, sine wave table step size, and desired output frequency of the PWM sine wave.
- Constraints:
    - Sustainability
        - Humidity, dust, and temperatures outside of the specified range can cause the VFD to malfunction.
    - Noise levels
        - The VFD is noise sensitive and is susceptible to external electrical noise, which could decrease the reliability of the system.

- Power
  - The VFD requires stable three-phase AC power. Spikes or fluctuations could damage some of the system's components.
- Mechanical wear and maintenance
  - All parts of the VFD are subject to general mechanical wear and tear over time, especially if operated under high loads or in challenging environments.
- Cost and installation
  - The VFD system can be expensive to produce, especially when considering large-scale applications. Consideration must be given to the specific needs of the project

3.3 System Description

The VFD motor controller is comprised of a DC link (includes a rectifier and DC bus), power controller, microcontroller, and optoelectronics. The DC link converts AC input into DC and minimizes the noise of the DC signal. The power controller controls the voltage. The microcontroller is used to produce PWM signals that will drive the motor. Optoelectronics allows for communication between the high voltage side and the low voltage microcontroller.
This is a four-person project, and the roles are be split into:

- Firmware:
  - Write code in C with MPLAB that programs the microcontroller so that it functions with the rest of the VFD system according to desired specifications.
  - Use a potentiometer to allow user to change the frequency of three-phase PWM signal.
  - Print debug and feedback signals to UART console to allow for even further debugging.
  - Test program on development board to ensure functionality then ensure proper integration with the project's microcontroller.
- Optoelectronics
  - The first part of the optoelectronics portion includes two isolation circuits. One of those takes digital signals in and directly converts them to analog. The second, being more complicated, takes analog input, then, using an opto-isolator, the circuit converts the input to digital signals to be sent to the microcontroller. The analog side of the project works with very high voltage, while the digital side works with low voltage. Because of this, the two sides cannot be connected or the microcontroller will be overpowered and break, so the optoelectronics send signals and data across using light.
  - The second part of the sensors portion is a constant current and voltage measurement. This will be done using a current sensing resistor, which has a

very small resistance, and a simple voltage divider circuit. These measurements will then be sent back to the firmware.

- Microcontroller:
  - The MCU board is supplied by taking in the 120VAC from the wall and using an AC/DC converter to step it down to 15VDC. The voltage is then stepped down once more using the LM2595s-3.3 buck converter to a usable 3.3V that the MCU can handle. It receives feedback through low-voltage analog signals that represent voltage and current. The MCU will then send out PWM signals to the H bridge and power control system which are used to adjust the output voltage and frequency supplied to the motor. The board will also supply the optoelectronics subsystem with a common GND and 3.3V, and the power subsystem with a common GND, 15V, and 3.3V.
- Power:
  - The rectifier takes in three-phase AC power and converts it into DC power. For each phase, there are parallel diodes acting as a one-way bridge for the current allowing it to flow in only one direction. To maintain the correct current polarity the diode opens and closes in sequence as the AC waveform alternates. The DC output is then filtered by capacitors within the DC link to provide a stable DC voltage for the microprocessor.
  - The power control will take the DC power and the PWM signals and output three modified sign waves at 120 degrees to the motor. It will also output three measurement signals that can be used as diagnostic feedback: voltage, current, temperature
  - The relay will take in a relay signal originating by the button press on the microcontroller and toggle the motor between on and off states.
  - Converters will be used as part of auxiliary power to generate the necessary voltage levels for the different integrated chips. The two converters on the power PCB with be a 3.3 V_{DC} to isolated 5 V_{DC} converter and a 15 V_{DC} to isolated 15 V_{DC}.

3.4 Modes of Operations

There are two primary modes of operation for the variable frequency drive:

On: The motor receives 10-60Hz PWM signals depending on the position of the potentiometer located on the microcontroller PCB. The frequency of the PWM signals changes the speed of the motor.

Off: The relay is tripped, and the motor is no longer receiving any power.

3.5 Users

The potential user for our VFD would be a plant operator or manager. The goal is to have it set up to ensure a seamless installation and programming process for use in a closed or open loop. It is assumed that a plant operator would know and understand what a VFD is and what it does. There will be simple controls that the user can change to make the VFD do what they want it to do such as a potentiometer for controlling the frequency of output signals, and a start/stop button.

3.6 Support

We plan to create a manual describing what the VFD does and how to use it. The VFD will support usage, programming, and debugging through MPLAB X IDE. It will allow the user to control the motor's on/off function and its speed while showing the user several different variables including the measured motor speed from the tachometer.

4. Scenario(s)

4.1 HVAC

VFDs can be extremely useful in a Heating, Ventilation, and Air Conditioning (HVAC) system. VFDs can act as speed controllers for HVAC motors because they actively adjust the speed rate of those motors based on the building load demands. Without a VFD, an HVAC system will run at full power regardless of whether the system requires full power at that time. By running only as powerful as needed, energy and money are saved. Additionally, a VFD allows an HVAC motor to slowly get up to the full speed which is called a “soft start”. This lessens wear and tear on these HVAC motors, which helps them to last longer and saves money. VFDs can help HVAC systems to control temperature and pressure, among other measurements in a building, they help to save energy and extend the life of HVAC motors.

4.2 Conveyor Belt

In conveyors for production lines, a VFD allows an operator to control the belt's speed to match the production line's needs. This ensures that the belt is not moving too slow to decrease efficiency or too fast to risk injuring employees and consume copious amounts of energy. Wear and tear is also decreased by being able to run the belt at the most appropriate speed as to not overwork the parts or run the belt for too long. The slow start that a VFD can offer helps to reduce wear and tear on the conveyor parts so they can gradually get up to speed. There is the ability to gradually slow down the belt when turning off, which also extends the life of equipment. The VFD increases energy efficiency and reduces maintenance which both save money.

4.3 VFD Pump

VFDs are sought after for submersible pumps for several applications. The main benefit of a VFD in a water pump is the ability to keep a certain parameter constant. VFDs are commonly used in construction sites to keep groundwater at a constant level to build. The VFD does this by adjusting the speed of the motor controlling the water pump. Additionally, VFDs are useful in maintaining constant pressure in a water tank. VFDs are crucial in the operation of water pumps across many different disciplines because their speed control allows for constant parameters in the specified system.

5. Analysis

5.1 Summary of Proposed Improvements

Improvements:

- Extended life of parts: being able to vary frequency and torque help to reduce overloading equipment and, in turn, makes them last longer
- Increased efficiency: instead of a system having to run at a constant speed, a VFD allows the system to run only as fast as needed, which reduces energy consumption, and saves money
- Limits safety risks: having a system run at the proper speed and maintain a safe amount of torque helps decrease the risk of injury

5.2 Disadvantages and Limitations

Disadvantage:

- VFD can damage motor windings and bearings
- Can ruin insulation
- Creates harmonics (potentially interfering with communications and data processing)
- Can be very expensive
- Is prone to overheating if not in a properly ventilated area.
- Can create voltage spikes which could ruin the motor.

5.3 Alternatives

Alternatives to a VFD:

- Eddy Current Drive: induces a magnetic field that is adjustable to control the speed of a motor
- Soft Starters: Soft starters gradually increase the voltage to an AC motor during startup to reduce damage. They provide a more economical choice for applications where torque and speed control are only required during motor startup and stop.
- Cycloconverters: Cycloconverters convert the frequency of AC power from one frequency to another without having to convert to DC and back to AC. They are typically used in applications that require low speeds and high torque. This is a good alternative for when direct frequency conversion is more suitable than VFDs.

5.4 Impact

One environmental impact of a VFD is the ability to lower greenhouse gas emissions. By increasing the motor's energy efficiency through frequency adjustments to match load requirements, VFDs decrease the emissions released during the energy generation process.

A second environmental impact of a VFD is the degradation caused by mining the raw materials. While mining procures the rare metals needed for electronics, it also causes deforestation from the removal of forests, soil erosion from the disruption of the soil structure, and air pollution from the diesel emissions and dust particles generated by the machinery.

Another environmental impact of a VFD is the end-of-life disposal of electronic waste. Disposal of electronics can be challenging and, if done improperly, can contaminate soil and water with toxic metals. Alternatively, VFDs reduce this risk by reducing equipment wear, extending replacement intervals, and reducing disposal frequency.

A social impact of a VFD is the change in employment opportunities. The automation of the VFD reduces the need for operators who manually control the motor speed. Conversely, VFDs generate new roles including engineers who design the systems, miners who extract the raw materials, factory workers who assemble the components, and technicians who handle the maintenance.

Another social impact of a VFD is the safety risks. The VFD's automation of motor control within safe torque limits reduces the risk of accidents or injuries. However, if the VFD is improperly insulated or maintained, high voltages can pose a risk of electrical shock, and insulation breakdowns can pose a risk of short circuits and fire.

One ethical concern of a VFD is supply chain transparency. The raw materials for VFDs are often sourced from mines in countries with varying labor laws. In these mines, workers may face safety risks, including interacting with hazardous chemicals and dangerous machinery. Additionally, these mines should also avoid exploitative practices, including unfair compensation and unreasonable working hours.

A second ethical concern of a VFD is testing sufficiency. Before bringing a VFD to market, it should meet safety standards and demonstrate reliable operation. The testing process should include quality construction to prevent defects, comprehensive testing to identify defects, and clear documentation to resolve defects.

VFD Motor Controller

Mackenzie Miller
Andrew Nguyen
Aidan Rader
Ryan Regan

INTERFACE CONTROL DOCUMENT

REVISION – 0
26 September 2024

INTERFACE CONTROL DOCUMENT
FOR
VFD Motor Controller

PREPARED BY:

____
Author Date

APPROVED BY:

____
Project Leader Date

____
John Lusher II, P.E. Date

____
T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
0	9/26/2024	VFD Motor Controller Team		Draft Release
1	12/5/2024	VFD Motor Controller Team		Microcontroller and firmware descriptions updated
2	4/28/2025	VFD Motor Controller Team		Final Report Update

Table of Contents

List of Tables

List of Figures

No table of figures entries found.

1. Overview

This Interface Control Document (ICD) for the VFD Motor Controller provides an overview of the requirements and specifications for a Variable Frequency Drive that will be used to control a motor and it’s speed. Created by Ryan Regan, Andew Nguyen, Mackenzie Miller, and Aidan Rader, this document describes the physical, electrical, thermal, and communication interfaces for the system.

Key Sections:

- Physical Interface: This section depicts the physical aspects of the VFD including weight, dimensions, and physical/spatial requirements.

- Thermal Interface: The VFD includes a thermal monitoring system using the VFO pin and ITRIP pins that will shut the system down if the temperature levels are too high

- Electrical Interface: The electrical specifications include motor power, DC supply voltage, and the microcontroller’s stepped-down voltage supply. The DC voltage will be regulated by an H-bridge, and optoelectronic circuits will be used to isolate the microcontroller from the high voltage section of the system.

- User Interface: The VFD’s firmware allows the user to control the on/off functionality of the system as well as the speed of the motor with a button and potentiometer. The microcontroller can be plugged into a computer to view other different diagnostic feedback variables from the system.

- Communication Protocols: Optoelectronics will be used for communication between the high-voltage power control and the low-voltage microcontroller. Additionally, the system will use UART for serial communication and will have USB connectivity for programming and debugging.

2. References and Definitions

2.1 Definitions

AC	Alternating Current
DC	Direct Current
GUI	Graphical User Interface
ICD	Interface Control Document
MHz	Megahertz (1,000,000 Hz)
MCU	Micro Controller Unit
mA	Milliamp
mW	Milliwatt
N/A	Not Applicable
TBD	To Be Determined
VFD	Variable Frequency Drive

3. Physical Interface

3.1 Weight

The VFD motor controller may weigh up to 6lbs. The motor itself may weigh up to 13lbs.

3.2 Dimensions

3.2.1 Dimension of Optoelectronics and Feedback

Component	Diameter	Length	Width	Height
Digital Isolator	N/A	0.406”	0.406”	0.104”
Analog to Digital Optoisolator	N/A	0.442”	0.354”	0.158”
Operational Amplifier	N/A	0.382”	0.25”	0.400”
12 Pin Connector	N/A	0.854”	0.389”	0.271”
8 Pin Connector	N/A	1.089”	0.447”	0.272”

Table 1: Optoelectronics and Feedback Dimensions

3.2.2 Dimensions of MCU

Component	Diameter	Length	Width	Height
dsPIC33CK256MP508	N/A	0.394”	0.394”	0.039”
LM2595s-3.3 Buck Converter	N/A	0.400”	0.180”	0.450”
IRM 15-15 AC/DC Connector	N/A	2.06”	1.07”	0.945”
20 Pin connector	N/A	1.68”	0.276”	0.335”
5 Pin Connector	N/A	0.314”	0.433”	0.370”
3 Pin Connector	N/A	0.600”	0.322”	0.393”
Potentiometer	0.236” (actuator)	0.433”	0.386”	1.173”

Table 2: MCU Dimensions

The physical dimensions of the subsystem will be relative to the size of the user’s laptop and the size of the motor, both should not exceed the size of a standard tabletop.

3.2.3 Dimensions of Rectifier & DC Link

Component	Diameter	Length	Width	Height
VUE75-06NO7	N/A	1.850”	1.193”	0.799”
IKCM30F60GD	N/A	1.417”	0.827”	0.201”
Capacitor(s)	N/A	TBD	TBD	TBD
Inductor(s)	N/A	TBD	TBD	TBD

Table 3: Rectifier & DC Link Dimensions

4. Thermal Interface

4.1 Temperature Sensing

The power control shall monitor the temperature via the VFO pin, and if the VFD temperature rises above a safe value, the ITRIP pin will automatically shut the project down to avoid damaging parts.

5. Electrical Interface

5.1 Primary Input Power

The motor being used is 0.25HP producing 186.425W of power. The voltage supplied to the system is 295VDC which is then taken in by the H-bridge to generate a line-to-line voltage of 208V. 15VDC power is supplied to the MCU from the AC/DC power supply fed by the main power. This voltage will be stepped down to 3.3V for the MCU to use.

5.2 Voltage and Current Levels

Component	**Voltage (V)**	**Current (mA)**
dsPIC33CK	3.3V	50mA
IKCM30F60GD	600V	60A

Table 4: Maximum Voltage and Current Values

5.3 Signal Interfaces

Pulse Width Modules will be sent from the microcontroller to the power control of the VFD project using optoelectronic circuitry to ensure the microcontroller does not come into contact with the high voltage in the power control.
Three phase voltage values will be sent to the microprocessor from the power control using optoelectronic circuitry to ensure that the microprocessor does not come into contact with the high voltage in the power control.

5.4 User Control Interface

One feature of the VFD's microcontroller will be a potentiometer that allows the user to control the frequency of the three phase PWM's sine wave by increasing the step size that the program uses to step through a sine wave table. Increasing the PWM signal's frequency will increase the speed of the motor. The user will be able to change the frequency by turning a potentiometer. Another feature of the microcontroller is the start/stop button. This will give the user the option to turn the motor on and off without having to disconnect the system's power. In addition to the physical user interface, the VFD will display several debug variables to the UART console in MPLAB X IDE when connected to a computer. The user will be able to view the voltage, current, and temperature in a UART console.

Pin	Function
1	PWM1H
2	GPIO
3	PWM1L
4	GPIO
9	MCLR
11,32,50,70	VSS
12,31,51,71	VDD
17,22,39,44	GPIO
19	GPIO (Potentiometer)
24	GPIO (LED 2)
25	AVDD
26	AVss
34	Clock input
37	GPIO (LED 1)
42	GPIO (Relay)
60	PGD1
61	PGC1
68	UART TX
69	UART RX
75	PWM3H
76	PWM3L
78	PWM2H
80	PWM2L

Table 5: Pin interface for DSPIC33CK256MP508 Microcontroller

6. Communications / Device Interface Protocols

6.1 Optoelectronic Communications

The high voltage power control of the VFD will communicate to the low voltage or MCU via optoelectronics, or light to ensure that the MCU is not overpowered by the 120 VAC in the analog power control side.

6.2 Firmware and MCU Communications

The microcontroller is programmed with C code through MPLab's X IDE and will have several firmware implementations such as an on/off button's functionality and a three-phase PWM wave's frequency control.

6.3 Device Peripheral Interface

The MCU will use UART interface for serial communication between devices. This is what will allow a laptop to communicate with the microcontroller through USB connection.

6.4 Host Device

The C code shall be ran on MPLab's X IDE, which should be downloaded on the users computer. This will be used to program the microcontroller and output any debug variables necessary while the system is running.

VFD Motor Controller

Mackenzie Miller
Andrew Nguyen
Aidan Rader
Ryan Regan

FUNCTIONAL SYSTEM REQUIREMENTS

REVISION – 1
28 April 2025

Functional System Requirements

for

VFD Motor Controller

Prepared by:

____

Author Date

Approved by:

____

Project Leader Date

____

John Lusher, P.E. Date

____

T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
0	9/26/2024	VFD Motor Controller Team		Draft Release
1	12/5/2024	VFD Motor Controller Team		Reference documents updated
2	4/28/2025	VFD Motor Controller Team		Final Report Update

Table of Contents

List of Tables

List of Figures

1. Introduction

1.1. Purpose and Scope

The VFD motor controller is intended to control the speed and torque of an AC motor by varying the frequency and voltage supplied. This project will use three phase power as the input that will be converted to DC and then transmitted to a microcontroller via optoelectronics and used to power a motor. The microcontroller also will send signals to the DC link, again using optoelectronics, that contain the desired frequency. The VFD motor controller will come with a user interface that allows the user to input the frequency needed and start or stop the system as needed. This VFD shall be implemented to increase efficiency and save energy.

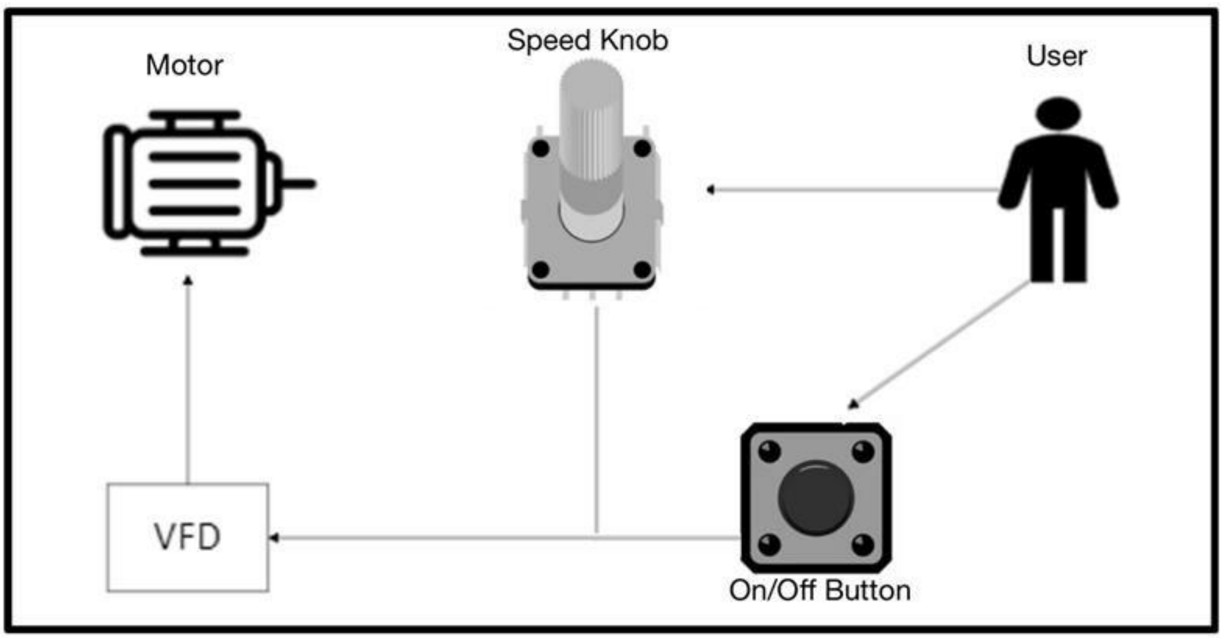

Figure 1: Project Conceptual Image

The following definitions differentiate between requirements and other statements.

Shall:	This is the only verb used for the binding requirements.
Should/May:	These verbs are used for stating non-mandatory goals.
Will:	This verb is used for stating facts or declaration of purpose.

1.2. Responsibility and Change Authority

Briefly describe who has the responsibility for making sure the requirements are met (i.e., team leader) and who has the authority to make the changes (i.e., client and team leader).

Subsystem	**Team Member**
Optoelectronics & Monitoring	Mackenzie Miller
Microcontroller	Andrew Nguyen
Rectifier & DC Link	Aidan Rader
Firmware	Ryan Regan

Table 1: Subsystem Designations

2. Applicable and Reference Documents

2.1 Applicable Documents

The following documents, of the exact issue and revision shown, form a part of this specification to the extent specified herein:

Document Number	Revision/Release Date	Document Title
IEEE 519-2014	3/27/2014	Standard for Harmonics
NFPA 70 Article 430	08/8/2023	Motors, Motor Circuits and Controllers

Table 2: Applicable Documents

2.2 Reference Documents

The following documents are reference documents utilized in the development of this specification. These documents do not form a part of this specification and are not controlled by their reference herein.

Document Number/ Publisher	Revision/Release Date	Document Title
Microsemi	Revision 0/11-03-2005	DM330030 Datasheet
Microchip Technology	Revision 11/06-2022	dsPIC33CK256MP508 Family Data Sheet
Texas Instruments	Revision 3/05-2016	LM2595 SIMPLE SWITCHER® Power Converter 150-kHz 1-A Step-Down Voltage Regulator datasheet
Skyworks	Revision B/09-2023	Si861x/2x Low-Power, Single- and Dual-Channel Digital Isolator
Infineon	Revision 2019-10-16	Control Integrated POwer System (CIPOS™) IKCM30F60GD
IXYS	Revision 2021	VUE75-06NO7
Mean Well	Revision 10-30-2024	IRM-15 series Datasheet

Table 3: Reference Documents

2.3 Order of Precedence

In the event of a conflict between the text of this specification and an applicable document cited herein, the text of this specification takes precedence without any exceptions.

All specifications, standards, exhibits, drawings or other documents that are invoked as “applicable” in this specification are incorporated as cited. All documents that are referred to within an applicable report are for guidance and information only, except ICDs that have their relevant documents considered to be incorporated as cited.

3. Requirements

3.1 System Definition

The VFD motor controller is comprised of a DC link (includes a rectifier and DC bus), power controller, microcontroller, and optoelectronics. The DC link converts AC input into DC and minimizes the noise of the DC signal. The power controller controls the voltage. The microcontroller is used to produce PWM signals that will drive the motor. Optoelectronics allows for communication between the high voltage side and the low voltage microcontroller.
This is a four-person project split into these roles:

Firmware (Ryan Regan):
Write code in C using MPLab X IDE that programs the microcontroller so that it functions with the rest of the VFD system according to desired specifications. In other words, program a potentiometer to allow user to change the frequency of a three-phase PWM signal. The firmware will also implement debug statements that output to MPLAB X IDE's UART console to allow user to view different variables as the program is running. Before integrating with the microcontroller and other subsystems, the firmware will be tested on a development board to ensure functionality.

Optoelectronics (Mackenzie Miller):
The first part of the sensors portion includes two optoelectronic circuits. One of those takes digital signals in and directly converts them to analog. The second, being more complicated, takes analog input, then, using an opto-isolator, the circuit converts the input to digital signals to be sent to the microcontroller. The analog side of the project works with very high voltage, while the digital side works with low voltage. Because of this, the two sides cannot be connected or the microcontroller will be overpowered and break, so the optoelectronics send signals and data across using light. The second part of the sensors portion is a constant current and voltage measurement. This will be done using a current sensing resistor, which has a very small resistance, and a simple voltage divider circuit. These measurements will then be sent back to the firmware. Finally, there will be a tachometer that measures the RPMs of the motor that is also sent to the firmware.

Microcontroller (Andrew Nguyen):
The MCU is supplied by the 15 VDC from the AC power supply fed by the main power. The voltage will then be stepped down to a usable 3.3V. It receives feedback through low-voltage analog signals that represent voltage and current. The MCU will then send out PWM signals to the H bridge and power control system which are used to control the inverter stage of the VFD which helps to adjust the output voltage and frequency supplied to the motor.

Rectifier and DC Link (Aidan Rader):

The rectifier takes in three-phase AC power and converts it into DC power. For each phase, there are parallel diodes acting as a one-way bridge for the current allowing it to flow in only one direction. To maintain the correct current polarity the diode opens and closes in sequence as the AC waveform alternates. The DC output is then filtered by capacitors within the DC link to provide a stable DC voltage for the microprocessor.

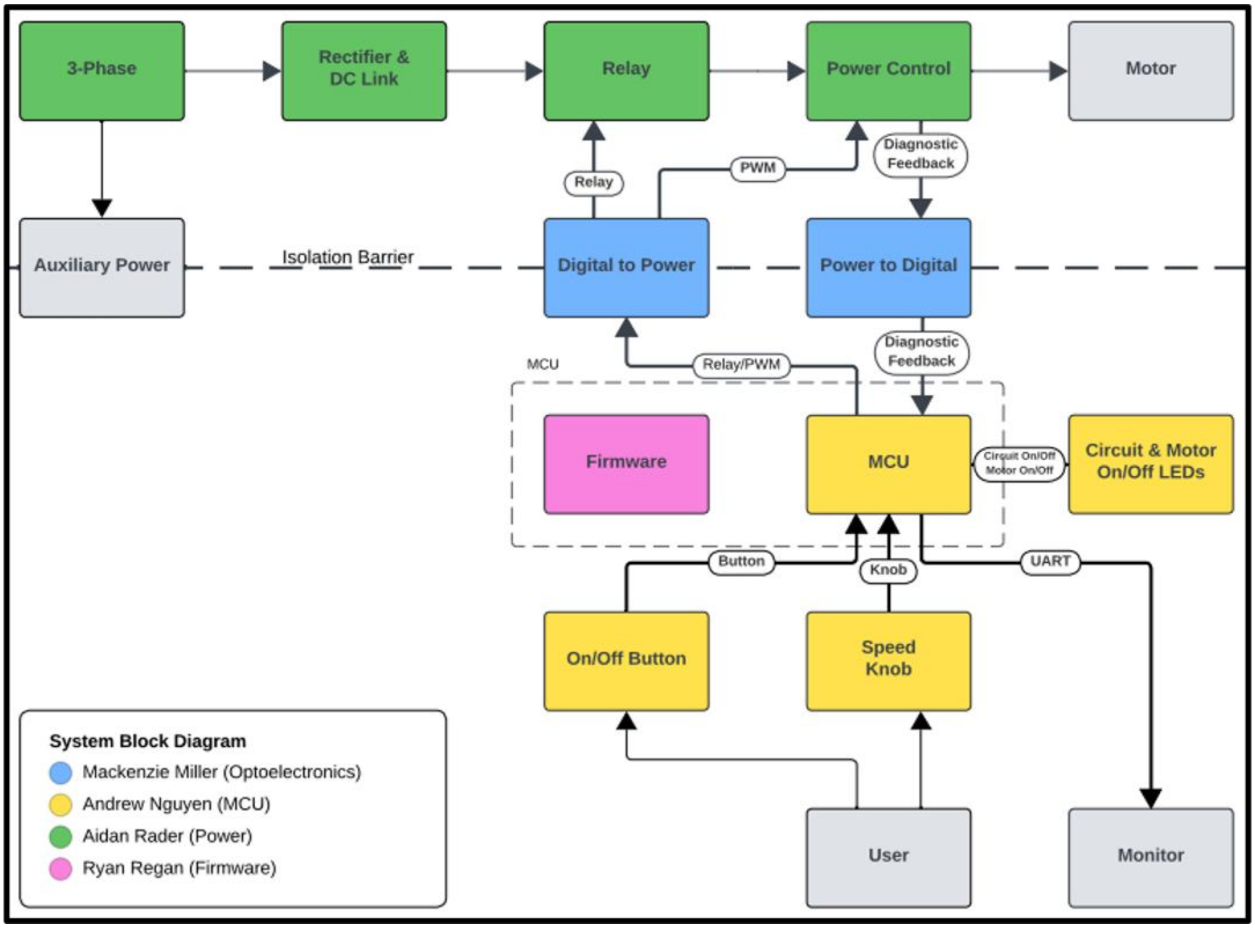

Figure 2: Block Diagram of System

3.2 Characteristics

3.2.1 Functional / Performance Requirements

3.2.1.1 Speed and Torque Requirement

The 0.25HP VFD motor shall run up to 1800RPM. The torque requirement for this is 73lb-ft of torque.

3.2.1.2 Frequency Requirement

The VFD motor controller shall be able to handle frequencies from 5Hz to 60Hz.

Rationale: American standard frequency for alternating current

3.2.1.3 Temperature Requirement

The VFD motor controller shall be able to operate at temperatures ranging from 0°C to 70°C.

Rationale: Commercial temperature rating standards

3.2.2 Physical Characteristics

3.2.2.1 System Components

The system shall consist of three printed circuit boards (PCBs) connected by wires.

3.2.3 Electrical Characteristics

3.2.3.1 Inputs

The presence or absence of any combination of the input signals in accordance with ICD specifications applied in any sequence shall not damage the VFD motor controller, reduce its life expectancy, or cause any malfunction, either when the unit is powered or when it is not.

No sequence of command shall damage the VFD, reduce its life expectancy, or cause any malfunction.

Rationale: By design, should limit the chance of damage or malfunction by user/technician error.

3.2.3.2 Input Voltage Level

The input voltage level for the VFD shall be three phase 120 VAC.

Rationale: VFD specification, Sponsor

3.2.3.3 Input Noise and Ripple

The maximum ripple allowed on the dsPIC33CK is 165mV peak to peak.

Rationale: The specifications from the dsPIC33CK datasheet at 3.3V

3.2.3.4 External Commands

The VFD Motor Control Team shall document all external commands in the appropriate ICD.

Rationale: The ICD will capture all interface details from the low level electrical to the high-level packet format.

3.2.3.5 Visual Output

The VFD shall include a Graphical User Interface that displays output measurements.

Rationale: Provides the ability to see the outputs when VFD is running

3.2.3.6 Connectors

The VFD shall use terminal blocks for all power and signal connections to ensure secure and vibration-resistant connections.

Rationale: Noise must be limited to maintain signal integrity

3.2.3.7 Failure Propagation

3.2.3.7.1 Overtemperature Shutdown

The VFD shall have a sensor that automatically shuts the system down if the temperature rises above 70 degrees Celsius.

3.2.3.8 Built in Test (BIT)

The VFD shall have an internal subsystem that will generate test signals and evaluate the VFD responses and determine if there is a failure.

3.2.3.9 Digital to Power Continuity

The digital to power opto-isolators shall have a voltage of 0V across each component when measuring from input to output.

3.2.3.10 Power to Digital Continuity

The power to digital opto-isolators shall have a voltage of 0V across each component when measuring from input to output.

3.2.3.11 15 V_{DC} to Isolated 15 V_{DC} Conversion

Power converters shall convert 15 V_{DC} to isolated 15 V_{DC}.

3.2.3.12 15 V_{DC} to 3.3 V_{DC} Conversion

Power converters shall convert 15 V_{DC} to 3.3 V_{DC}.

3.2.3.13 3.3 V_{DC} to Isolated 5 V_{DC} Conversion

Power converters shall convert 3.3 V_{DC} to isolated 5 V_{DC}.

3.2.3.14 Low Voltage Auxiliary Power

System shall convert 15 V_{DC} to isolated 15 V_{DC} , 3.3 V_{DC}, and isolated 5 V_{DC}.

3.2.3.15 120 V_{AC} to 15 V_{DC} Conversion

Power converters shall convert 120 V_{AC} to isolated 15 V_{DC}.

3.2.3.16 120 V_{AC} Auxiliary Power

System shall convert 120 V_{AC} to 15 V_{DC}, isolated 15 V_{DC} , 3.3 V_{DC}, and isolated 5 V_{DC}.

3.2.3.17 Relay

Relay shall toggle the motor with the relay signal.

3.2.3.18 Relay Signal Isolation

Digital to power shall convert the relay signal from 3.3 V_{DC} to 5 V_{DC}.

3.2.3.19 Relay Signal Generation
MCU shall toggle the relay signal to 3.3 V_{DC} with the button signal.

3.2.3.20 Firmware Relay Signal Generation
Dev board shall toggle the relay signal to 3.3 V_{DC} with the button signal.

3.2.3.21 Button Signal Generation
Button shall toggle the button signal to 3.3 V_{DC} with a button press.

3.2.3.22 On/Off Button
System shall toggle the motor between on and off state with a button press.

3.2.3.23 120 VAC to 112 VDC Rectification
Rectifier shall convert 120 V_{AC} to 112 V_{DC}.

3.2.3.24 120 V_{AC} to 112 V_{DC} Rectification
Rectifier shall convert 120 V_{AC} to 112 V_{DC}.

3.2.3.25 Power Control
Power control shall invert three high and three low 5 V_{DC} PWM signals into three modified sine waves.

3.2.3.26 PWM Signal Isolation
Digital to power shall convert three high and three low PWM signals from 3.3 V_{DC} to 5 V_{DC}.

3.2.3.27 PWM Signal Generation
MCU shall generate three high and three low 3.3 V_{DC} PWM signals with the knob signal.

3.2.3.28 Knob Signal Generation
Rotating potentiometer changes frequency to speed up and slow down motor.

3.2.3.29 Firmware PWM Signal Generation
Dev board shall generate three high and three low 3.3 V_{DC} PWM signals with the knob signal.

3.2.3.30 Firmware Knob Signal Generation
Rotating dev board potentiometer changes frequency to speed up and slow down motor.

3.2.3.31 Low Voltage PWM Control
System shall change the motor speed with the knob signal.

3.2.3.32 112 V_{DC} PWM Control
System shall change the motor speed with the knob signal.

3.2.3.33 Motor On/Off Signal Generation
MCU shall toggle the motor on/off LED signal to 3.3 V_{DC} with the button signal.

3.2.3.34 Motor On/Off LED

System shall toggle the motor on/off LED between on and off states with the motor on/off signal.

3.2.3.35 System On/Off Signal Generation
MCU shall toggle the circuit on/off LED signal to 3.3 V_{DC} when the board receives power.

3.2.3.36 System On/Off LED
System shall toggle the circuit on/off LED when the board is connected to power with the circuit on/off signal

3.2.3.37 Firmware Motor On/Off LED Signal Generation
Firmware shall toggle the dev board's on/off LED signal to 3.3 V_{DC} with the button signal.

3.2.3.38 Firmware Motor On/Off LED
Firmware shall toggle the dev board's on/off LED between on and off states with the motor on/off signal.

3.2.3.39 Firmware System LED Signal Generation
Firmware shall toggle the dev board's on/off LED signal to 3.3 V_{DC} when the board receives power.

3.2.3.40 Firmware System LED
Firmware shall toggle the dev board's on/off LED when the board is connected to power with the circuit on/off signal

3.2.3.41 Computer Display
MCU will display diagnostic feedback information to the UART console on a computer running MPLAB X IDE

3.2.3.42 Diagnostic Feedback Signal Isolation
Diagnostic feedback circuit shall transport the voltage, current, and temperature signals from across the isolation barrier to the MCU.

3.2.3.43 Diagnostic Feedback Signal Generation
Power Control shall generate 60 V_{DC} voltage, current, and temperature feedback signals.

3.2.3.44 Diagnostic Feedback
System shall send the voltage, current, and temperature signals to the computer display.

Appendix A: Acronyms and Abbreviations

BIT	Built-In Test
CCA	Circuit Card Assembly
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
EO/IR	Electro-optical Infrared
FOR	Field of Regard
FOV	Field of View
GPS	Global Positioning System
GUI	Graphical User Interface
Hz	Hertz
ICD	Interface Control Document
kHz	Kilohertz (1,000 Hz)
LCD	Liquid Crystal Display
LED	Light-emitting Diode
mA	Milliamp
MHz	Megahertz (1,000,000 Hz)
MTBF	Mean Time Between Failure
MTTR	Mean Time To Repair
mW	Milliwatt
PCB	Printed Circuit Board
RMS	Root Mean Square
TBD	To Be Determined
TTL	Transistor-Transistor Logic
USB	Universal Serial Bus
VFD	Variable Frequency Drive
VME	VERSA-Module Europe

Appendix B: Definition of Terms

VFD Motor Controller

Mackenzie Miller
Andrew Nguyen
Aidan Rader
Ryan Regan

EXECUTION AND VALIDATION PLAN

REVISION – 1
28 April 2025

Execution Plan

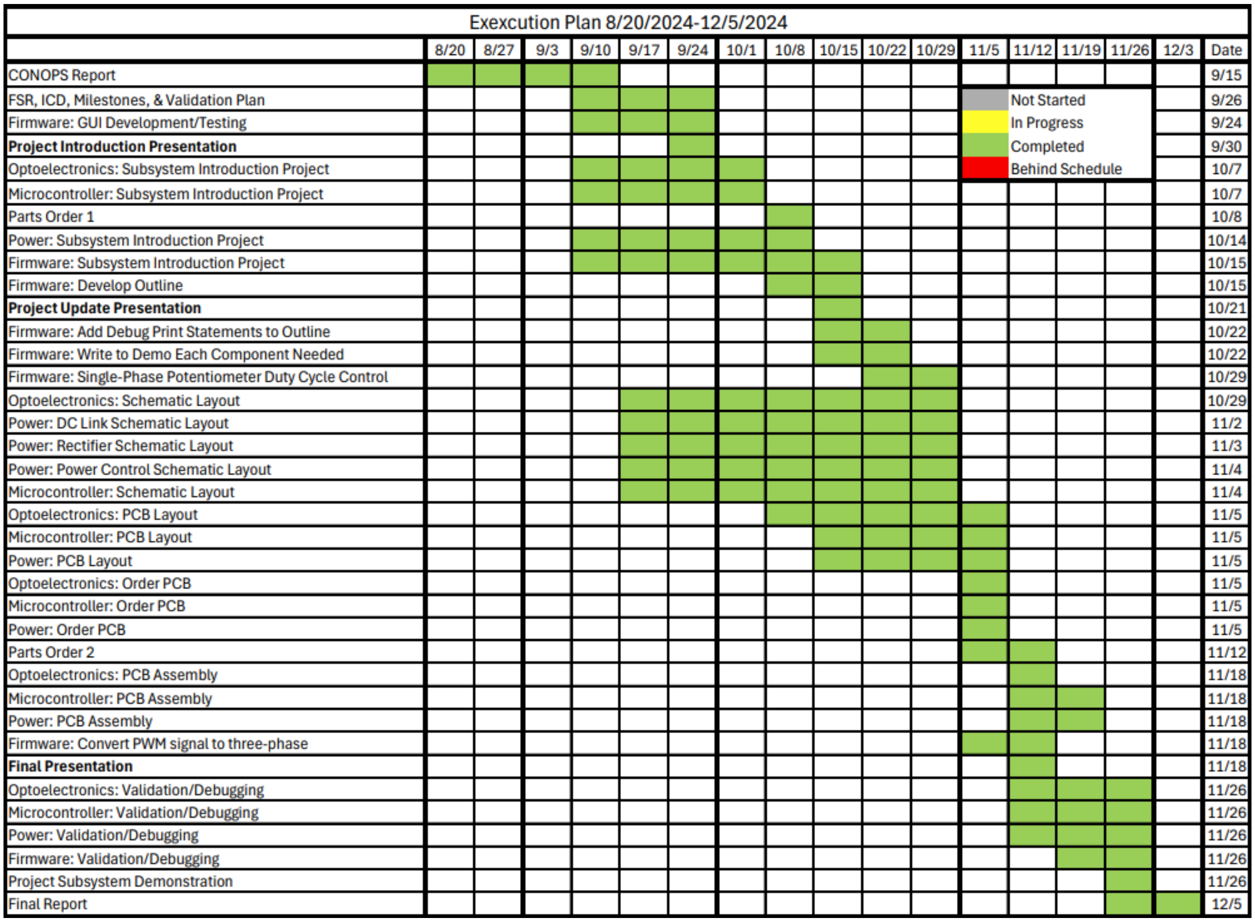

Figure 1: Execution Plan Fall 2024

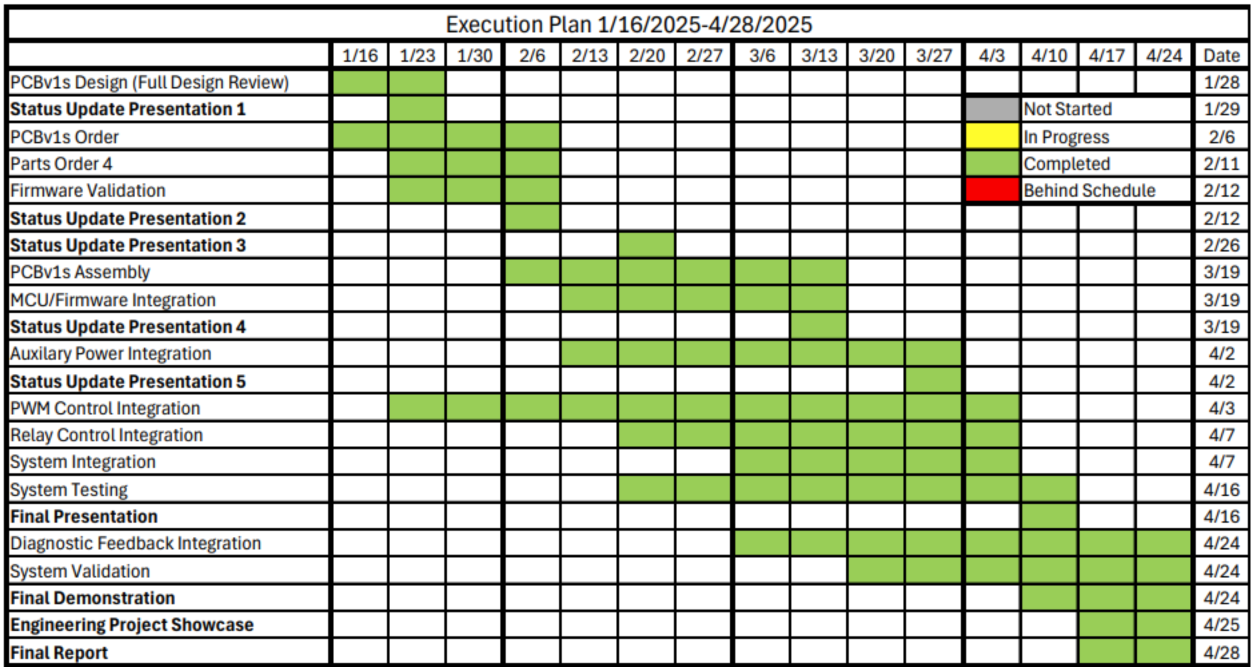

Figure 2: Execution Plan Spring 2025

Validation Plan

Paragraph #	Test Name	Success Criteria	Methodology	Status	Responsible Engineer(s)
3.2.3.11	15 V_{DC} to Isolated 15 V_{DC} Conversion	Power converters shall convert 15 V_{DC} to isolated 15 V_{DC}.	Apply 15 V_{DC} input using a DC power supply. Verify 15 V_{DC} output using an oscilloscope.	Tested	Aidan
3.2.3.12	15 V_{DC} to 3.3 V_{DC} Conversion	Power converters shall convert 15 V_{DC} to 3.3 V_{DC}.	Apply 15 V_{DC} input using a DC power supply. Verify 3.3 V_{DC} output using an oscilloscope.	Tested	Drew
3.2.3.13	3.3 V_{DC} to Isolated 5 V_{DC} Conversion	Power converters shall convert 3.3 V_{DC} to isolated 5 V_{DC}.	Apply 3.3 V_{DC} input using a DC power supply. Verify 5 V_{DC} output using an oscilloscope.	Tested	Aidan
3.2.3.14	Low Voltage Auxiliary Power	System shall convert 15 V_{DC} to isolated 15 V_{DC} , 3.3 V_{DC},	Apply 15 V_{DC} input using a DC power supply. Verify 15 V_{DC} , 3.3 V_{DC}, and 5	Tested	Mackenzie, Drew, Aidan

		and isolated 5 V_{DC}.	V_{DC} outputs using an oscilloscope.		
3.2.3.15	120 V_{AC} to 15 V_{DC} Conversion	Power converters shall convert 120 V_{AC} to isolated 15 V_{DC}.	Apply 120 V_{AC} input using a 120 V_{AC} outlet. Verify 15 V_{DC} output using an oscilloscope.	Tested	Drew
3.2.3.16	120 V_{AC} Auxiliary Power	System shall convert 120 V_{AC} to 15 V_{DC}, isolated 15 V_{DC} , 3.3 V_{DC}, and isolated 5 V_{DC}.	Apply 120 V_{AC} input using a 120 V_{AC} outlet. Verify 15 V_{DC}, 15 V_{DC} , 3.3 V_{DC}, and 5 V_{DC} outputs using an oscilloscope.	Tested	Mackenzie, Drew, Aidan
3.2.3.17	Relay	Relay shall toggle the motor with the relay signal.	Apply 15 V_{DC} coil voltage, 10 V_{DC} contact voltage, and 5 V_{DC} relay signal using a DC power supply. Verify 10 V_{DC} output using an oscilloscope.	Tested	Aidan
3.2.3.18	Relay Signal Isolation	Digital to power shall convert the relay signal from 3.3 V_{DC} to 5 V_{DC}.	Apply 5 V_{DC} supply voltage, 3.3 V_{DC} supply voltage, and 3.3 V_{DC} relay signal using a DC power supply. Verify 5 V_{DC} output using an oscilloscope.	Tested	Mackenzie
3.2.3.19	Relay Signal Generation	MCU shall toggle the relay signal to 3.3 V_{DC} with the button signal.	Apply 3.3 V_{DC} supply voltage and 3.3 V_{DC} button signal using a DC power supply. Verify 3.3 V_{DC} output using an oscilloscope.	Tested	Drew, Ryan
3.2.3.20	Firmware Relay Signal Generation	Dev board shall toggle the relay signal to 3.3	(On the dev board) Apply 3.3 V_{DC} supply voltage and 3.3 V_{DC} button signal using	Tested	Ryan

		V_{DC} with the button signal.	a DC power supply. Verify 3.3 V_{DC} output using an oscilloscope.		
3.2.3.21	Button Signal Generation	Button shall toggle the button signal to 3.3 V_{DC} with a button press.	Apply 3.3 V_{DC} supply voltage using a DC power supply and initiate a button press. Verify 3.3 V_{DC} output using an oscilloscope.	Tested	Drew
3.2.3.22	On/Off Button	System shall toggle the motor between on and off state with a button press.	Apply 15 V_{DC} coil voltage, 10 V_{DC} contact voltage, 5 V_{DC} supply voltage, and 3.3 V_{DC} supply voltage using a DC power supply. Initiate one button press and later a second button press. Verify the motor rotates after the first button press and stops after the second button press using a video.	Tested	All
3.2.3.23	10 VAC to 12 VDC Rectification	Rectifier shall convert 120 V_{AC} to 112 V_{DC}.	Apply 120 V_{AC} input using a 120 V_{AC} outlet. Verify 112 V_{DC} output using an oscilloscope.	Tested	Aidan
3.2.3.24	120 V_{AC} to 112 V_{DC} Rectification	Rectifier shall convert 120 V_{AC} to 112 V_{DC}.	Apply 120 V_{AC} input using a 120 V_{AC} outlet. Verify 112 V_{DC} output using an oscilloscope.	Tested	Aidan
3.2.3.25	Power Control	Power control shall invert three high and three low 5 V_{DC} PWM	Apply 40 V_{DC} bus voltage and 15 V_{DC} supply voltage using a DC power supply. Apply three high	Tested	Aidan

		signals into three modified sine waves.	and three low 5 V_{DC} PWM signals at 120˚ phase shifts using a function generator. Verify three modified sine waves at 120˚ phase shifts using an oscilloscope.		
3.2.3.26	PWM Signal Isolation	Digital to power shall convert three high and three low PWM signals from 3.3 V_{DC} to 5 V_{DC}.	Apply isolated 5 V_{DC} supply voltage and 3.3 V_{DC} supply voltage using a DC power supply. Apply three high and three low 3.3 V_{DC} PWM signals at 120˚ phase shifts using a function generator. Verify three high and three low 5 V_{DC} PWM signals at 120˚ phase shifts using an oscilloscope.	Tested	Mackenzie
3.2.3.27	PWM Signal Generation	MCU shall generate three high and three low 3.3 V_{DC} PWM signals with the knob signal.	Apply 3.3 V_{DC} supply voltage and 3.3 V_{DC} speed knob signal using a DC power supply. Verify three high and three low 3.3 V_{DC} PWM signals at 120˚ phase shifts using an oscilloscope.	Tested	Drew, Ryan
3.2.3.28	Knob Signal Generation	Rotating potentiometer changes frequency to speed up and slow down motor.	Oscilloscope each signal output, spin knob and ensure the motor speed varies as expected.	Tested	Drew, Ryan

3.2.3.29	Firmware PWM Signal Generation	Dev board shall generate three high and three low 3.3 V_{DC} PWM signals with the knob signal.	(On dev board) Apply 3.3 V_{DC} supply voltage and 3.3 V_{DC} speed knob signal using a DC power supply. Verify three high and three low 3.3 V_{DC} PWM signals at 120˚ phase shifts using an oscilloscope.	Tested	Ryan
3.2.3.30	Firmware Knob Signal Generation	Rotating dev board potentiometer changes frequency to speed up and slow down motor.	(On dev board) Oscilloscope each signal output, spin knob and ensure the motor speed varies as expected.	Tested	Ryan
3.2.3.31	Low Voltage PWM Control	System shall change the motor speed with the knob signal.	Apply 40 V_{DC} bus voltage, 15 V_{DC} supply voltage, isolated 5 V_{DC} supply voltage and 3.3 V_{DC} supply voltage using a DC power supply. Verify the motor rotates at a different speed after a knob turn using a video.	Tested	All
3.2.3.32	112 V_{DC} PWM Control	System shall change the motor speed with the knob signal.	Apply 112 V_{DC} bus voltage, 15 V_{DC} supply voltage, isolated 5 V_{DC} supply voltage and 3.3 V_{DC} supply voltage using a DC power supply. Verify the motor rotates at a different speed	Untested	All

			after a knob turn using a video.		
3.2.3.33	Motor On/Off Signal Generation	MCU shall toggle the motor on/off LED signal to 3.3 V_{DC} with the button signal.	Directly inject 15V to MCU and use multimeter at the motor LED via.	Tested	Drew, Ryan
3.2.3.34	Motor On/Off LED	System shall toggle the motor on/off LED between on and off states with the motor on/off signal.	Directly inject 15V to MCU, then press relay toggle button and ensure the motor LED toggles.	Tested	Drew, Ryan
3.2.3.35	System On/Off Signal Generation	MCU shall toggle the circuit on/off LED signal to 3.3 V_{DC} when the board receives power.	Directly inject 15V to MCU and use multimeter at the system LED via.	Tested	Drew, Ryan
3.2.3.36	System On/Off LED	System shall toggle the circuit on/off LED when the board is connected to power with the circuit on/off signal	Directly inject 15V to MCU and ensure the system LED turns on.	Tested	Drew, Ryan
3.2.3.37	Firmware Motor On/Off LED Signal Generation	Firmware shall toggle the dev board's on/off LED signal to 3.3 V_{DC} with the button signal.	Directly inject 15V to dev board and use multimeter at the motor LED via.	Tested	Ryan

3.2.3.38	Firmware Motor On/Off LED	Firmware shall toggle the dev board's on/off LED between on and off states with the motor on/off signal.	Directly inject 15V to dev board, then press relay toggle button and ensure the motor LED toggles.	Tested	Ryan
3.2.3.39	Firmware System LED Signal Generation	Firmware shall toggle the dev board's on/off LED signal to 3.3 V_{DC} when the board receives power.	Directly inject 15V to dev board and use multimeter at the system LED via.	Tested	Ryan
3.2.3.40	Firmware System LED	Firmware shall toggle the dev board's on/off LED when the board is connected to power with the circuit on/off signal	Directly inject 15V to dev board and ensure the system LED turns on.	Tested	Ryan
3.2.3.41	Computer Display	MCU will display diagnostic feedback information to the UART console on a computer running MPLAB X IDE.	Connect system to MPLAB X IDE using a laptop and USB-UART cable. Verify diagnostic print statements are being output to UART console.	Untested	Drew, Ryan
3.2.3.42	Diagnostic Feedback Signal Isolation	Diagnostic feedback circuit shall transport the voltage, current, and temperature signals from	Apply 15 VDC supply voltage, isolated 15 VDC supply voltage, and three 15 VDC feedback signals using a DC power supply. Verify three	Untested	Mackenzie

		across the isolation barrier to the MCU.	15 VDC outputs using an oscilloscope.		
3.2.3.43	Diagnostic Feedback Signal Generation	Power Control shall generate 60 V_{DC} voltage, current, and temperature feedback signals.	Apply 40 V_{DC} bus voltage and 15 V_{DC} supply voltage using a DC power supply. Apply three high and three low 5 V_{DC} PWM signals at 120˚ phase shifts using a function generator. Verify three 60 V_{DC} outputs using an oscilloscope.	Untested	Aidan
3.2.3.44	Diagnostic Feedback	System shall send the voltage, current, and temperature signals to the computer display.	Each feedback signal will be verified through the UART console output.	Untested	All

VFD Motor Controller

Mackenzie Miller
Andrew Nguyen
Aidan Rader
Ryan Regan

SUBSYSTEM REPORT

REVISION – 1
28 April 2025

SUBSYSTEM REPORT

FOR

VFD Motor Controller

PREPARED BY:

Author Date

APPROVED BY:

Project Leader Date

John Lusher, P.E. Date

T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
0	12/5/2024	VFD Motor Controller Team		Draft Release
1	4/28/2025	VFD Motor Controller Team		Final Report Update

Table of Contents

List of Tables

List of Figures

1. Introduction

The VFD motor controller is designed to regulate the speed and torque of an AC motor by adjusting the frequency and voltage it receives. This project utilizes three-phase power as the input, which will be converted to DC and transmitted a microcontroller to power the motor. The microcontroller will also send and receive signals, via optoelectronics, to the DC link to deliver the user-specified frequency set through the user interface. The VFD is intended to enhance efficiency and conserve energy.

2. Optoelectronics Subsystem Report

2.1. Subsystem Introduction

The optoelectronics subsystem of the VFD serves as the go between for the microcontroller and the motor and power electronics. The microcontroller and high voltage side cannot encounter one another because the microcontroller operates at 3.3V, and the power electronics and motor operate at 120 VAC from the wall. Optoelectronics help transport voltages from the power control to the microcontroller and PWMs from the microcontroller to the power control.

2.2. Subsystem Details

The power to digital circuitry of this subsystem includes three circuits that convert signals that represent the voltage, current, and temperature of the system that are received to around 15V so the microcontroller (MCU) can process them. The first circuit takes the output of the DC rectifier, the second circuit is for the current and voltage monitoring, and the third circuit is for temperature control. These values are to be processed by the MCU and displayed on a UART console so the system can be monitored. These measurements are transported across an isolation barrier using light via emitters and receivers made into an integrated circuit. Below, in Figure 1, one of the circuits can be seen. The PCB realization of this is seen in Figure 3.

There are also three circuits that take pulse width modules (PWMs) in and transport them across another isolation barrier. The three isolators have been separated as follows: the first one transports PWM 1H and 1L and the relay signal, the second transports 2H and 2L, and the third is for 3H and 3L. The six PWMs are created by the MCU and are sent out at 3.3V. The isolators take in 3.3V and output the same PWMs at 5V to the power side. This segment is necessary in order to maintain the integrity of separating the power and digital sides. The schematic of this circuit can be seen below in Figure 2. The PCB realization of this is seen in Figure 4.

In the beginning of this project, there was to be a tachometer as a part of this subsystem, but near the end, it was cut due to timing. Had there been two or three more weeks, there could have been a tachometer to measure the revolutions per minute (RPMs) of the motor. This is not necessary for the completion of the project, but it would have been an interesting additional feature.

These circuits are executed on a printed circuit board (PCB) which yields a very compact design with many components.

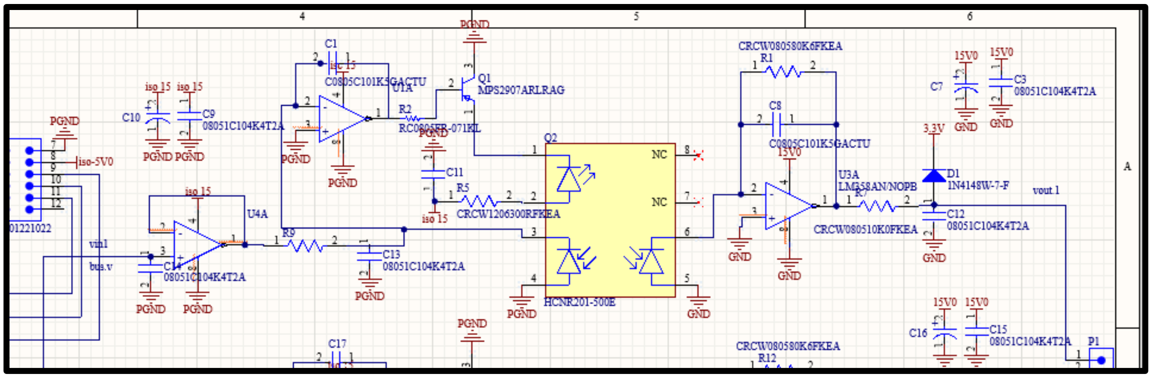

Figure 1: Power to Digital Schematic

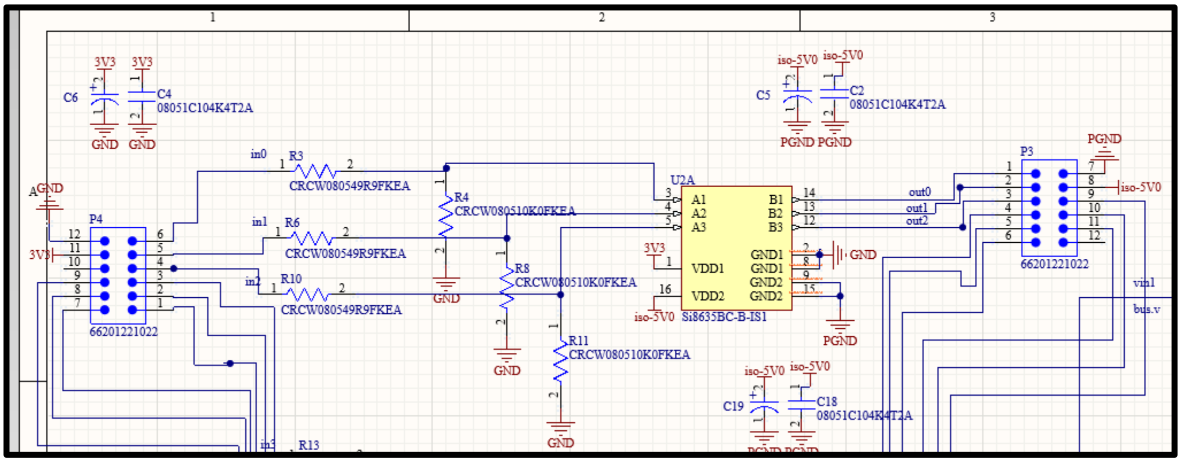

Figure 2: Digital to Power Schematic

2.3. Subsystem Validation

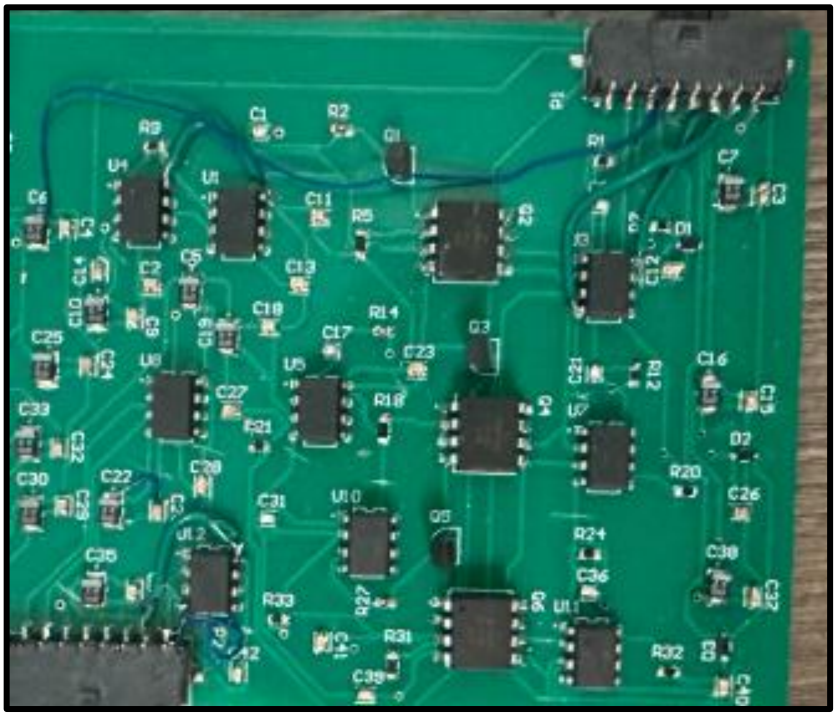

Figure 3: Power to Digital Portion of PCB

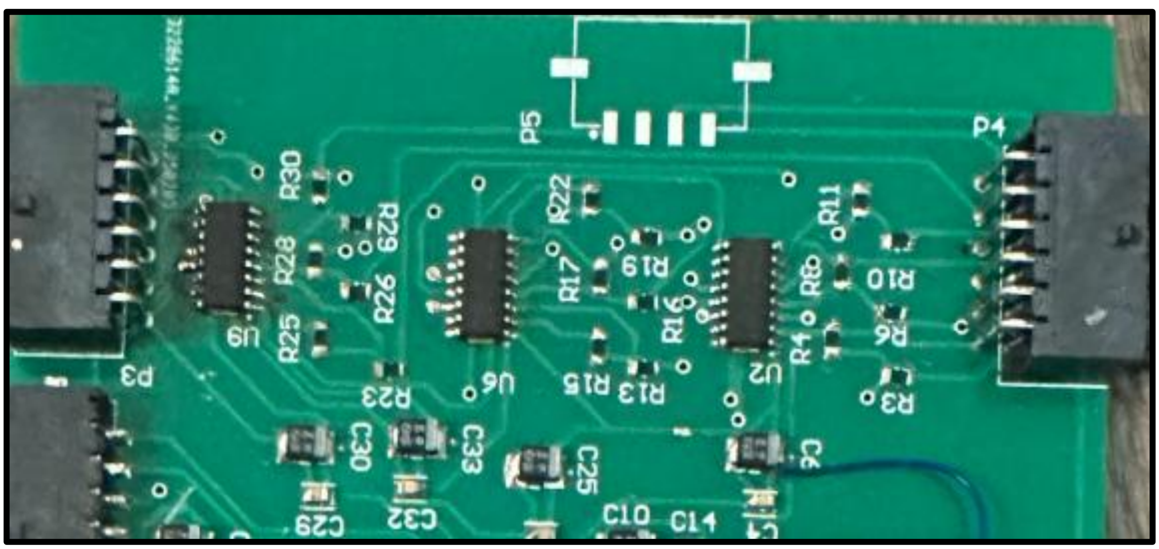

Figure 4: Digital to Power Portion of PCB

This subsystem consisted of three different PCB designs. The first one lasted the whole first semester, and it worked pretty well. There were no trace errors that were known at the time, so when it came time to order a second one, the only change was the connector locations. The second PCB survived PWM integration and all associated testing. When attempting to test auxiliary power, a large voltage drop was discovered on this subsystem. The auxiliary power went through the power and MCU boards successfully, and this board was the final link. Based on this, the opto board was thought to be the issue. Along with this problem, the operational amplifiers (op-amps) were getting extremely hot. Originally, it was thought that a capacitor went out and was causing a short. The problem, however, was much larger than that. The Vcc and GND pins on each of nine op-amps were switched which caused the large voltage drop and essentially circuit failure. A third PCB was designed to fix this problem, but an incorrect version of the files, possibly unsaved, was submitted which led to a bad board being printed. Due to the short amount of time left in the semester, the best option available was to cut the incorrect traces with a knife and externally connect the Vcc and GND the correct way. This can be seen in Figure 5 below.

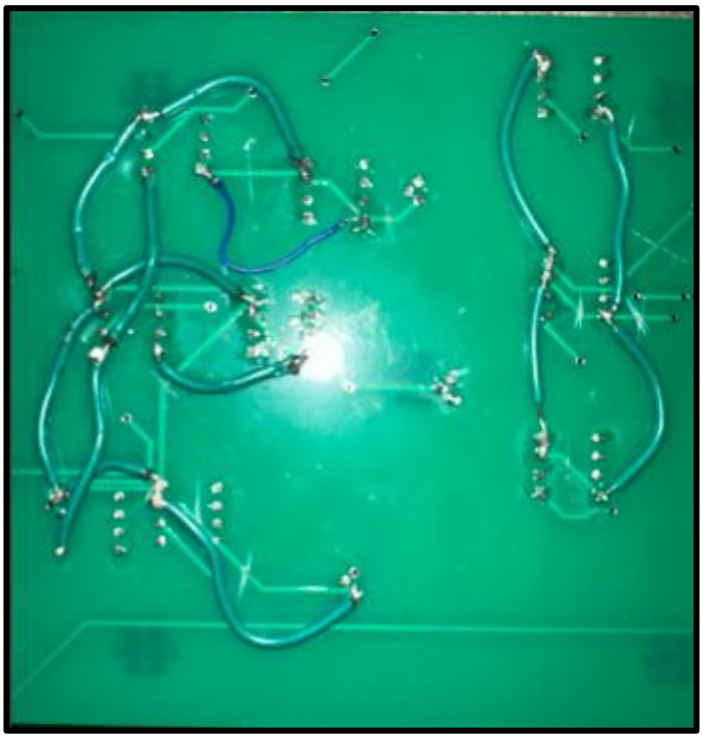

Figure 5: *Cut Traces and External Wires to Fix Op-Amps*

2.3.1. Digital to Power Isolation Validation

The first part of this subsystem that was tested and then integrated was the PWM isolation. When this portion was tested last semester, the isolators fried due to a testing error, so the parts were replaced and tested with PWMs coming out of the development (dev) board. The purpose of this board was to bypass the MCU until it was ready. The first time the opto PCB was connected to the dev board for the PWM input, it worked perfectly. The 3.3V input PWMs were converted to 5V PWM wave outputs for all three phases high and low. When it came time to test this with the MCU, the same result was observed. With the MCU, the input PWMs were 3.4V, and the outputs were 6.0V to 6.2V as seen in Figures 6 and 7. This is slightly higher than expected, but it was within the requirements, so these results were successful.

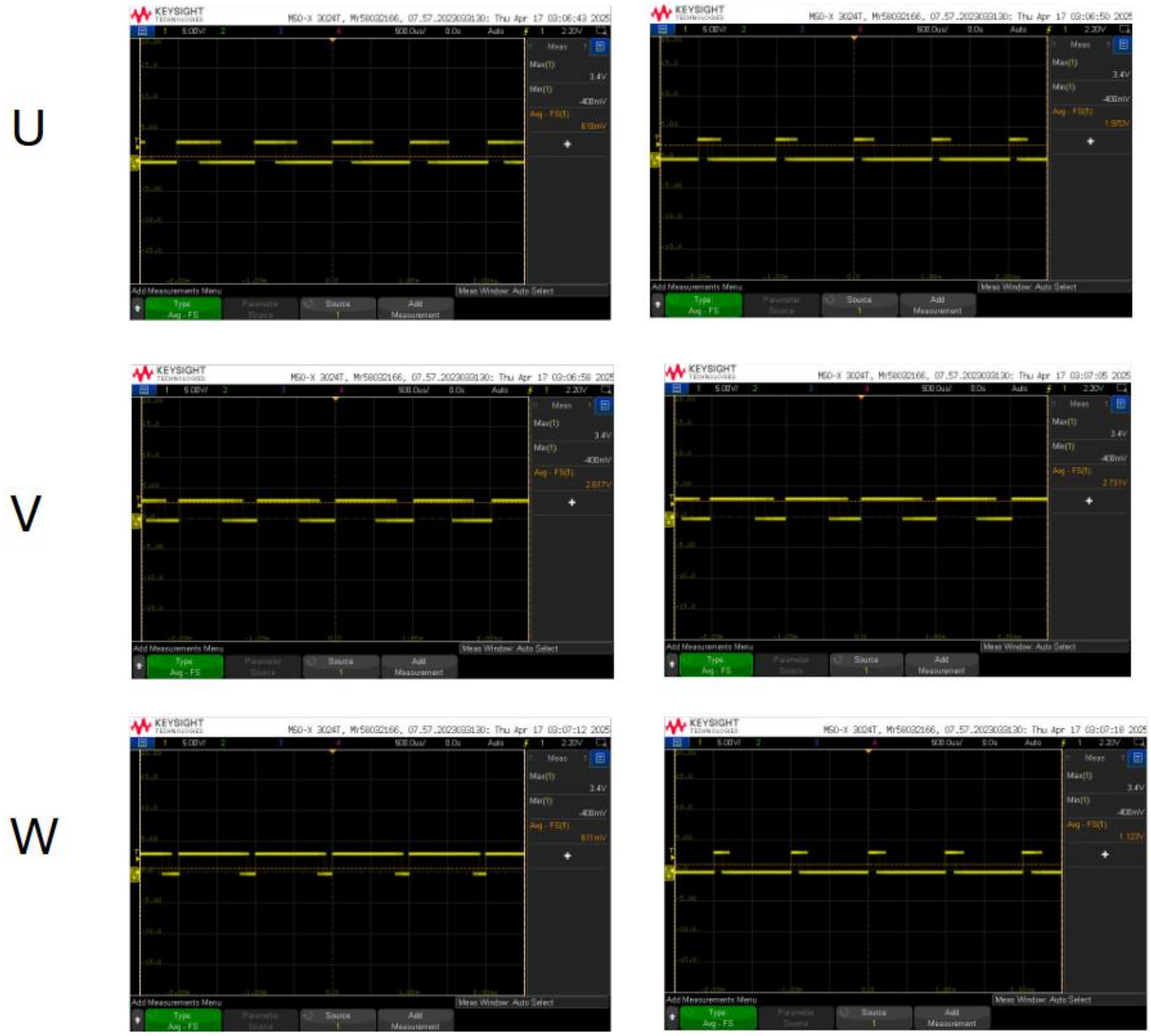

Figure 6: PWM input – Before Isolation

Figure 7: PWM Output – After Isolation

PWM Input Voltage	PWM Output Voltage
PWM 1H – 3.4V	PWM 1H – 6.2V
PWM 1L – 3.4V	PWM 1L – 6.2V
PWM 2H – 3.4V	PWM 2H – 6.0V
PWM 2L – 3.4V	PWM 2L – 6.2V
PWM 3H – 3.4V	PWM 3H – 6.2V
PWM 3L – 3.4V	PWM 3L – 6.2V

Table 1: PWM Input and Output Isolation Voltages

In the beginning of this project, this part of the subsystem was tested by connecting the PWM inputs to the development board and obtaining the 3.3V and 5V Vcc voltages for the isolators from a power supply. When full system testing was completed, these values were obtained by connecting the PWM input wires to the MCU, and the 3.3V and 5V Vcc voltages were obtained from the power and MCU subsystems via auxiliary power.

In addition to the correct values being output, the speed of these PWMs responded to the speed knob or potentiometer spinning. The speed knob functionality is essential to the controllability of the motor, and since the output of the PWM isolation worked with the speed knob, the PWM isolation was fully functional.

Another portion of the digital isolation was the relay control. On the optoelectronics board, this included a signal of 3.3V input when the motor was on and 0V when the motor was off. This on and off functionality was realized by a button on the MCU board. In Figures 8, 9, and 10, it is shown that when the button is pressed on the input voltage is 3.3V and output is 5V, and when the button is off, the output voltage was 0V.

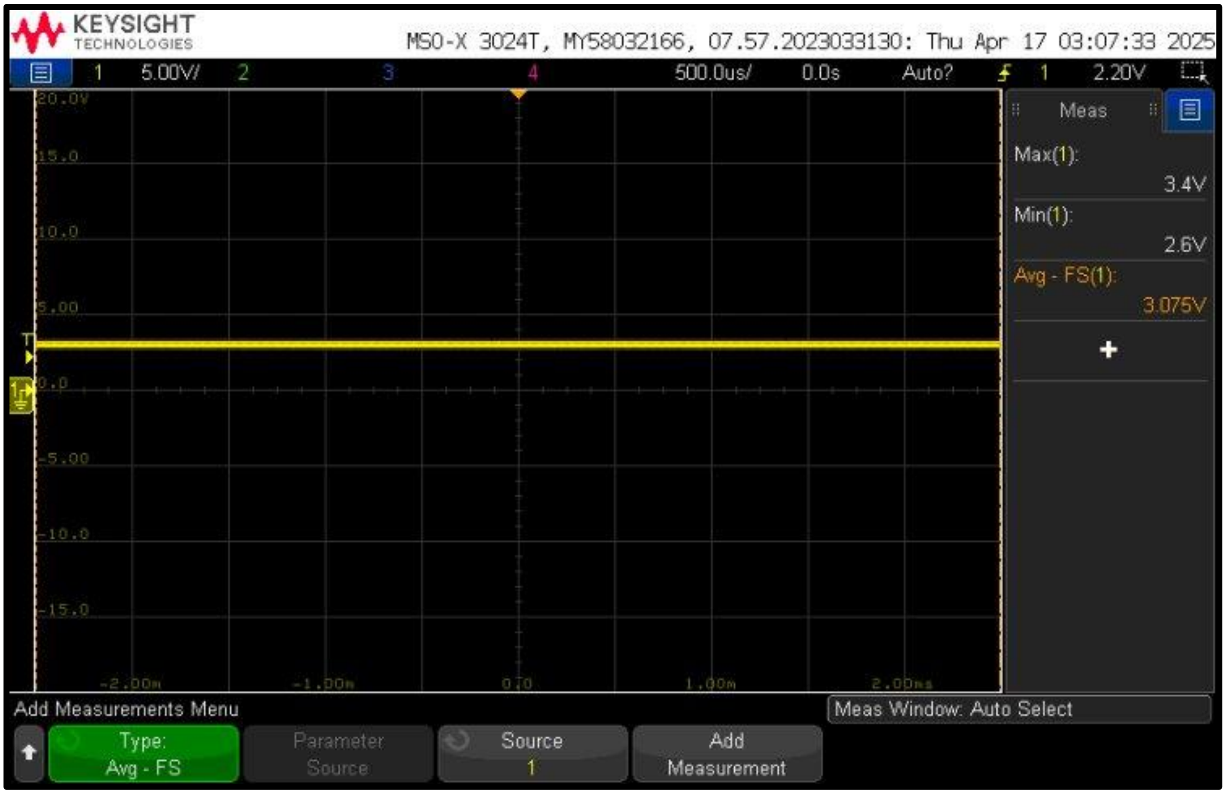

Figure 8: Relay Isolation Input Button On (3.4V)

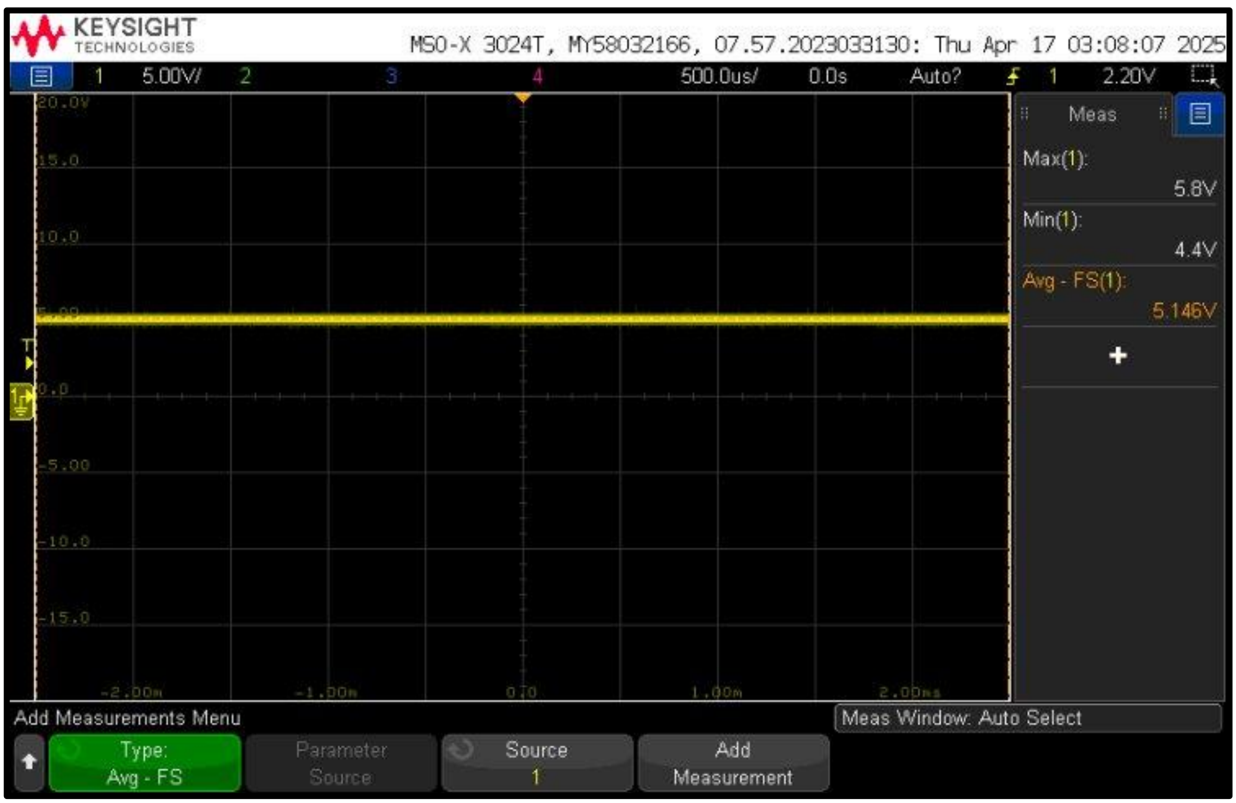

Figure 9: Relay Isolation Output Button On (5.8V)

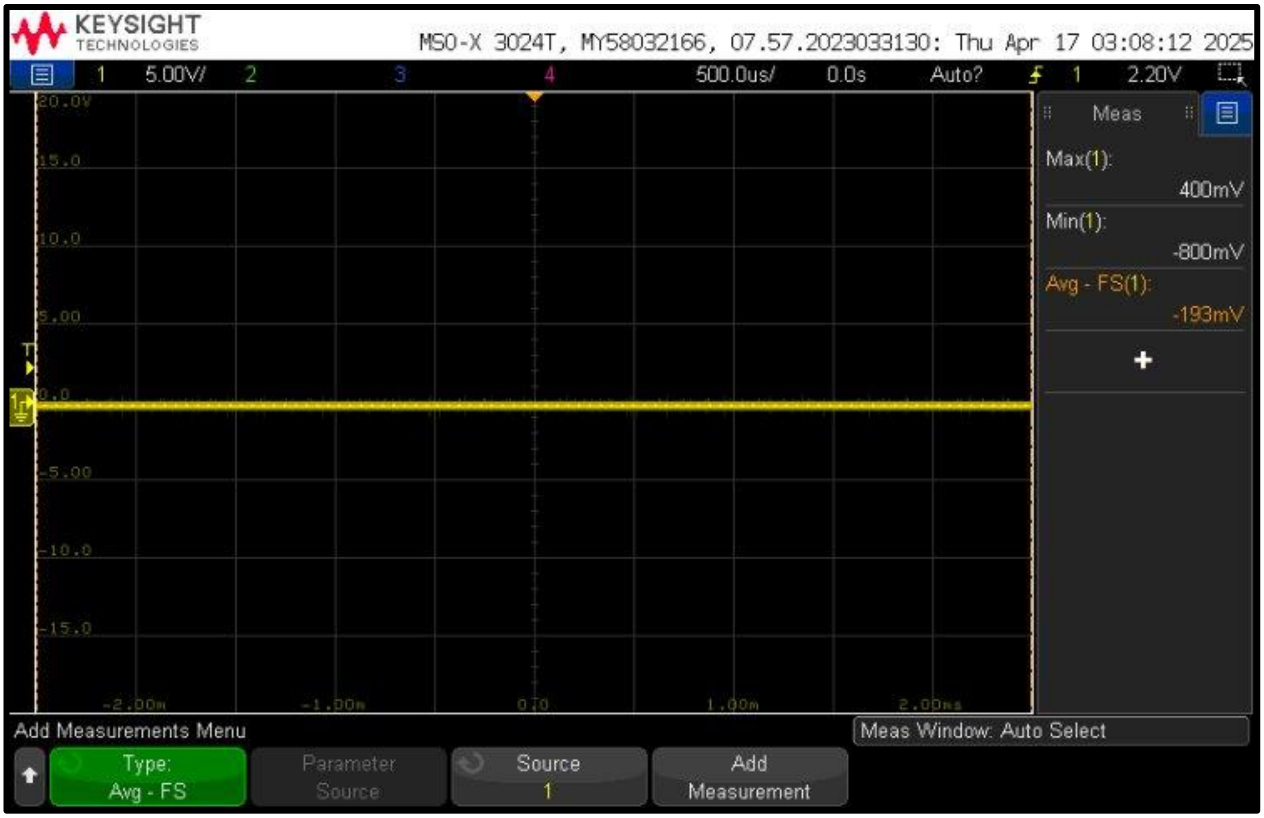

Figure 10: Relay Isolation Output Button Off (~0V)

An interesting mistake that was made in this portion of the optoelectronics board was the 3.3V Vcc for the isolators. The power to digital side requires 3.3V for the diodes, so 3.3V is on two connectors. Upon close inspection, the 3.3V Vcc was not actually routed to the 3.3V on the connector, but the circuit was working for a good amount of time when it appears that 3.3V was never connected, so it never should have been working. The isolators did end up stopping working which is what inspired the trace inspection. It was a very easy fix, just adding a thin blue wire, as seen in Figure 3 above, to connect 3.3V to the traces with the isolators. It is a mystery that has still not been solved, but now the circuit works because the error has been corrected.

2.3.2. Power to Digital Isolation Validation

This part of the subsystem proved to be much more difficult to test and validate than the PWM and relay isolation. This portion of the optoelectronics subsystem was never fully tested and validated. In the beginning of the project, the tests that were thought to be correct and proof of the circuitry working were not correct. The Vcc and GND as mentioned earlier were switched on all nine op-amps for this side of the board, so the op-amps never turned on which means the isolators did not get the correct voltages, so the whole test was inaccurate. This issue was first discovered when connecting auxiliary power, which was mentioned earlier.

The other subsystems were able to successfully create 15V from the 120VAC wall voltage, and when it was plugged into this board, the voltage dropped from 15V to around 4V. After cutting and replacing the incorrect traces to fix the problem, the voltage drop was gone. In Figure 11 below, the lack of voltage drop can be seen, as the DC power supply is set to supply 15V and it is.

Figure 11: 15V Supply Voltage Showing No Drop

Once this issue was solved, it was time to actually test. The requirements state that at full power, 120VAC, around 60V should be input from the power control and 15V should be output to the MCU. This requirement was for each of the three circuits: voltage, current, and temperature feedback. Because the 120VAC is a pretty dangerous voltage, it was advised to wait until the last moment to plug the system into the wall. To test this part of the subsystem, without having the full power feedback values, 15V input voltage was applied, and the values were as follows:

Feedback Signal	Input Voltage	Output Voltage
bus.v	15 V	3.9 V
ilim.v	15 V	3.9 V
ips.temp	15 V	3.9 V

Table 2: Diagnostic Feedback Input and Output Voltages

Based on this nomenclature, bus.v is the voltage that the motor is running at, ilim.v is the current, and ips.temp is the temperature. Based on the ratio, 15V:3.9V simplifies to 1V:0.26V. Had the full 60V been input, based on the ratio, ~15.6V would have been output. This voltage is within the 15-20V threshold in the requirement, so it is reasonable to assume this circuitry would work and be accurate at full power.

2.4. Subsystem Conclusion

Overall, this subsystem worked as expected, and it performed correctly when connected to the other subsystems to form the full system. The PWM isolation circuit works well and is an important part of spinning the motor. The PWMs are input from the MCU at 3.4V and output to the power control at 6.2V and 6.0V. This isolation helps the MCU to not come in contact with the high voltage of the motor and power electronics, and the increase in voltage allows the PWMs to be large enough to be taken in by the power control. The relay or on/off signal also works correctly, where before isolation there is 3.4V when the button is on before the isolators and 5.8V after. When the button is off, the relay signal is 0V before and after isolation. The power to digital isolation which is used for the diagnostic feedback also works as expected. This part was never fully tested due to timing, however. The motor spinning was the main focus of the team, so the feedback section was second priority. The optoelectronics section of feedback worked, but there were other problems with integration that could have been solved had there been a few more weeks. The validation of this part of the optoelectronics subsystem seemed to be correct, so in theory, it should have worked correctly at full power.

3. Microcontroller Subsystem Report

3.1. Subsystem Introduction

This subsystem refers to the hardware of the microcontroller portion of the VFD motor controller. In order for the firmware to control the motor, the microcontroller must be implemented properly. This subsystem is powered by two of the three phase AC power supply which is sent to the MCU for functionality.

3.2. Subsystem Details

The DSPIC33CK256MP508 microcontroller was selected for this project. It has an adequate amount of analog and digital pins to support the VFD motor and the required PWM signals necessary. This MCU has 80 pins however only 32 are needed for this application. The MCU is supplied by the 15 VDC from the AC power supply fed by the main power. The voltage will then be stepped down to a usable 3.3V using a 3.3V fixed buck converter. The buck converter takes the 15V from the AC/DC converter to a usable voltage for the microcontroller. This 3.3V is needed for the microcontroller along with the Pickit4 debugger and the UART serial interfaces.

The MCU subsystem receives feedback through low-voltage analog signals representing voltage and current. The MCU then sends out PWM signals to the H bridge and power control system which are used to control the inverter stage of the VFD which helps to adjust the output voltage and frequency supplied to the motor.

The hardware necessary for the application is placed onto a PCB and the firmware will be coded onto the MCU to control the PWM signals. A potentiometer is included to vary the frequency of these PWM signals. A push button will also be used to turn the motor on and off which is indicated by a green LED. A 5 pin connector is used to connect to the Pickit5 debugger which is used to flash the code onto the MCU. A 3 pin connector is used for UART serial communication using a cable to connect to a PC. Finally, a 20 pin connector is used to send the PWM signals back to the H bridge and to connect to the GPIO pins. This PCB will later be connected to the power and optoelectronics subsystems.

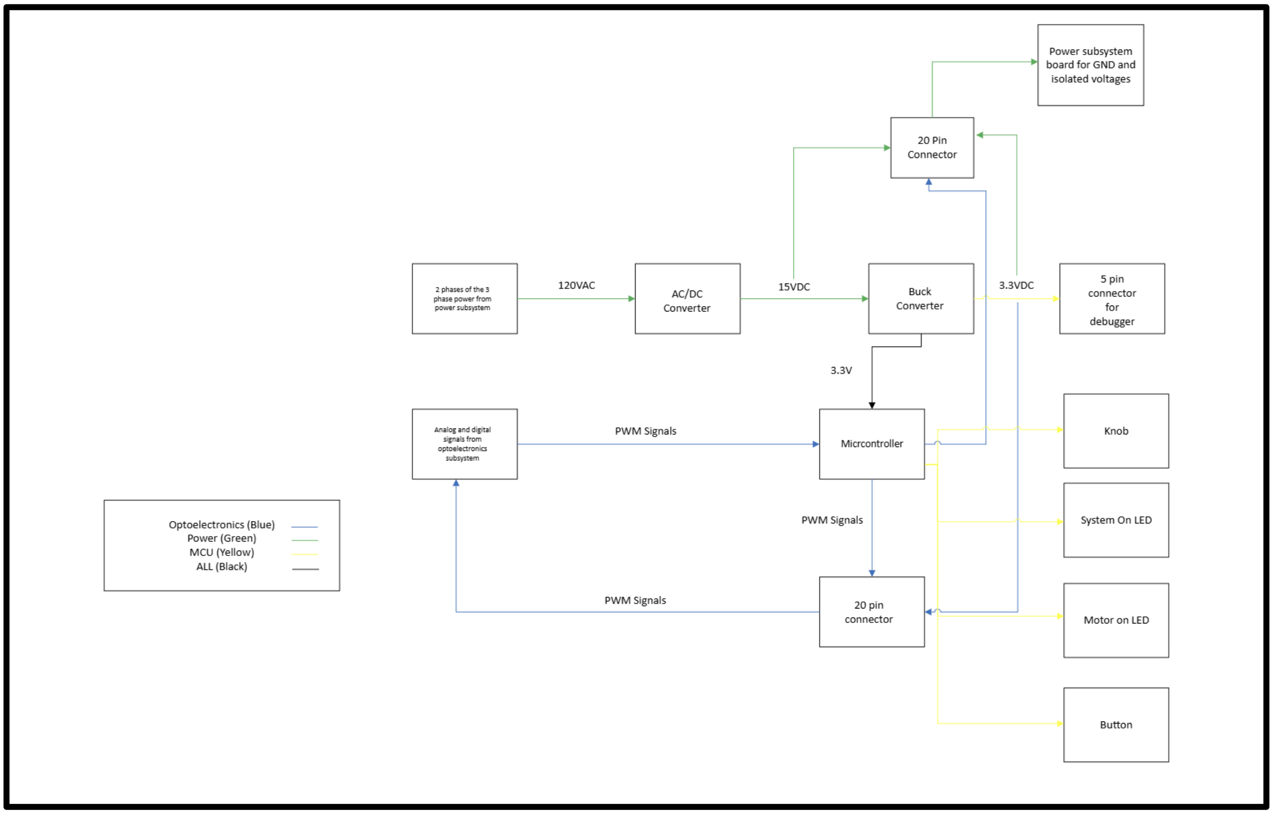

Figure 12: MCU Block Diagram

Pin	Function
1	PWM1H
2	GPIO
3	PWM1L
4	GPIO
9	MCLR
11,32,50,70	VSS
12,31,51,71	VDD
17,22,39,44	GPIO
19	GPIO (Potentiometer)
24	GPIO (LED 2)
25	AVDD
26	AVss
34	Clock input
37	GPIO (LED 1)
42	GPIO (Relay)

60	PGD1
61	PGC1
68	UART TX
69	UART RX
75	PWM3H
76	PWM3L
78	PWM2H
80	PWM2L

Table 3: Pin interface for DSPIC33CK256MP508 Microcontroller

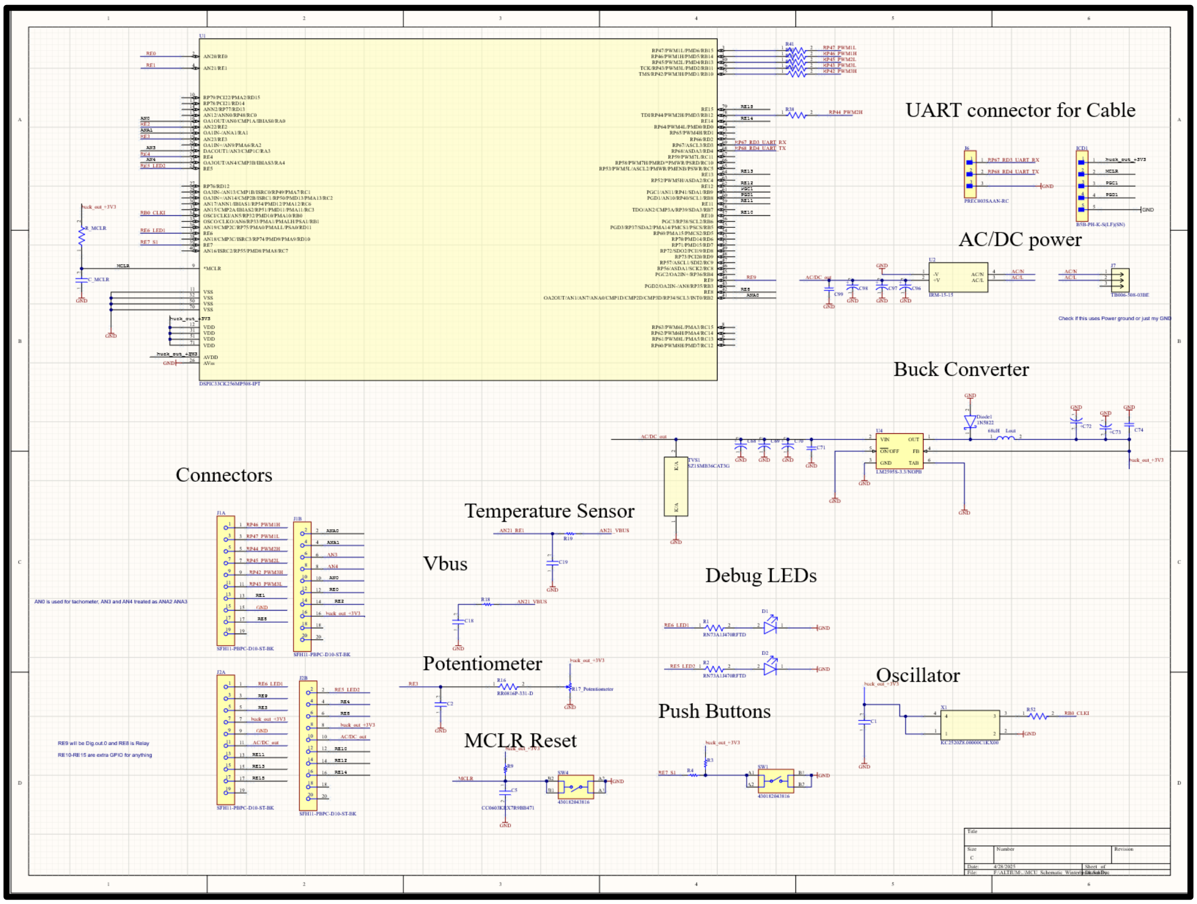

Figure 13: MCU subsystem schematic

3.3. Subsystem Validation

This subsystem refers to the hardware portion and PCB design for the microcontroller. This includes the 15V AC/DC converter, 3.3V buck converter, MCU, potentiometer, system and motor LEDs, and the on/off button. The IRM-15-15 AC/DC converter will take in the 120VAC wall power from the power subsystem board and convert it to 15VDC. This 15VDC will then be stepped down using the LM2595s-3.3 to convert it to 3.3V in order for the MCU to function.

The MCU will then be coded by the firmware to implement the system and motor LEDs, the potentiometer (knob), the on/off button, and to send out the PWM signals to the optoelectronics board.

3.3.1. Converter Validation

The first portion of the subsystem that was tested and validated were the power converters. The 3.3V buck converter was the first converter to be validated. Last semester the schematic and corresponding pcb routing was done incorrectly. For the final revision of the buck converter, different through hole capacitors were added along with a different diode. This converter was tested by injecting a variety of voltages using the DC power supply along with load resistors for the line and load regulation tests. The AC/DC converter was then tested after ensuring that the buck converter functioned properly. This was only tested by using wall power (~120VAC) and plugging it into the board and measuring the output on one of the 20-pin connectors used for supplying the Power subsystem board.

Load (mA)	Vout (V)
0	3.2993
30	3.2952
50	3.2824
72	3.2812
95	3.2601
122	3.2512
153	3.2243

Table 4: Load regulation validation for buck converter (LM2595-3.3s)

V_{in} (V)	V_{out} (V)
16.0	3.003
15.5	3.2997
15.0	3.2997
14.5	3.2995

Table 5: Line regulation validation for buck converter (LM2595-3.3s)

Vin (VAC)	Vout (VDC)
118	15.0

Table 6: Validation for 15V AC/DC converter (IRM-15-15)

3.3.2. Button and Relay Signal Validation

This portion of validation was done after the board was fully assembled. The button and relay signal was unable to be tested the first semester because the buck converter circuit was made incorrectly, causing the MCU to not have the proper power. After ensuring the MCU received the proper 3.3V from the buck converter, the code was flashed using a Pickit5 and MPLAB using the 5-pin connector on the board. After this was done, the button was given functionality over the relay signal. The first test done was to ensure the button functioned properly and could turn on and off. The second test ensured the relay was funcitoning properly and was connected to the button.

Button	
Not Pressed	Pressed
3.25V	-.048V

Table 7: Button signal voltage when button is toggled

Relay	
On	Off
3.281V	-.045V

*Table 8**:** Relay signal voltage when toggled*

3.3.3. LED Validation

After ensuring the button and relay works the LEDs were validated. The original LEDs failed to function properly due to improper soldering, however the component was small and there was a risk of burning a pad when trying to get them off. The addition of two LEDs -- one red and one blue – was done as a compromise. These LEDs were placed in the vias for the original LEDs and blue wired to a GND via. They were positioned to avoid a possible short. The first LED to be tested was the power on LED (red). When the components receive the 3.3V from the buck converter the red LED shall light up as shown below. The blue LED turns on when the button is pressed indicating that the motor is on.

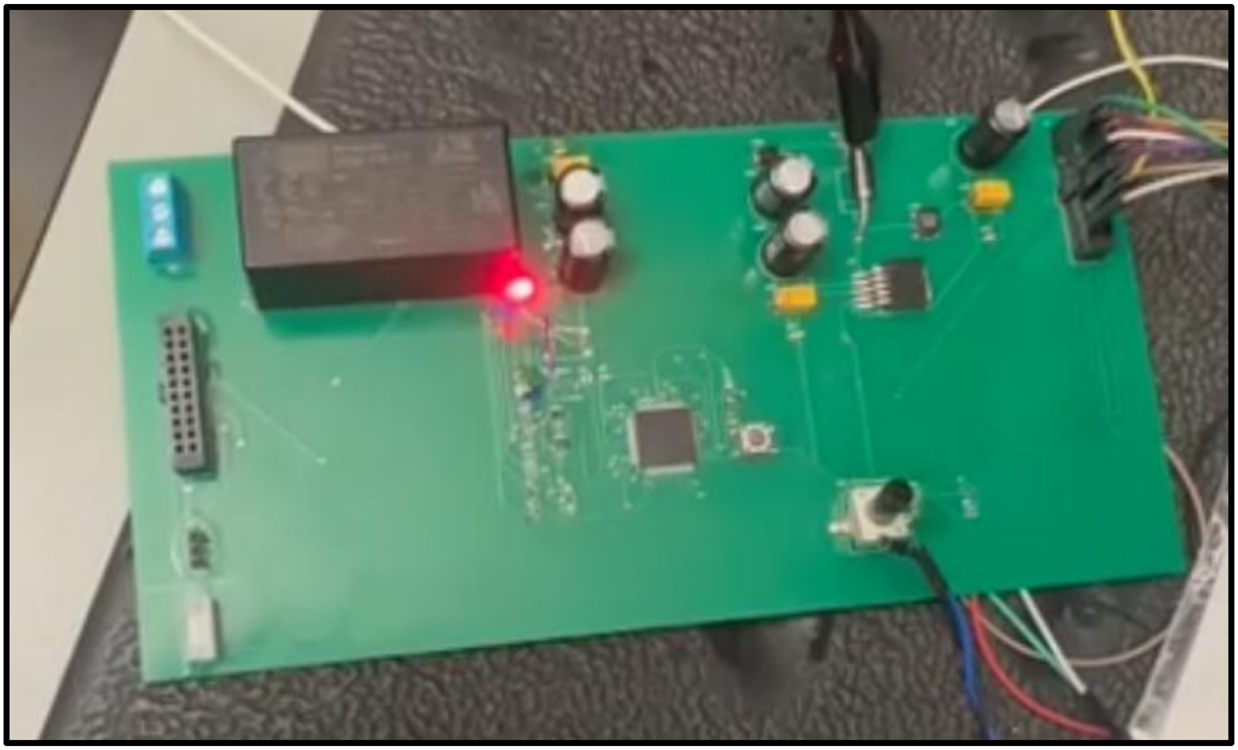

Figure 14: Power On LED (red)

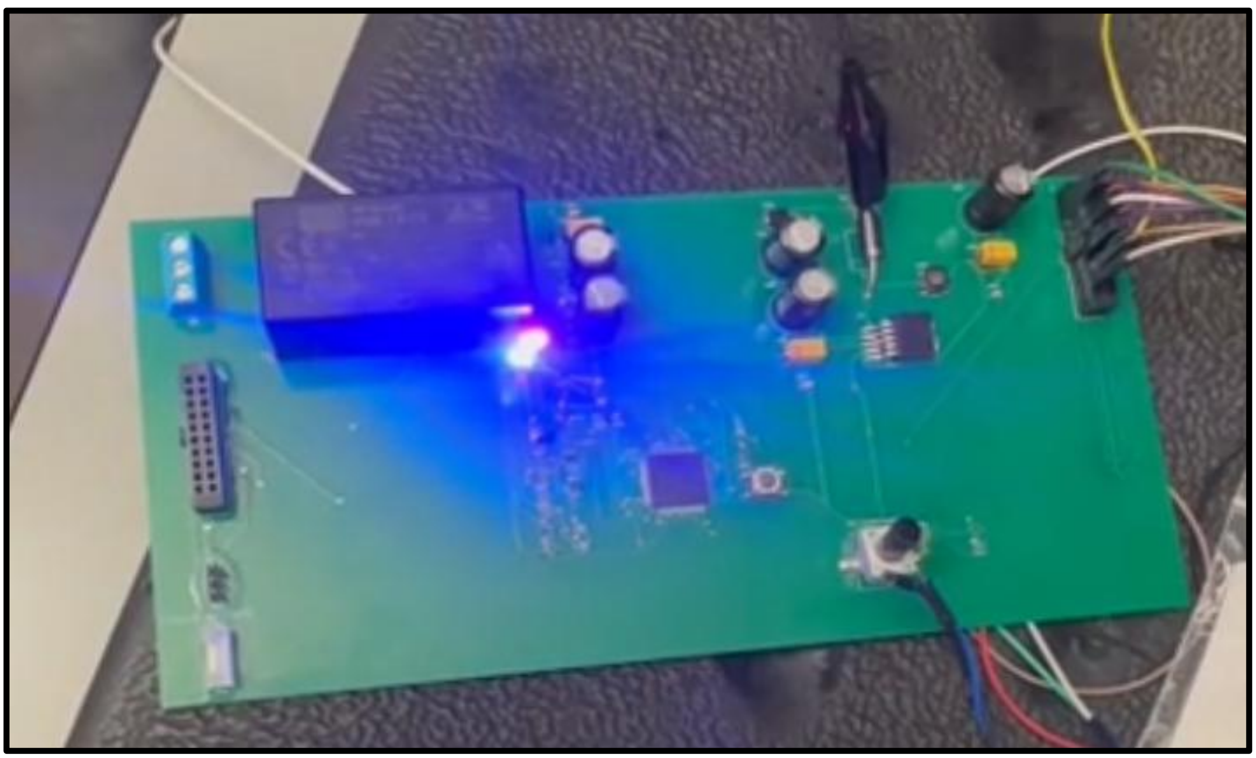

Figure 15: Motor On LED (blue)

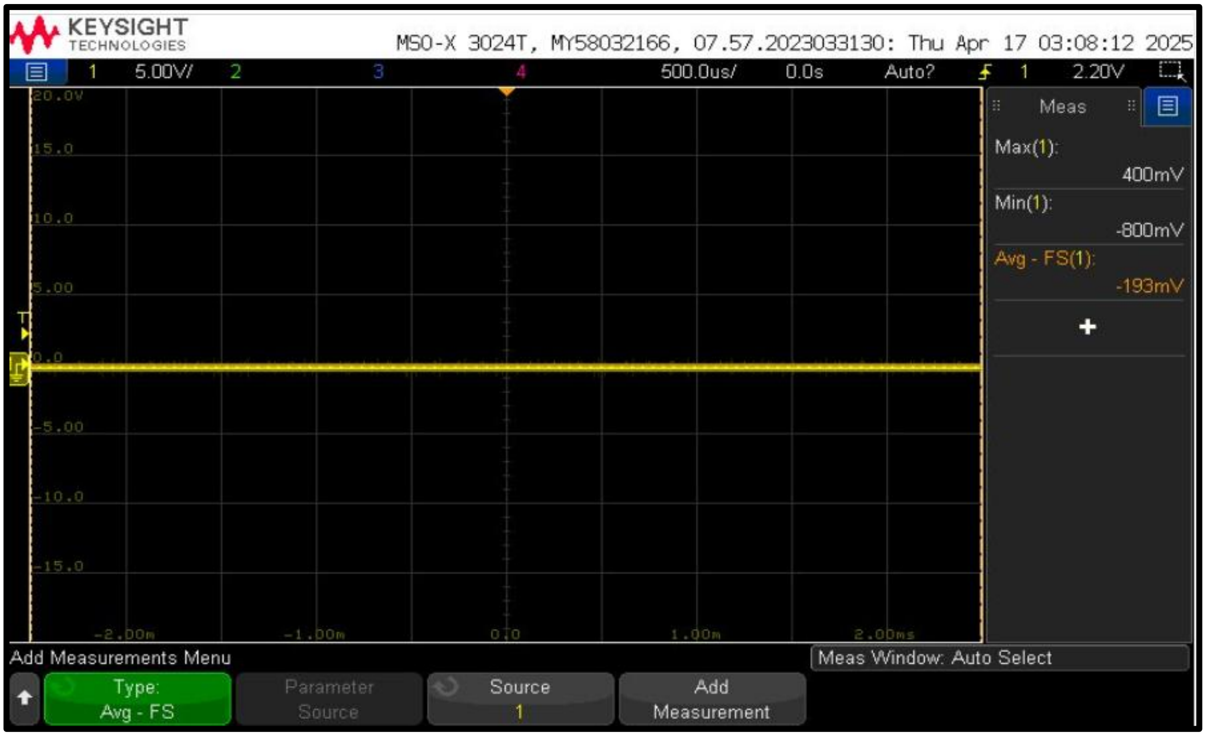

Figure 16: MCU toggling motor on/off LED voltage (Off)

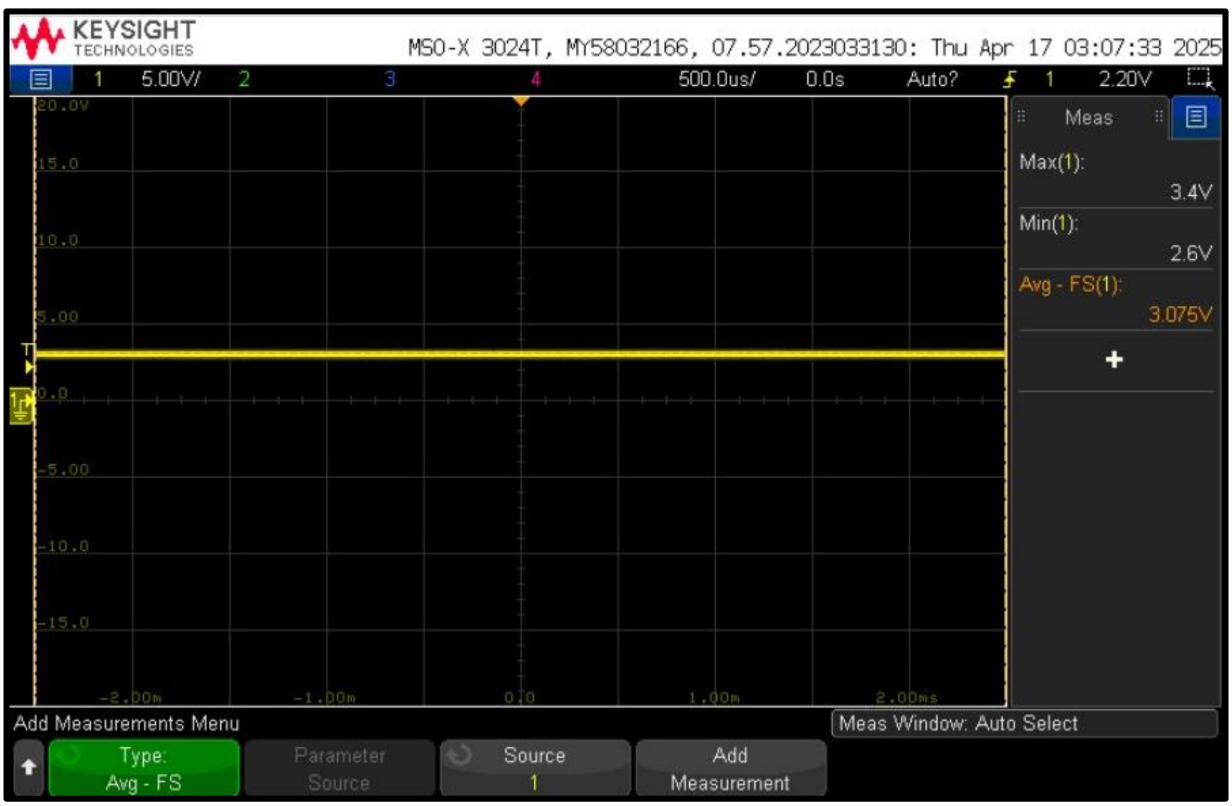

Figure 17: MCU toggling motor on/off LED voltage (On)

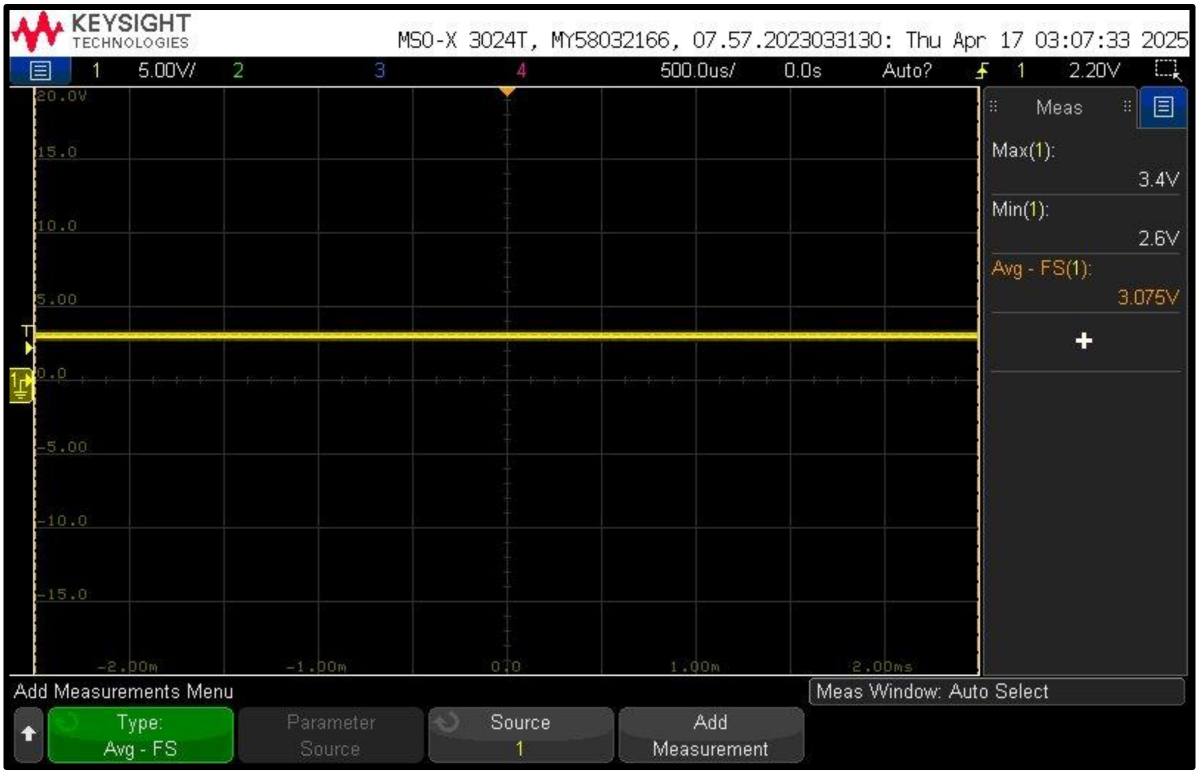

Figure 18: MCU toggling system on/off LED voltage (On)

Figure 19: MCU toggling motor on/off LED voltage (Off)

3.3.4. Knob and PWM Validation

In order for the motor to function properly, the correct PWM signals need to be produced by the MCU. Three high and three low signals need to be produced, and the knob must be able to change the frequency of these signals. The second revision of the PCB did not have the knob (potentiometer) properly routed. Pin 19 on the MCU was then routed to the potentiometer for it to be coded and function properly.

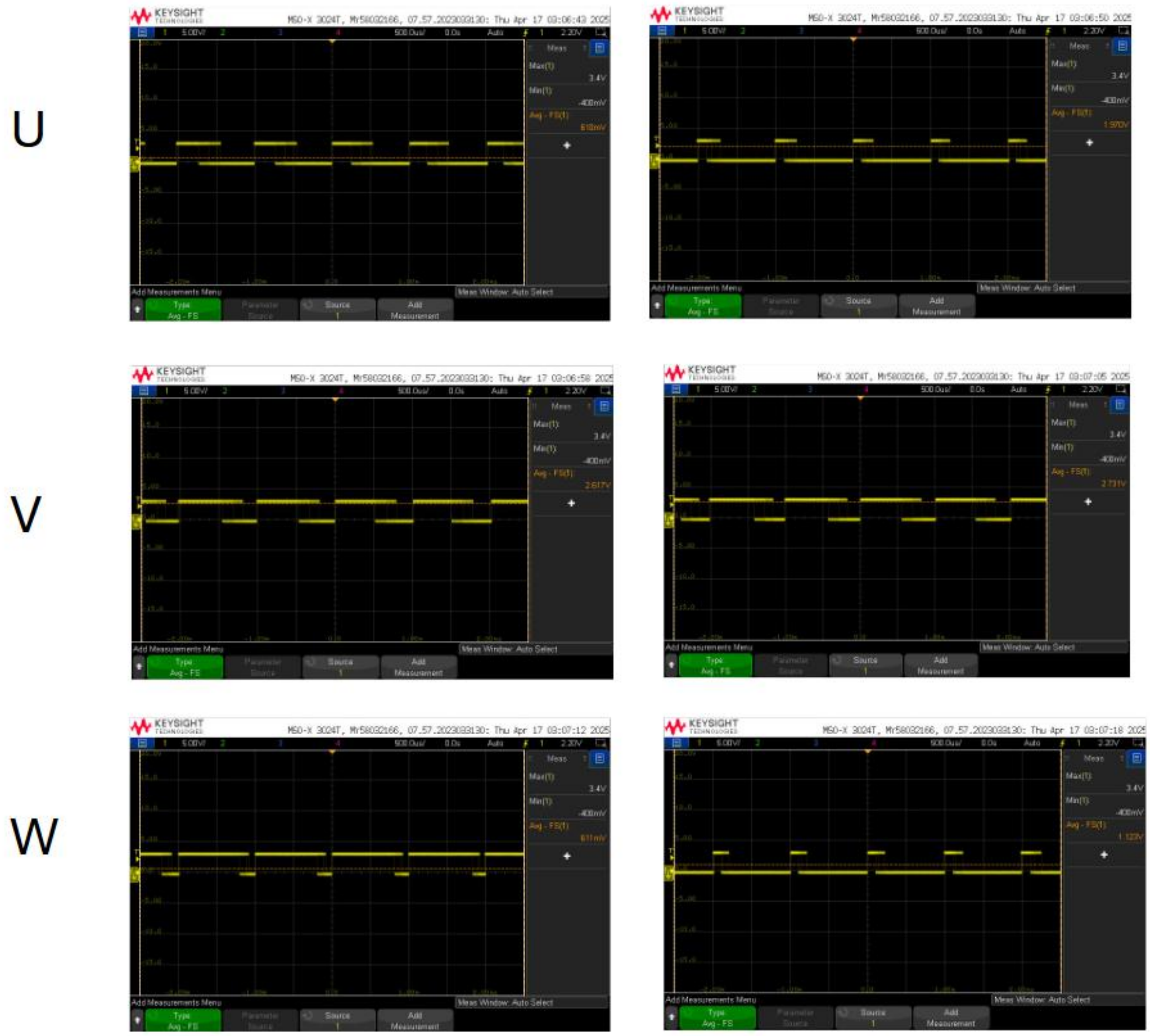

Figure 20: Three high and three low PWMs with knob signal

Knob	
Positions	Voltage
Low	-.045V
Middle	1.603V
High	3.285

Figure 21: Knob position and corresponding voltage level

The lower voltage levels indicates that the motor will be spinning at a lower speed. As the knob is turned counterclockwise, the voltage reaches its maximum (~3.3V) indicating it's at its maximum speed.

Signal	**Frequency (kHz)**	**Minimum Voltage**	**Maximum Voltage**	**Phase Offset (degrees)**	**Low Knob Value - Low Frequency**	**High Knob Value - High Frequency**
1H	1	-0.13V	3.65V	0	TRUE	TRUE
1L	1	-0.13V	3.65V	0	TRUE	TRUE
2H	1	-0.13V	3.65V	120	TRUE	TRUE
2L	1	-0.13V	3.65V	120	TRUE	TRUE
3H	1	-0.13V	3.65V	240	TRUE	TRUE
3L	1	-0.13V	3.65V	240	TRUE	TRUE

Figure 22: Knob effect on PWM signals

3.4. Subsystem Conclusion

Overall this subsystem functioned properly up until the end. The 15V AC/DC converter brick stopped working after a few full system tests at the end of the semester. The diagnostic feedback was also unable to be coded using the TTL-232R-RPi debug cable and MPLAB. The rest of the functionalities of the MCU worked without fail. After switching from the faulty Pickit4 to the newer Pickit5, the code was easily able to be flashed onto the MCU. It was able to output the proper PWMs for the motor to run and was also able to control the speed using the knob. The start/stop button worked as well and the blue LED was turned on when the button was pressed on. Although the LEDs caused a few hiccups along the way, a compromise was made and the through-hole LEDs functioned as needed. The red system power LED lit up when the board received the proper amount of voltage. With the 15V being injected to the LM2595-3.3s buck converter to bypass the broken 15V AC/DC converter, the full system was able to obtain the needed 3.3V and 15V from this subsystem's PCB board. The MCU received the proper voltage and a safe amperage level; the power board was able to receive the common GND, and the 15V and 3.3V needed for isolation, and the optoelectronics board was able to receive the PWMs, 3.3V, and common GND from the board. With this shortcut, the full system was able to function properly and the validation indicates the system should work under full power.

4. Power Subsystem Report

4.1. Subsystem Introduction

The power subsystem of the VFD motor controller is responsible for supplying power to both an AC induction motor and the entire system. It is designed to operate with 120 V_{AC} three-phase power and has been tested to ensure safe and reliable performance. The following sections present an analysis of the subsystem's operation and validation.

4.2. Subsystem Validation

4.2.1. Rectifier and DC Link Validation

A rectifier converts AC to DC, while a DC link filters the output. In Figure 23, the rectifier schematic is shown. First, a power input connector receives the desired three-phase power. Next, a set of capacitors in parallel with varistors is connected across each phase. The capacitors filter the voltage to reduce noise, and the varistors provide protection against voltage surges. Finally, a full-wave rectifier converts the AC to DC.

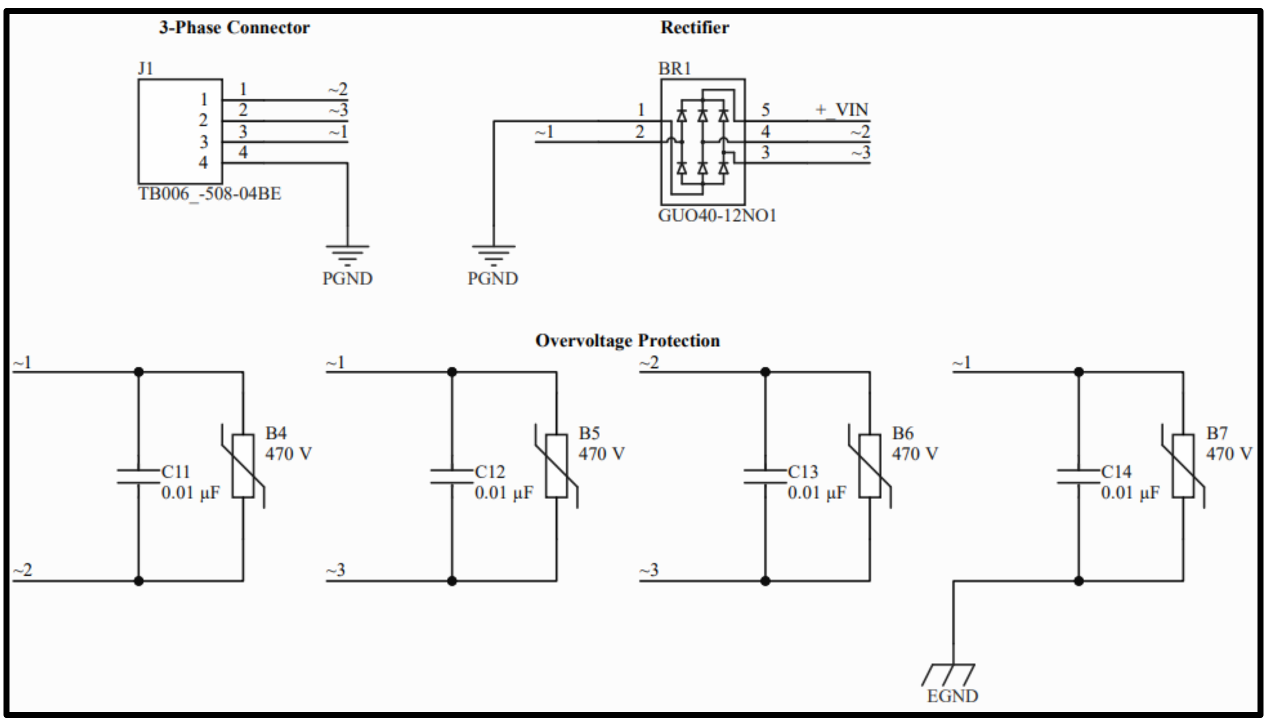

Figure 23: Rectifier Schematic

In Figure 24, the DC link schematic is shown. First, a fuse provides protection against overcurrent. Next, a varistor in parallel with a set of capacitors is connected to ground. The varistor provides protection against voltage surges, and the capacitors filter the voltage to reduce noise.

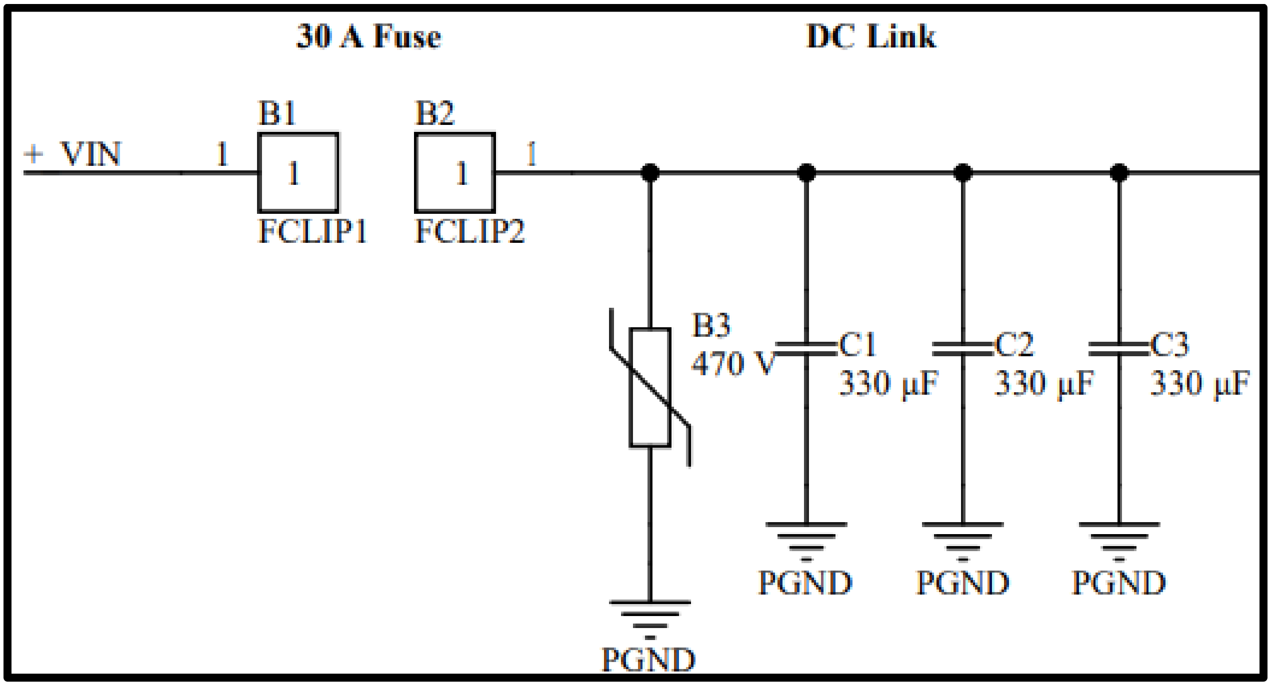

Figure 24: DC Link Schematic

The rectifier and DC link were validated by supplying an input AC voltage and measuring the output DC voltage, as shown in Figure 25. While the integrated system is designed to operate at 120 V_{AC}, the validation testing was conducted at 10 V_{AC}. The three-phase AC was generated using three waveform generators, with phase A and phase B values in Figure 26 and phase C values in Figure 27. An amplitude of 3.536 V_{rms} was used to produce the desired voltage. Additionally, each phase was offset by 120˚ to simulate a three-phase system, with phase C further adjusted due to the waveform generators not being synchronized.

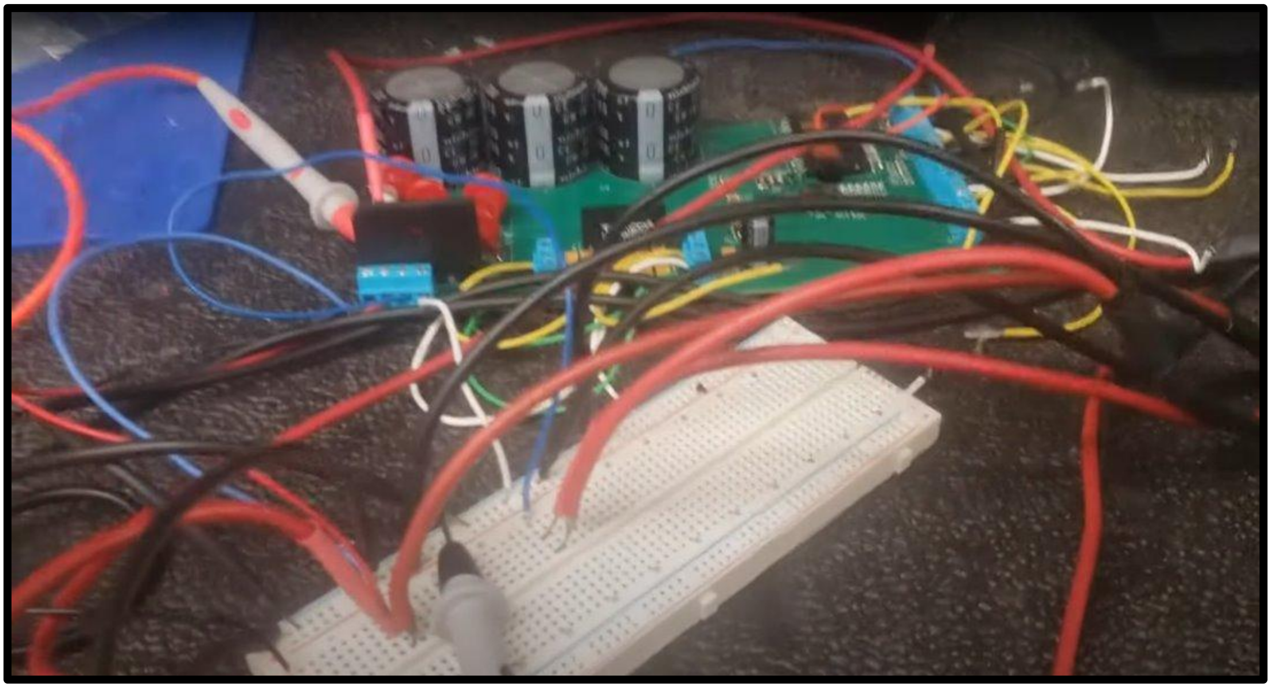

Figure 25: Rectifier and DC Link Circuit

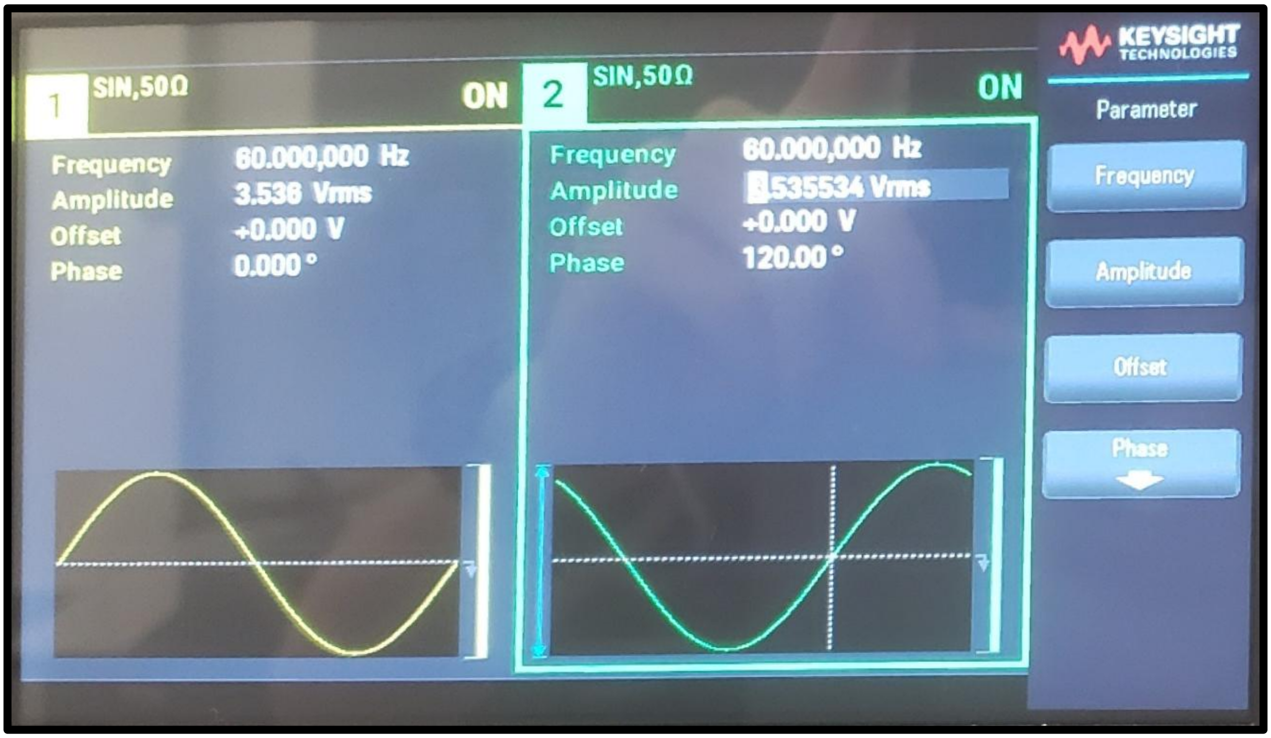

Figure 26: Input Phase A and Phase B

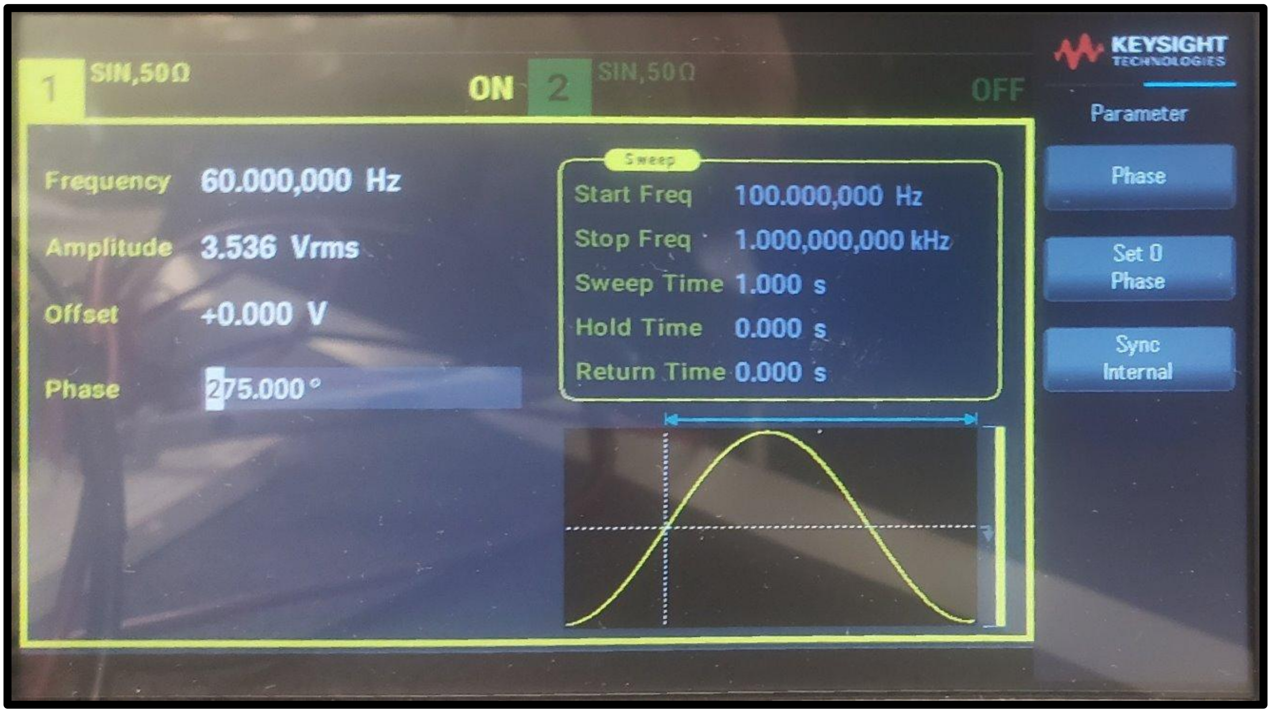

Figure 27: Input Phase C

Upon measuring the input AC waveforms shown in Figure 28, the green (phase A), blue (phase B), and purple (phase C) sinusoidal waves demonstrate a phase shift of 120 degrees and an amplitude of 10 V_{AC}. Furthermore, the measured DC output is 9.6 V_{DC} with 600 mV of noise. This differs slightly from the expected output of 12 V_{DC} .This discrepancy can be attributed to the forward voltage drop across the rectifier diodes and losses within the rectifier and DC link circuitry. Despite these factors, the measured value remains within an acceptable range for the system's operation.

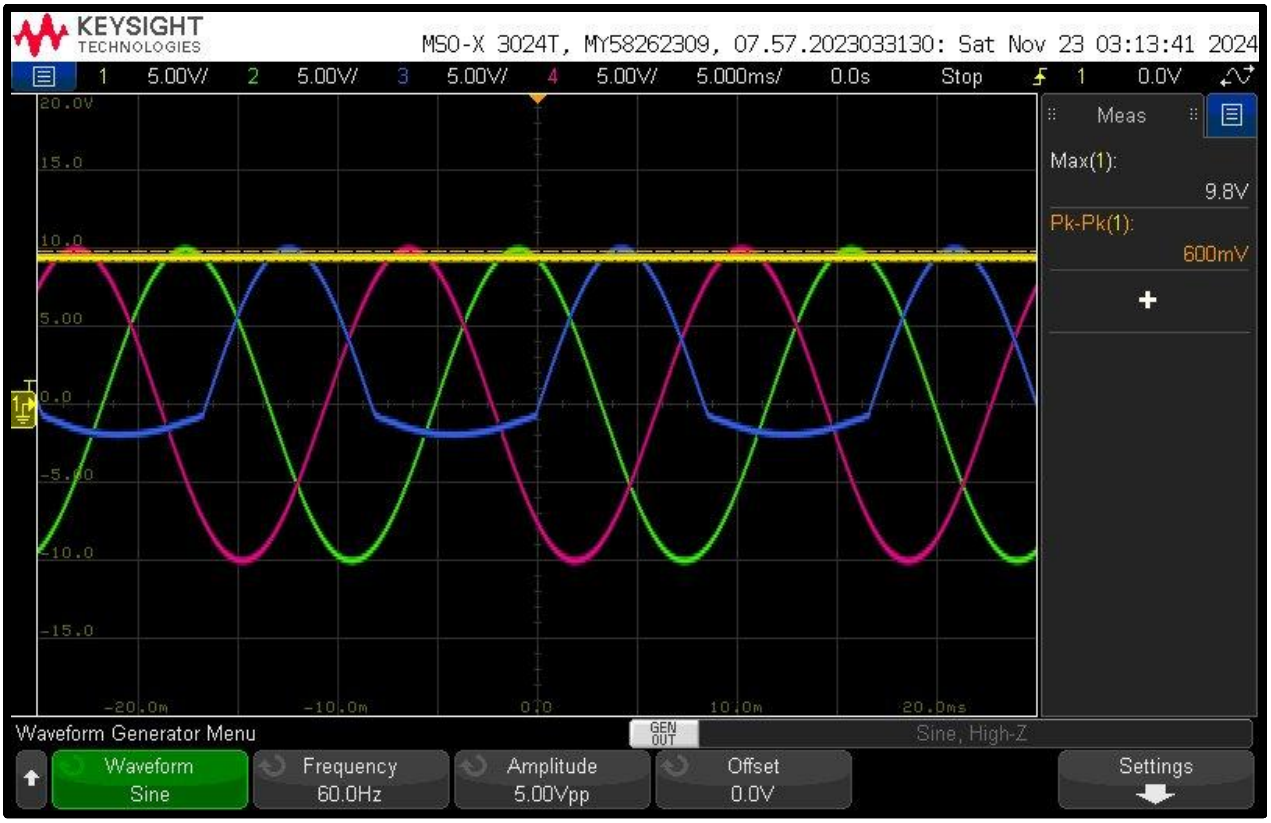

Figure 28: Input Phase A, Phase B, Phase C; Output DC

4.2.2. Power Control Validation

The power control takes an input of DC voltage from the DC Link and based on the input PWM signals from the microcontroller, inverts the DC voltage into the appropriate AC voltage to power the motor. The power control was validated by applying an input DC voltage and PWM signals and measuring the output waveforms. In Figure 29, the measured output waveforms are shown to be the expected modified sine waves at 120 degree phase shift. Furthermore, the input dc voltage was generated using a DC power supply and the PWM signals were generated using the integrated system.

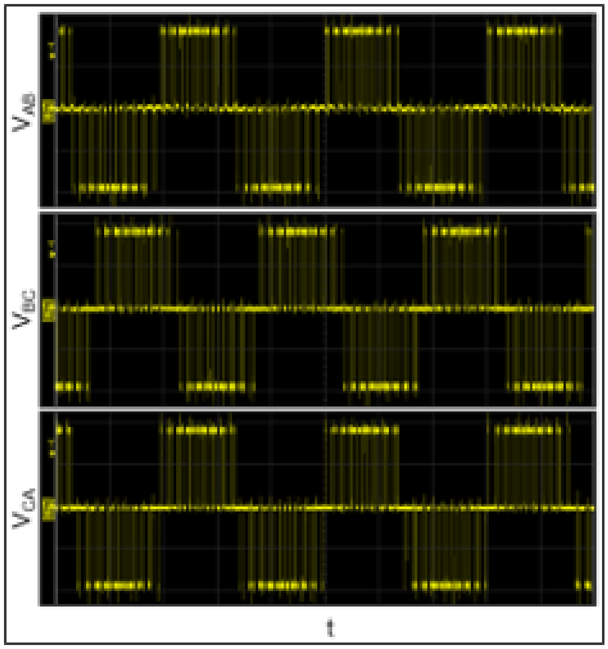

Figure 29: Power Control Output

4.2.3. 3.3 V_{DC} to Isolated 5 V_{DC} Converter Validation

The 3.3 V_{DC} to isolated 5 V_{DC} converter converts 3.3 V_{DC} to isolated 5 V_{DC}. In Figure 30, the 3.3V/ISO_5V converter schematic is shown. First, an input connector receives the input 3.3 V_{DC}. Next, a set of capacitors in parallel is connected to ground, to filter the voltage and reduce noise. Then, a power converter steps up and isolates the 3.3 V_{DC} to 5 V_{DC}. Finally, another set of capacitors in parallel is connected to ground, to filter the voltage and reduce noise.

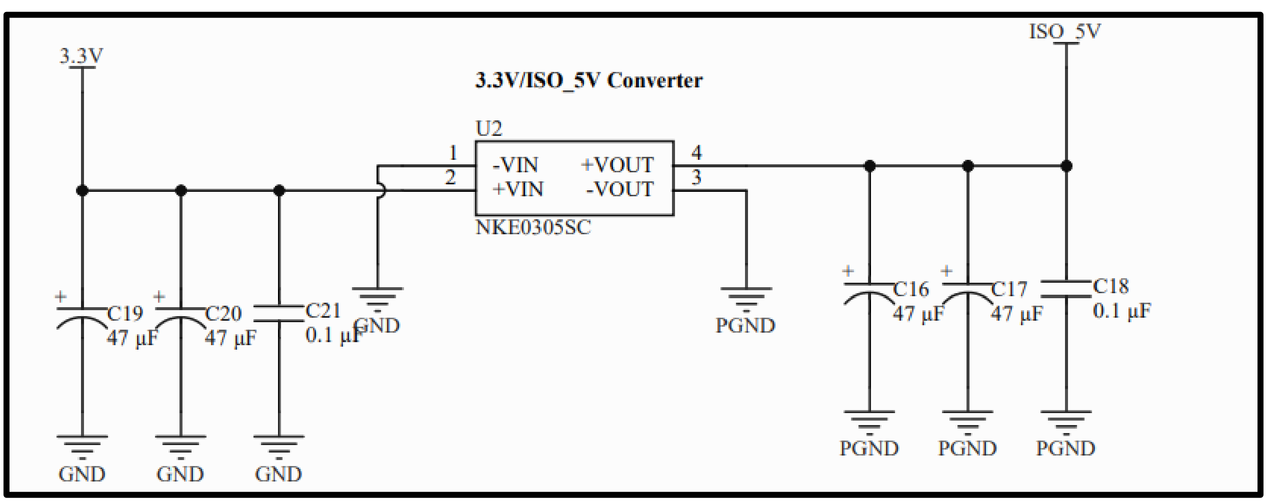

Figure 30: 3.3V/ISO_5V Converter Schematic

The 3.3 V_{DC} to isolated 5 V_{DC} converter was validated by supplying an input voltage and measuring the output voltage under varying loads. The validation testing was conducted at an input voltage of 3.3 V_{DC} using a DC power supply and the varying currents using an electronic load. The resulting data was recorded in Table 9.

I_{out} (mA)	V_{out} (V)
12	5.381
50	5.188
100	4.981
150	4.782
200	4.581

Table 9: 3.3V/ISO_5V Converter Load Regulation Test

Furthermore, the 3.3 V_{DC} to isolated 5 V_{DC} converter was validated by supplying various input voltages and measuring the output voltage. The validation testing was conducted at a range of values similar to the operating voltage of 3.3 V_{DC} and was generated using a DC power supply. The resulting output voltages were recorded in Table #.

V_{in} (V)	$V_{in,pp}$ (mV)	V_{out} (V)	$V_{out,pp}$ (mV)
2.8	500	4.9	500
3.2	500	4.8	500
3.3	500	5.3	500
3.6	900	5.7	500

Table 10: 3.3V/ISO_5V Converter Line Regulation Test

Upon measuring the output voltages in Table 9 and Table 10, the converter behaved as expected for its full range of voltage and current values.

4.2.4. 15 V_{DC} to Isolated 15 V_{DC} Converter Validation

The 15 V_{DC} to isolated 15 V_{DC} converter converts 15 V_{DC} to isolated 15 V_{DC}. In Figure 31, the 15V/ISO_15V converter schematic is shown. First, an input connector receives the input 15 V_{DC}. Next, a set of capacitors in parallel is connected to ground, to filter the voltage and reduce noise. Then, a power converter isolates the 15 V_{DC} to 15 V_{DC}. Finally, another set of capacitors in parallel is connected to ground, to filter the voltage and reduce noise.

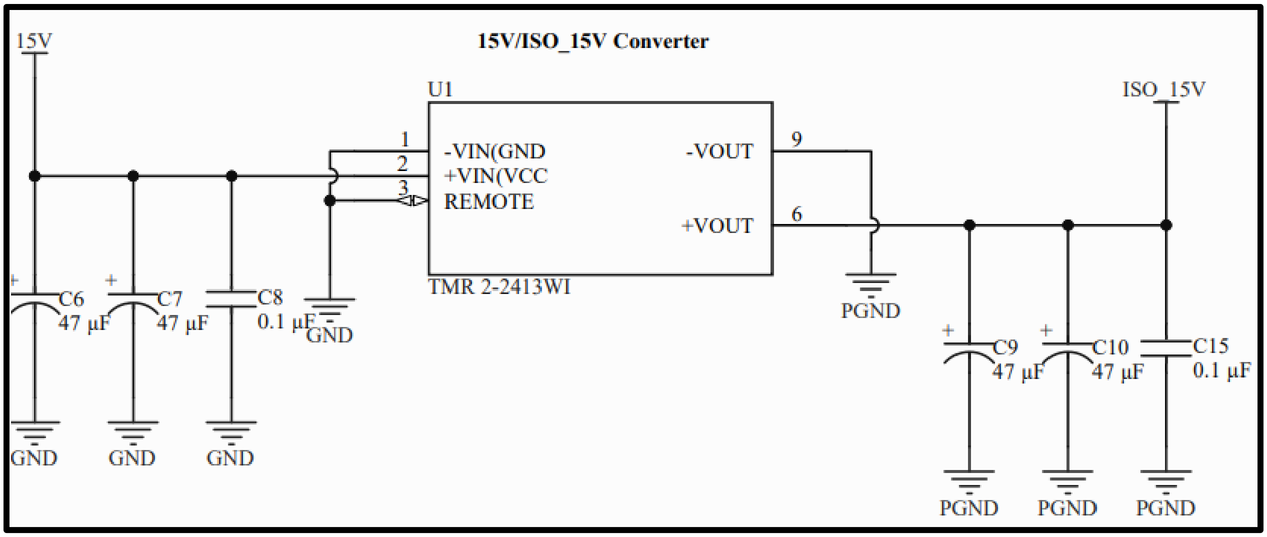

Figure 31: 15V/ISO_15V Converter Schematic

The 15 V_{DC} to isolated 15 V_{DC} converter was validated by supplying an input voltage and measuring the output voltage under varying loads. The validation testing was conducted at an input voltage of 15 V_{DC} using a DC power supply and the varying currents using an electronic load. The resulting data was recorded in Table 11.

I_{out} (mA)	V_{out} (V)
12	14.959
25	14.951
50	14.937
75	14.924
100	14.912
125	14.898
134	14.893

Table 11: 15V/ISO_15V Converter Load Regulation Test

Furthermore, the 15 V_{DC} to isolated 15 V_{DC} converter was validated by supplying various input voltages and measuring the output voltage. The validation testing was conducted at a range of values according to the datasheet and was generated using a DC power supply. The resulting output voltages were recorded in Table 12.

V_{in} (V)	$V_{in,pp}$ (mV)	V_{out} (V)	$V_{out,pp}$ (mV)
9.1	200	15.1	400
15.1	500	15.0	400
20.9	500	14.9	400
27.0	500	15.3	500
33.0	578	14.9	264
35.8	503	14.9	264

Table 12: 15V/ISO_15V Converter Line Regulation Test

Upon measuring the output voltages in Table 11 and Table 12, the converter behaved as expected for its full range of voltage and current values.

4.2.5. Relay Validation

The relay takes in the relay signal and toggles the motor power between on and off using a switch. In Figure 32, the schematic is shown. In the left contact position, the DC voltage is input using a DC power supply and in the right contact position the DC voltage will be output if the switch is closed. Next, the coil voltage is set to 15 VDC, but the circuit is incomplete because of the MOSFET position relative to PGND. So, the relay signal of 5 VDC is generated using a DC power supply to trigger the MOSFET, completing the coil voltage to PGND, and the switching the relay closed.

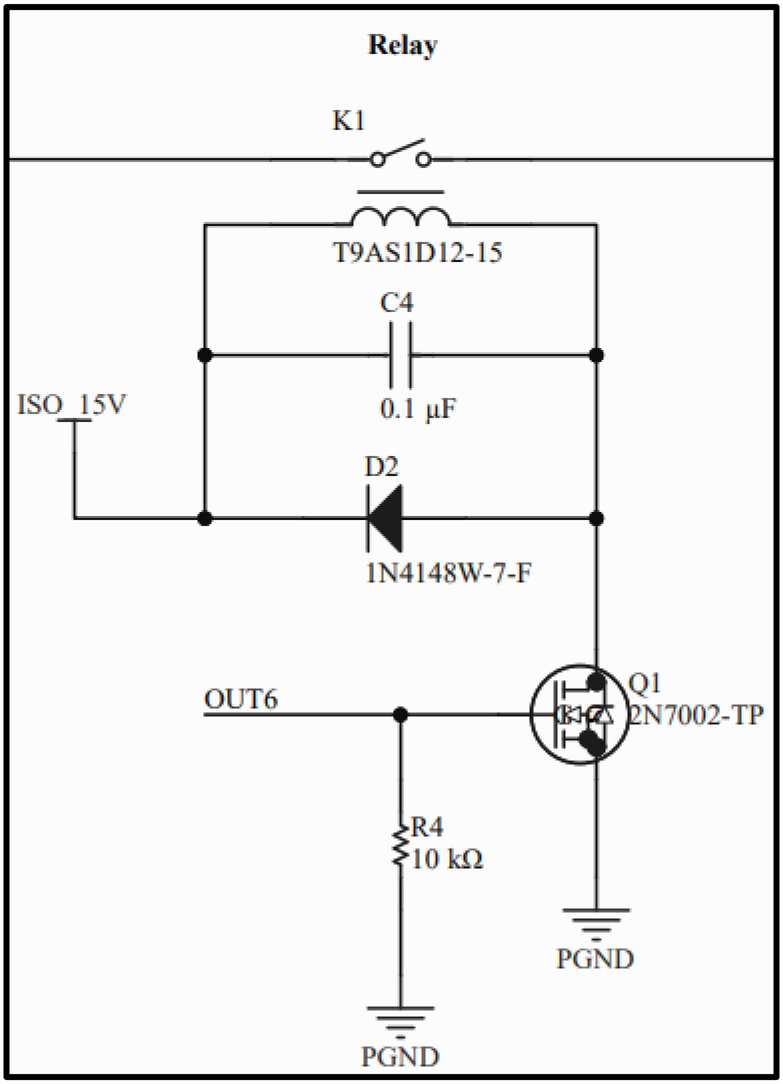

Figure 32: Relay Schematic

The relay was validated by supplying an input voltage of 15 V_{DC} to the contact and connecting the 15 V_{DC} input to the coil. Next, the 5 V_{DC} relay signal was generated using a DC power supply to close the switch and allow the 5 V_{DC} to pass through the relay as measured in Figure 33. Then, the simulated relay signal was disconnected to open the switch and stop the 5 V_{DC} to pass through the relay as measured in Figure 34.

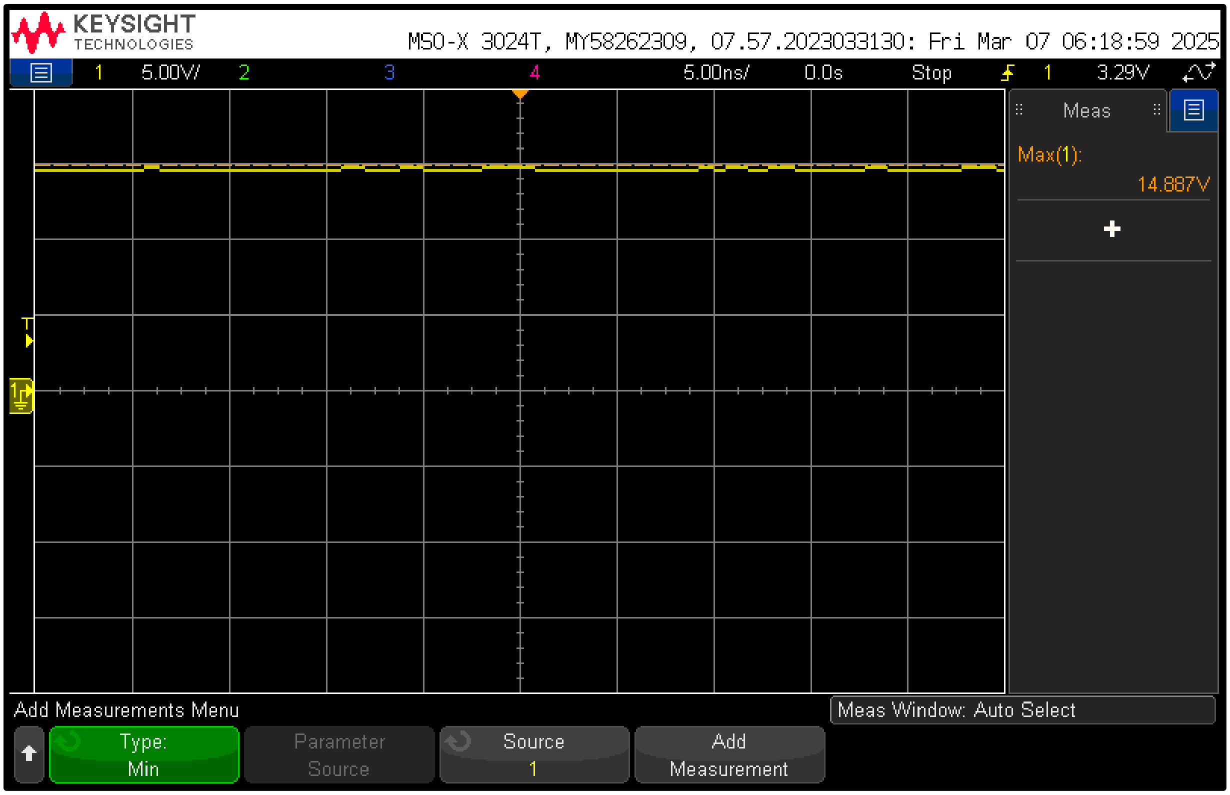

Figure 33: Relay On-State

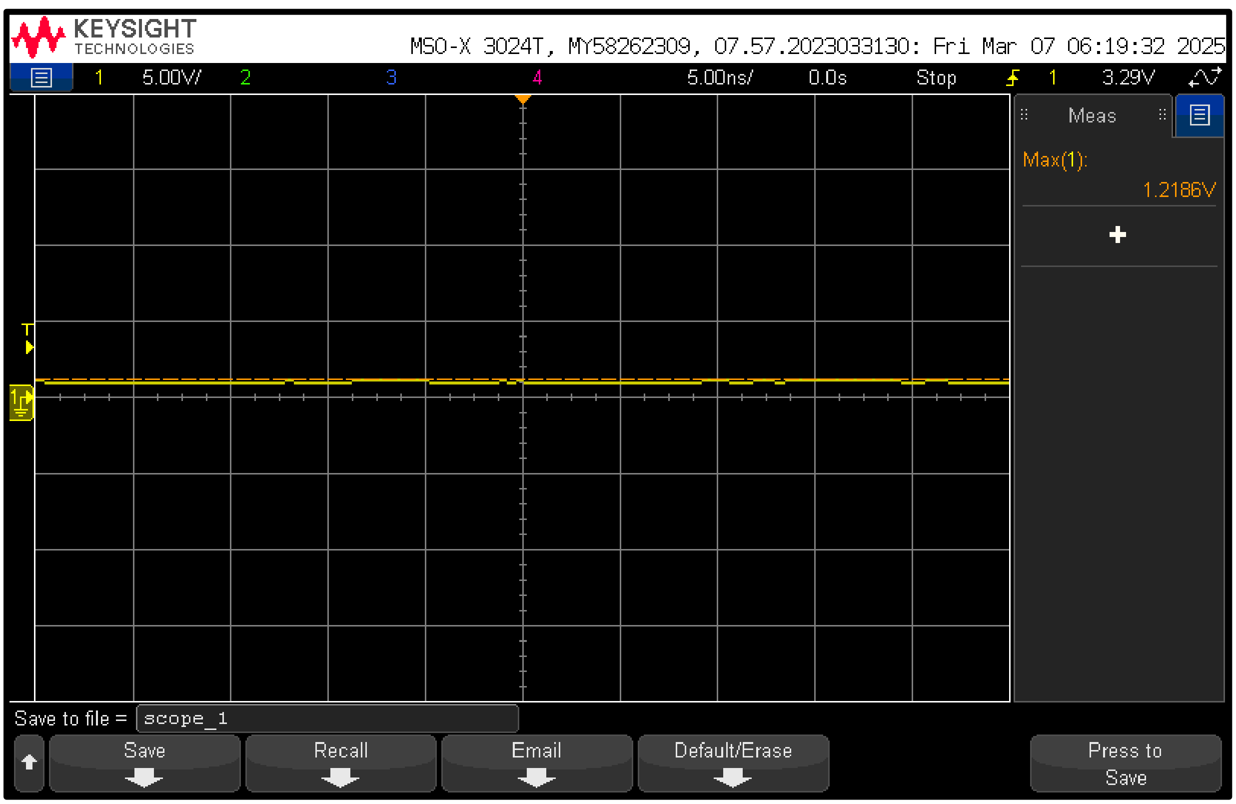

Figure 34: Relay Off-State

Upon measuring the output voltages in Figure 33, the yellow horizontal line shows approximately 15 V_{DC}. Upon measuring the output voltages in Figure 34, the yellow horizontal line shows approximately 1.2 V_{DC}. This aligns greatly with the expected change of 15 V_{DC} when toggling the relay. Any remaining voltage difference to zero, likely due to noise and measurement inadequacies.

4.2.6. Diagnostic Feedback Validation

The diagnostic feedback was designed to output the measurement signals of voltage, current, and temperature from the power control only when the full system is functioning. The voltage measurement came from the DC bus voltage after it has experienced voltage drop due to resistors. Due to this, low voltage testing resulted in an output value of zero because the DC bus voltage was not high enough. The current measurement came as a voltage from the power control into a current sensor designed to sense 30 A. Since this system was designed for 2 A, the voltage from the power control was not high enough and the current sensor resulted in an output value of zero. The temperature measurement came a power control pin and would have resulted in a voltage identifying the power control as overheating. Because the diagnostic feedback was pushed back to the end of the project and the full 120 V_{AC} was only utilized at demo, this portion of the project fell out of the completed scope.

4.3. Subsystem Conclusion

In conclusion the rectifier, DC link, power control, 3.3 V_{DC} to isolated 5 V_{DC} converter, 15 V_{DC} to isolated 15 V_{DC} converter, and relay were all working properly. On the other hand, the diagnostic feedback was not working properly. With a few more weeks the diagnostic feedback portion could have been adapted to the final project design, integrated with the rest of the project, and communicated the voltage, current, and temperature to the user via a monitor. Ultimately, the power subsystem succeeded in its objective to take in 120 V_{AC}, rectifier it to DC, invert it back to AC, and run the motor. It also succeeded in distributing auxiliary power, switching the motor on and off, and changing the motor speed in response to PWM signals.

5. Firmware Subsystem Report

5.1. Subsystem Introduction

This subsystem entails all programming required to make the microcontroller work. Without the microcontroller and its firmware, the remaining VFD subsystems could not function together to control the motor. The firmware is also responsible for presenting the user with an interface that allows direct control of the VFD's start/stop functionality, and PWM frequency (motor speed).

5.2. Subsystem Details

The firmware was all written, tested, and demonstrated in MPLAB's X IDE in the C language using a DSPIC33CK256MP508 microcontroller on a dsPIC33CK Curiosity Development Board (part number DM330030). There were three primary objectives of the firmware as part of this system. The first being to output a three phase PWM signal with adjustable frequencies based on a potentiometer. The second objective is to toggle a relay signal when a button is pressed on the microcontroller's PCB. Finally, the firmware should take three feedback signals that are input to the microcontroller and output them to the UART console. Two of these three objectives were successfully validated; the group decided to abandon the objective regarding the feedback signals due to problems with the values that they would send, and problems getting print statements to correctly appear in the UART console.

The program works by continuously reading the potentiometer's value and using that to calculate the desired frequency (10-60Hz) that should be outputted to the motor via the PWM signal. The program generates a table of sine values that are then mapped to duty cycle percentages between 1-100%. It then uses the High Speed PWM (HS PWM) module in MPLAB X IDE to produce 3 1kHz frequency PWM waves at a 120-degree offset (three phase) and their low counterparts. Upon calculating the desired 10-60Hz frequency based on the potentiometer value, the firmware calculates a step increment to determine how fast the PWMs are set to iterate through the sine table, which determines the frequency of these changing waves. This sine wave is the beginning of what the motor will eventually receive as an AC power signal after passing through the other three subsystems.

5.3. Subsystem Validation

To validate my subsystem, I tested it on the dsPIC33CK Curiosity Board. I was able to get it to properly display three different phases of a PWM signal iterating through a sine waveform of duty cycles that were all correctly offset by 120 degrees. I was also able to program the complete functionality of the potentiometer so that it can increase and decrease the frequency of the PWM signal. Though the duty cycles of the PWM waves continuously change, the oscilloscope's output of these three phases in a single frame of the loop is shown in the figure below. This validation shows that the firmware is ready for subsystem integration next semester.

When the program is running on my development board, it also prints several variables in each iteration of the while loop to a terminal window that can be opened in MPLAB. The variables output in each iteration are shown in the figure below. These print statements were later found to be slowing down the frequency of output PWM waves. Upon deleting the print statements, the correct frequencies were output. It is worth noting that this problem can also be solved by changing the frequencies of the PWM waves inside a timer interrupt and keeping the print statements in the main while loop; however, when this change was made, the motor did not run for an unknown reason.

Tables 7-8 and Figures 14-22 (in the microcontroller subsystem report) additionally show the correct firmware validation with the microcontroller of the start/stop button (relay signal) and PWM signals.

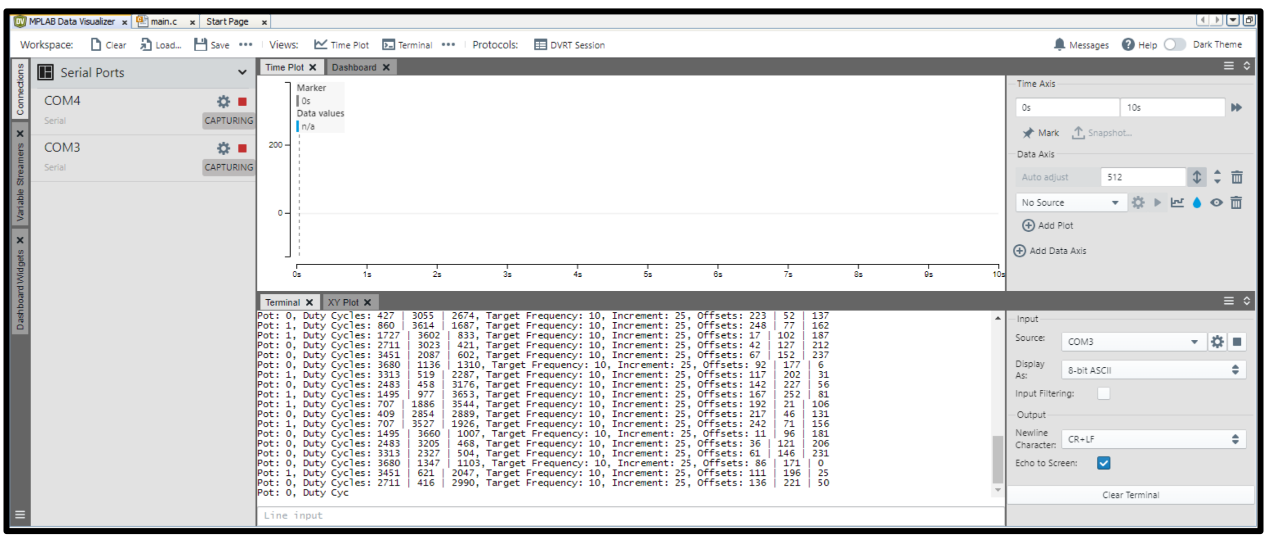

Figure 35: Debug Print Statements (Potentiometer at 0)

5.4. Subsystem Conclusion

All of my objectives for this project were complete or decided to be left incomplete. The motor was able to run in the end, meaning a successful complete integration between the four subsystems.

In the end, there were still a couple of problems left on the table to solve that could improve the VFD. First, the problem with the motor not turning when the PWMs are modified within a timer interrupt would need to be fixed to use print statements without messing up the PWM frequencies. Then, the problem with the microcontroller's communication with the UART console would need to be fixed to print any feedback signals. Lastly, there could be additional control added to the button that changes the direction of the motor. This would be done by swapping two of the PWM’s phases upon a certain duration press of the button or pressing the button a certain number of times in-sequence.

Overall, the firmware subsystem is validated, integrated, and – with the use of several solutions to issues that were found and documented throughout each presentation throughout the semester, provides a strong foundation for the future improvement of the programming of the VFD system.

VFD Motor Controller

Mackenzie Miller
Andrew Nguyen
Aidan Rader
Ryan Regan

FINAL SYSTEM REPORT

REVISION – 0
28 April 2025

FINAL SYSTEM REPORT

FOR

VFD Motor Controller

PREPARED BY:

Author Date

APPROVED BY:

Project Leader Date

John Lusher II, P.E. Date

T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
0	04/28/2025	VFD Motor Controller Team		Draft Release

Table of Contents

List of Figures

No table of figures entries found.

1. System Introduction

Our team was tasked with developing a Variable Frequency Drive (VFD). Most of the traditional motor control methods are inefficient and can lead to unnecessary energy consumption and wear on equipment. The aim of a VFD is to solve this problem by more precisely adjusting motor speed with frequency and voltage control, which helps to extend equipment lifespan and optimize performance.
Some key engineering challenges in our project include designing an efficient power conversion system, generating accurate three-phase pulse-width modulation (PWM) signals, and ensuring smooth integration with industrial applications. This project will help our sponsor and any other users to lower operational costs, enhance process control, and promote more sustainable energy use.

1.1. Block Diagram

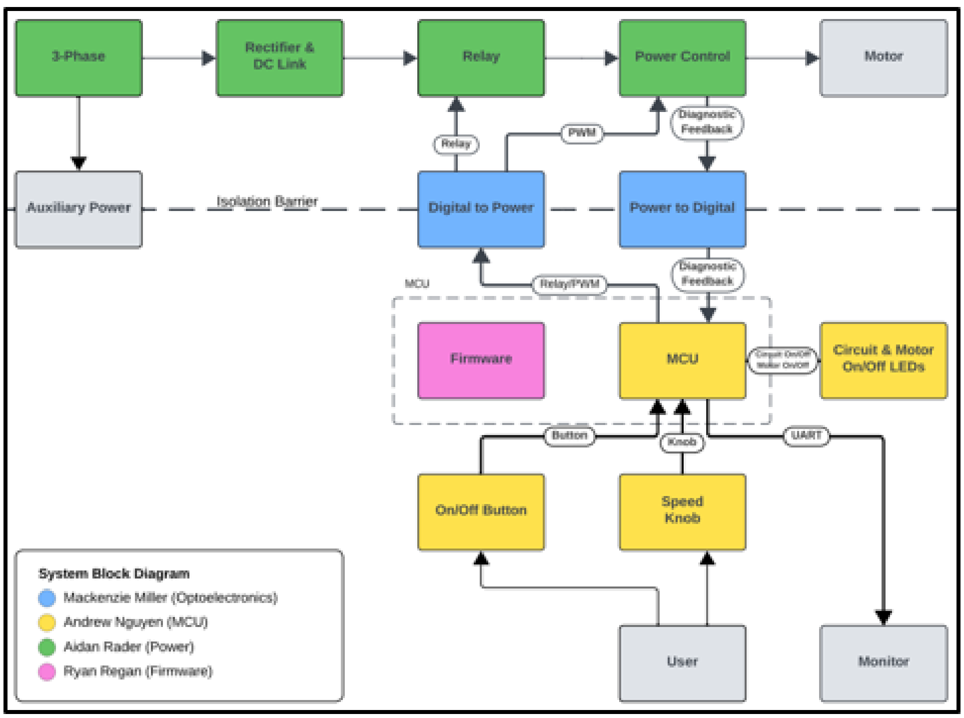

Figure 1: VFD Motor Controller Block Diagram

2. Integration

2.1. Auxiliary Power Integration

Auxiliary power integration includes taking 120VAC into the system from the wall, and transforming it to 15V, Iso15V, 3,3V, and Iso5V.The purpose of the Iso15V and Iso5V is to preserve the isolation barrier, where these two voltages have a different ground than the non-isolated voltages. Auxiliary power serves to create the voltages needed to power all of the components in the system. During integration of this part of the system, there were several challenges to overcome. The two main issues were component failure and testing errors. The 120VAC to 15V converter stopped working unknowingly sometime before the final demonstration. This component was not needed for testing at low voltage without the wall power, so when it came time to plug the system into the wall, this converter did not work. The converter was bypassed during the demo by applying 15V from a power supply to the output of the converter, and this showed that the auxiliary power worked for the rest of the system. Testing precautions were very important in testing the auxiliary power integration due to the high voltage coming from the wall. This made maintaining the isolation barrier and not mixing the grounds extremely important. During testing at one point, an oscilloscope probe was placed to measure the circuit which broke the isolation barrier. This broke the rectifier and exploded a trace out of the power electronics board. If the 120VAC to 15V converter had been replaced, the auxiliary power integration would have been a total success. Since this is a minor issue and was validated before breaking, auxiliary power was a success.

- ☐ 15 VDC to isolated 15 VDC Conversion

The 15 V_{DC} to isolated 15 V_{DC} converter takes 15 V_{DC} and outputs 15 V_{DC}. Parallel capacitors to the ground are used to filter and smooth the input and output voltages.

- ☐ **15 V_{DC} to 3.3 V_{DC} Conversion**

The 15 V_{DC} is sent to the 3.3 V_{DC} buck converter and outputs 3.3 V_{DC}. Parallel capacitors to the ground are used to filter and smooth the input and output voltages.

- ☐ **3.3 V_{DC} to Isolated 5 V_{DC} Conversion**

The 3.3 V_{DC} to isolated 5 V_{DC} converter takes 3.3 V_{DC} and outputs 5 V_{DC}. Parallel capacitors to ground are used to filter and smooth the input and output voltages.

- ☐ **120 V_{AC} to 15 V_{DC} Conversion**

The 120 V_{AC} from the wall outlet is taken in by the 15V AC/DC converter and outputs 15 V_{DC}. Parallel capacitors to the ground are used to filter and smooth the input and output voltages.

☐ **System On/Off Button**

An LED is connected to the MCU to indicate whether the system is on or off. The red through-hole LED will light up indicating the system is powered on and turn off when the system is not receiving power.

2.2. PWM Integration

PWM Integration consists of six pulse width modules (PWMs) that were created in the firmware, Figure 2, and flashed onto the microcontroller (MCU) being output at 3.3V, 120 degrees apart and input to the optoelectronics (opto) board where they are transported to the power electronics across an isolation barrier and brought to 5V. These 5V PWMs are then input into the power control where they are transformed into modified sine waves that spin the motor.

This integration happened in multiple steps. The first step was firmware on the development (dev) board connected to the optoelectronics board. The purpose of this was to test the isolation and voltage increase of the optoelectronics board while the MCU board was still in progress. Once the output of the opto board was validated with the dev board, the MCU was added to the integration. The firmware was flashed onto the DSPICK33 using a PICkit 5. A PICkit 4 was originally used, but for some reason the code flashing was unsuccessful. Once the output PWMs of the MCU were validated, as seen in Figure 3, six PWMs, three phases high and low, at 3.4V, the opto board was connected. The input PWMs for isolation were the same PWMs being output of the MCU, and the output PWMs can be seen in Figure 4, the same PWMs as the input, just at 6.0V and 6.2V. After this was validated, the power control was connected, and the output was observed to be a modified sine wave as seen in Figure 5. The final step of PWM integration was the motor spinning. Once the professor confirmed the power control output was correct, the motor was connected, and when it was powered up at a very low voltage, it spun. This completed PWM integration.

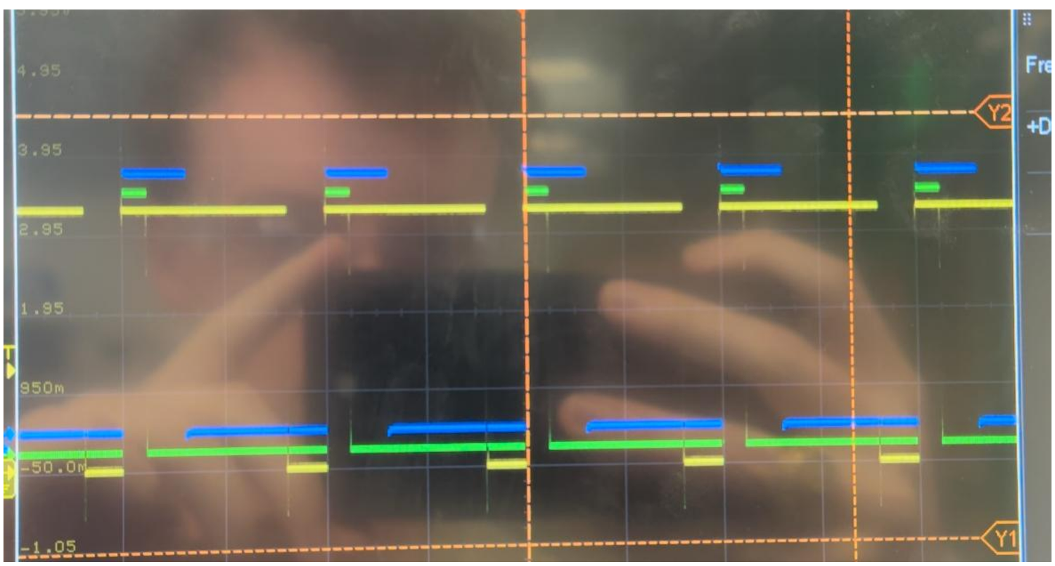

Figure 2: Oscilloscope output of PWMs created by Firmware

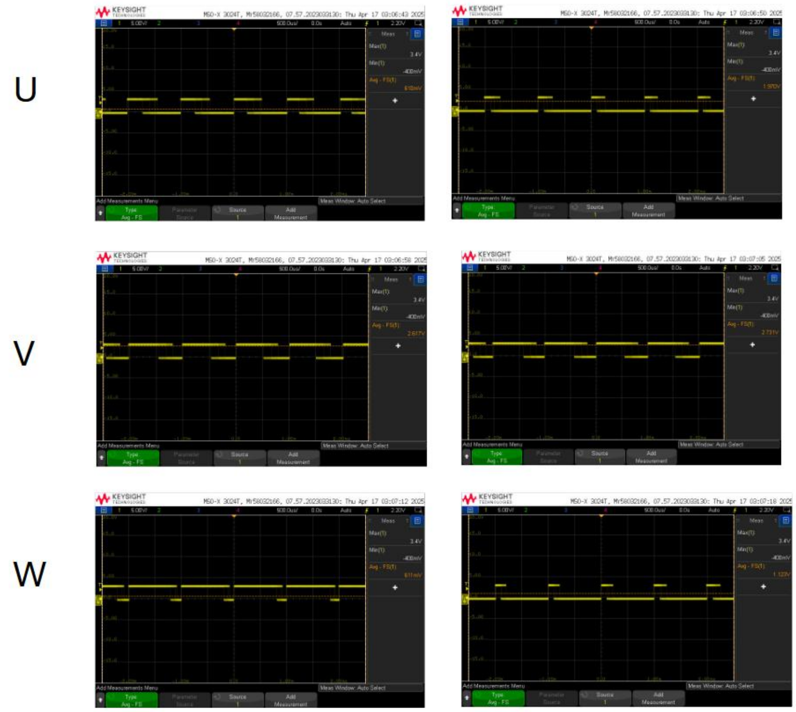

Figure 3: MCU output & Isolation Input PWMs (3.4V)

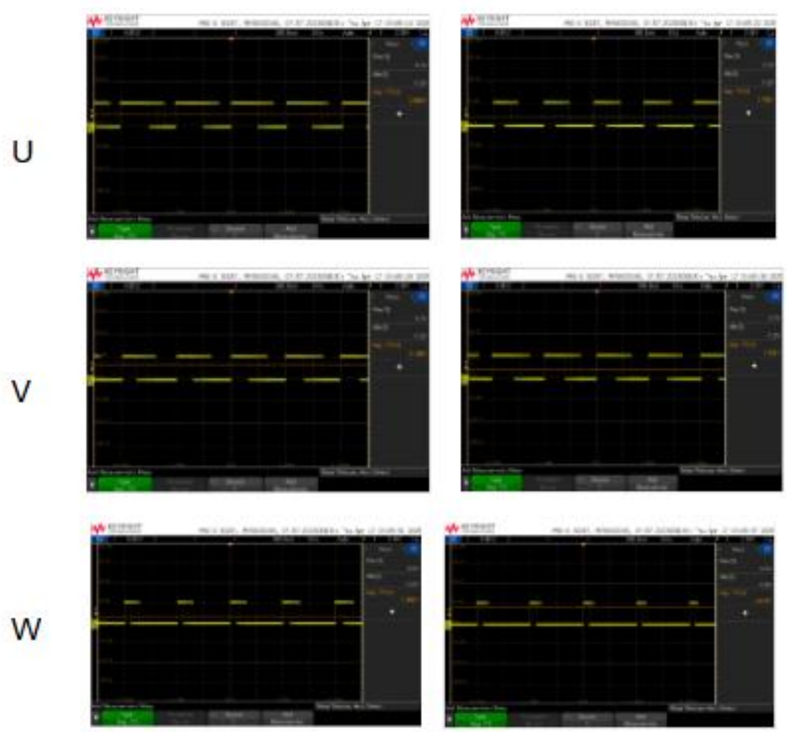

Figure 4: Isolation output PWMs (6.0V-6.2V)

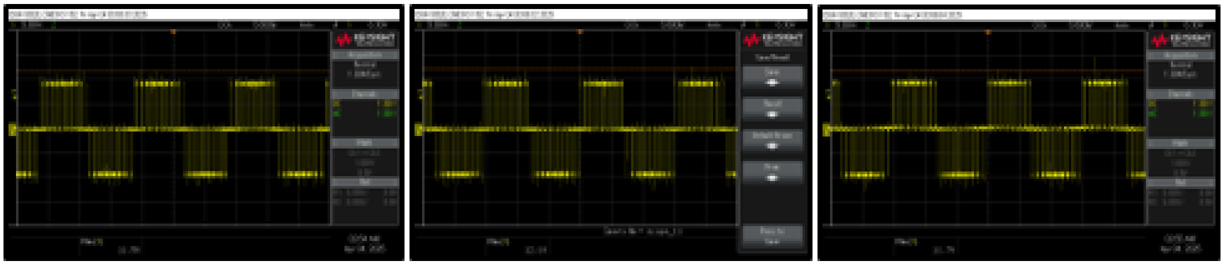

Figure 5: Output of the power control for all three phases

PWM integration was the most integral part of this system to get the motor to spin. There were many errors along the way that definitely slowed the process of getting the motor to spin, even at low voltage. A few notable hurdles were component failure, trace errors, and code errors. At one point, the optoelectronic board was working as expected, but due to a testing error, all of the isolators fried. This delayed the integration process because it took multiple days to get new parts in the mail and replace them. Additionally, on the power board, there was a trace error on the 0th revision causing a short of the power control, but was resolved on the next revision. Furthermore, the next revision had two more trace errors that were ultimately bypassed by over the board wires. One error was the bus voltage being used as the voltage measurement signal. Another error was the bootstrap capacitors being incorrectly in parallel with the high voltage outputs to the motor. Near the end of the project, the motor was spinning for a few days, then a new code was flashed to attempt our diagnostic feedback integration, and it ended up not working, so we reverted back to the old code backup.

- ☐ **Speed Knob Functionality**

One main requirement of the VFD motor controller is the ability to change the speed of the motor. In this system, a potentiometer, or speed knob, is how the speed controllability is implemented. When the potentiometer is spun, the frequency of the PWMs changes from 10Hz to 60Hz, linearly. In terms of integration, the knob was spun to adjust the speed at every step of integration and it worked pretty well. At the end of the project, the speed of the motor was the highest when the knob was in the middle, and it decreased as it was turned left or right. It is unknown why this happened, possibly an error in the code, but since this was a very minor error, it was not the team’s main focus. Overall, the speed knob did change the speed of the motor in a predictable way, so it was validated.

2.3. Relay Integration

Relay integration in this system is realized by a button on the MCU, isolation via optoelectronics, and the relay on the power electronics board. When the button is pressed, a second LED turns on to indicate the motor is on and the relay signal gets set to 15V. This validates that the MCU is on and sending out a 3.3V signal that goes through isolation and to the power control. When the button is pressed again, the relay signal goes to 3.3V which cuts off power to the motor. This means the motor is off. During integration there were a few instances where the relay stopped clicking when the button pressed on, meaning that the relay was not working. This, however, was quickly fixed by replacing the MOSFET in that circuit that had broken.

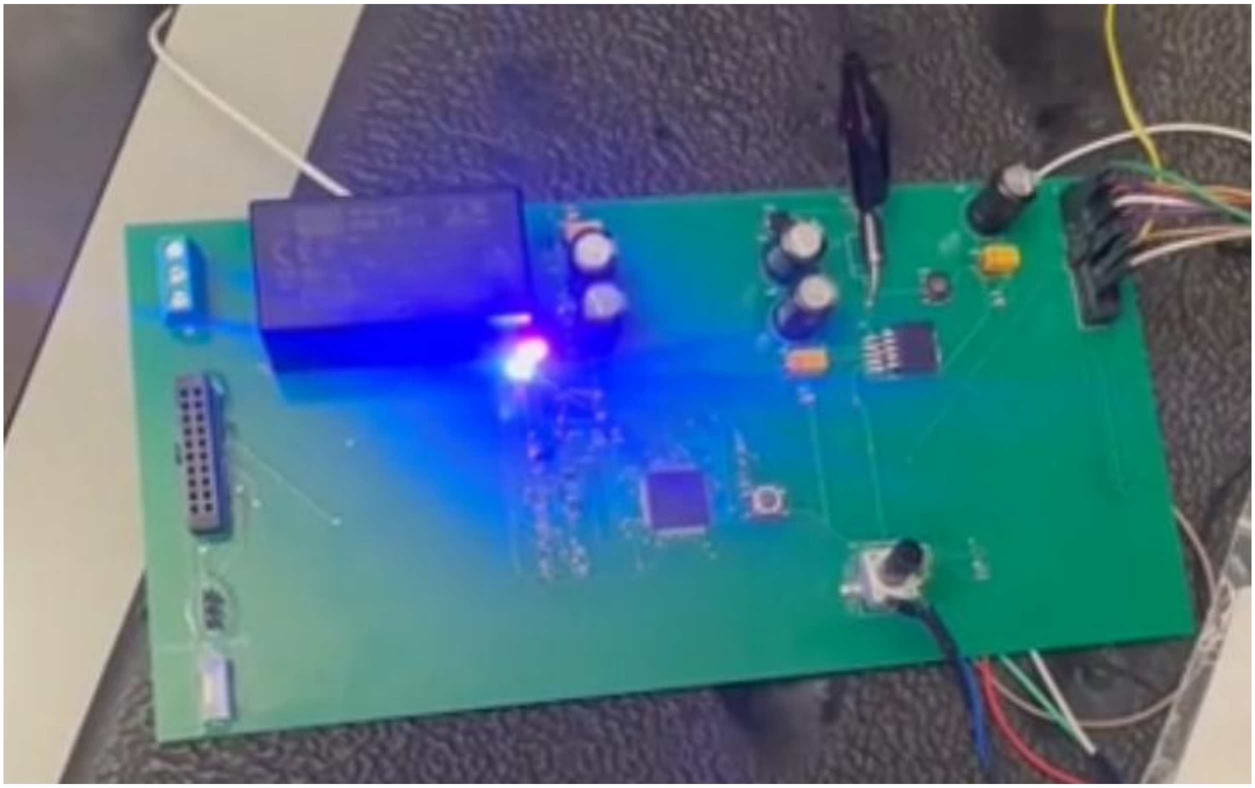

Figure 15: Motor On LED (blue)

2.4. Diagnostic Feedback Integration

Diagnostic feedback in this system was designed as a way to check the voltage, current, and temperature of the system at all times. This includes the three signals sent out of the power control on the power electronics board. These signals, each at around 60V were then taken as input to the optoelectronics board where they were decreased and then transported across the isolation barrier to

Diagnostic feedback was not integral to making the motor spin, so it was essentially saved for last. Once the motor was spinning and all other integration was completed, diagnostic feedback came into focus. There were some issues, however. On the power control side, the current was too low to be sensed and communicated, and the temperature also did not work. The voltage signal, which represents the bus voltage, should have worked when plugged into the wall, at the full 120VAC, but for all testing besides final demo, the voltage was much lower, between 30V and 51V. Since this voltage was significantly lower than 120VAC, it is believed that the value was too low, so that feedback signal was not being communicated.

Additionally, there were problems with the microcontroller printing to the UART console. Upon linking the correct UART “transfer” and “receive” pins in the firmware, the USB-UART cable was still unable to transfer any print statements to the UART console.

In terms of validation, the optoelectronics board was tested at low voltage, with 15V input. The input to output ratio is 1V:0.26V, so if the full 60V had been applied, it is reasonable to assume that 15.6V would be output which is within the acceptable region.

3. System Conclusion

The VFD Motor Controller project successfully met the goal of creating a system to control the speed of an AC motor by adjusting the input frequency and voltage. The motor spun successfully, and the speed knob and on/off button worked to control the motor. All four subsystems – including power, microcontroller, optoelectronics, and firmware – were successfully integrated and validated.

Though the potentiometer did not quite work as expected, it is most important that the motor worked and was controllable. The auxiliary power was officially finished during the final demonstration where it was successful in power the motor and components. The 120VAC to 15V converter failed during this test, but it was working previously, so it was assumed to just be a broken part. When this converter was bypassed, the motor spun as expected indicating the success of auxiliary power and PWM integration at full voltage, 120VAC. The motor also responded to the on/off button at low and high voltage, indicating success of relay integration. Diagnostic feedback was not quite successful mainly due to lack of time, but it was deemed an accessory feature to this project. Had there been a few more weeks, the problems could have been ironed out, and there would be feedback displayed on a UART console.