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

[Table of Contents III](#_Toc196770602)

[List of Tables IV](#_Toc196770603)

[List of Figures V](#_Toc196770604)

[1. Introduction 1](#_Toc196770605)

[2. Optoelectronics Subsystem Report 2](#_Toc196770606)

[2.1. Subsystem Introduction 2](#_Toc196770607)

[2.2. Subsystem Details 2](#_Toc196770608)

[2.3. Subsystem Validation 4](#_Toc196770609)

[2.3.1. Digital to Power Isolation Validation 5](#_Toc196770610)

[2.3.2. Power to Digital Isolation Validation 10](#_Toc196770611)

[2.4. Subsystem Conclusion 12](#_Toc196770612)

[3. Microcontroller Subsystem Report 13](#_Toc196770613)

[3.1. Subsystem Introduction 13](#_Toc196770614)

[3.2. Subsystem Details 13](#_Toc196770615)

[3.3. Subsystem Validation 15](#_Toc196770616)

[3.3.1. Converter Validation 16](#_Toc196770617)

[3.3.2. Button and Relay Signal Validation 16](#_Toc196770618)

[3.3.3. LED Validation 17](#_Toc196770619)

[3.3.4. Knob and PWM Validation 21](#_Toc196770620)

[3.4. Subsystem Conclusion 22](#_Toc196770621)

[4. Power Subsystem Report 23](#_Toc196770622)

[4.1. Subsystem Introduction 23](#_Toc196770623)

[4.2. Subsystem Validation 23](#_Toc196770624)

[4.2.1. Rectifier and DC Link Validation 23](#_Toc196770625)

[4.2.2. Power Control Validation 27](#_Toc196770626)

[4.2.3. 3.3 VDC to Isolated 5 VDC Converter Validation 28](#_Toc196770627)

[4.2.4. 15 VDC to Isolated 15 VDC Converter Validation 29](#_Toc196770628)

[4.2.5. Relay Validation 31](#_Toc196770629)

[4.2.6. Diagnostic Feedback Validation 34](#_Toc196770630)

[4.3. Subsystem Conclusion 34](#_Toc196770631)

[5. Firmware Subsystem Report 35](#_Toc196770632)

[5.1. Subsystem Introduction 35](#_Toc196770633)

[5.2. Subsystem Details 35](#_Toc196770634)

[5.3. Subsystem Validation 35](#_Toc196770635)

[5.4. Subsystem Conclusion 36](#_Toc196770636)

List of Tables

[*Table 1: PWM Input and Output Isolation Voltages* 8](#_Toc196765967)

[*Table 2: Diagnostic Feedback Input and Output Voltages* 11](#_Toc196765968)

[*Table 3: Pin interface for DSPIC33CK256MP508 Microcontroller* 15](#_Toc196765969)

[*Table 4: Load regulation validation for buck converter (LM2595s-3.3)* 16](#_Toc196765970)

[*Table 5: Line regulation validation for buck converter (LM2595s-3.3)* 16](#_Toc196765971)

[*Table 6: Validation for 15V AC/DC converter (IRM-15-15)* 16](#_Toc196765972)

[*Table 7: Button signal voltage when button is toggled* 17](#_Toc196765973)

[*Table 8: Relay signal voltage when toggled* 17](#_Toc196765974)

[*Table : 3.3V/ISO\_5V Converter Load Regulation Test* 29](#_Toc196765975)

[*Table : 3.3V/ISO\_5V Converter Line Regulation Test* 29](#_Toc196765976)

[*Table : 15V/ISO\_15V Converter Load Regulation Test* 30](#_Toc196765977)

[*Table : 15V/ISO\_15V Converter Line Regulation Test* 31](#_Toc196765978)

List of Figures

[*Figure 1: Power to Digital Schematic* 3](#_Toc196770555)

[*Figure 2: Digital to Power Schematic* 3](#_Toc196770556)

[*Figure 3: Power to Digital Portion of PCB* 4](#_Toc196770557)

[*Figure 4: Digital to Power Portion of PCB* 4](#_Toc196770558)

[*Figure 5*: *Cut Traces and External Wires to Fix Op-Amps* 5](#_Toc196770559)

[*Figure 6: PWM input – Before Isolation* 6](#_Toc196770560)

[*Figure 7: PWM Output – After Isolation* 7](#_Toc196770561)

[*Figure 8: Relay Isolation Input Button On (3.4V)* 9](#_Toc196770562)

[*Figure 9: Relay Isolation Output Button On (5.8V)* 9](#_Toc196770563)

[*Figure 10: Relay Isolation Output Button Off (~0V)* 10](#_Toc196770564)

[*Figure 11: 15V Supply Voltage Showing No Drop* 11](#_Toc196770565)

[*Figure 12: MCU Block Diagram* 14](#_Toc196770566)

[*Figure 13: MCU subsystem schematic* 15](#_Toc196770567)

[*Figure 14: Power On LED (red)* 18](#_Toc196770568)

[*Figure 15: Motor On LED (blue)* 18](#_Toc196770569)

[*Figure 16: MCU toggling motor on/off LED voltage (Off)* 19](#_Toc196770570)

[*Figure 17: MCU toggling motor on/off LED voltage (On)* 19](#_Toc196770571)

[*Figure 18: MCU toggling system on/off LED voltage (On)* 20](#_Toc196770572)

[*Figure 19: MCU toggling motor on/off LED voltage (Off)* 20](#_Toc196770573)

[*Figure 20: Three high and three low PWMs with knob signal* 21](#_Toc196770574)

[*Figure 21: Knob position and corresponding voltage level* 22](#_Toc196770575)

[*Figure 22: Knob effect on PWM signals* 22](#_Toc196770576)

[*Figure 23: Rectifier Schematic* 23](#_Toc196770577)

[*Figure 24: DC Link Schematic* 24](#_Toc196770578)

[*Figure 25: Rectifier and DC Link Circuit* 25](#_Toc196770579)

[*Figure 26: Input Phase A and Phase B* 25](#_Toc196770580)

[*Figure 27: Input Phase C* 26](#_Toc196770581)

[*Figure 28: Input Phase A, Phase B, Phase C; Output DC* 27](#_Toc196770582)

[*Figure 29: Power Control Output* 28](#_Toc196770583)

[*Figure 30: 3.3V/ISO\_5V Converter Schematic* 28](#_Toc196770584)

[*Figure 31: 15V/ISO\_15V Converter Schematic* 30](#_Toc196770585)

[*Figure 32: Relay Schematic* 32](#_Toc196770586)

[*Figure 33: Relay On-State* 33](#_Toc196770587)

[*Figure 34: Relay Off-State* 33](#_Toc196770588)

[*Figure 36: Debug Print Statements (Potentiometer at 0)* 36](#_Toc196770589)

# 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.

# Optoelectronics Subsystem Report

## 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.

## 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.

A diagram of a circuit

Description automatically generated

*Figure 1: Power to Digital Schematic*

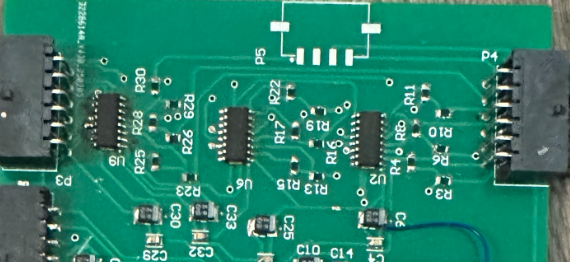
*Figure 2: Digital to Power Schematic*

A diagram of a circuit

Description automatically generated



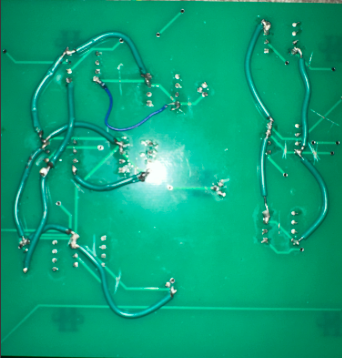
*Figure 3: Power to Digital Portion of PCB*



*Figure 4: Digital to Power Portion of PCB*

## Subsystem Validation

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.



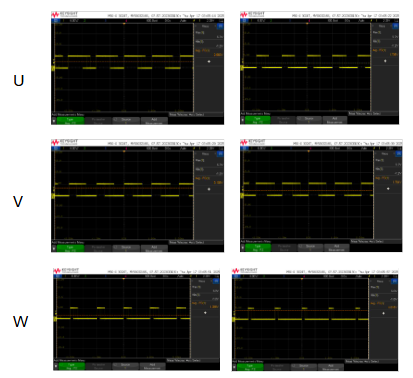
*Figure 5*: *Cut Traces and External Wires to Fix Op-Amps*

### 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.

A screen shot of a graph

AI-generated content may be incorrect.

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

A screen shot of a graph

AI-generated content may be incorrect.

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

A screen shot of a graph

AI-generated content may be incorrect.

*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.

### 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.

## 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.

# Microcontroller Subsystem Report

## 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.

## 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.

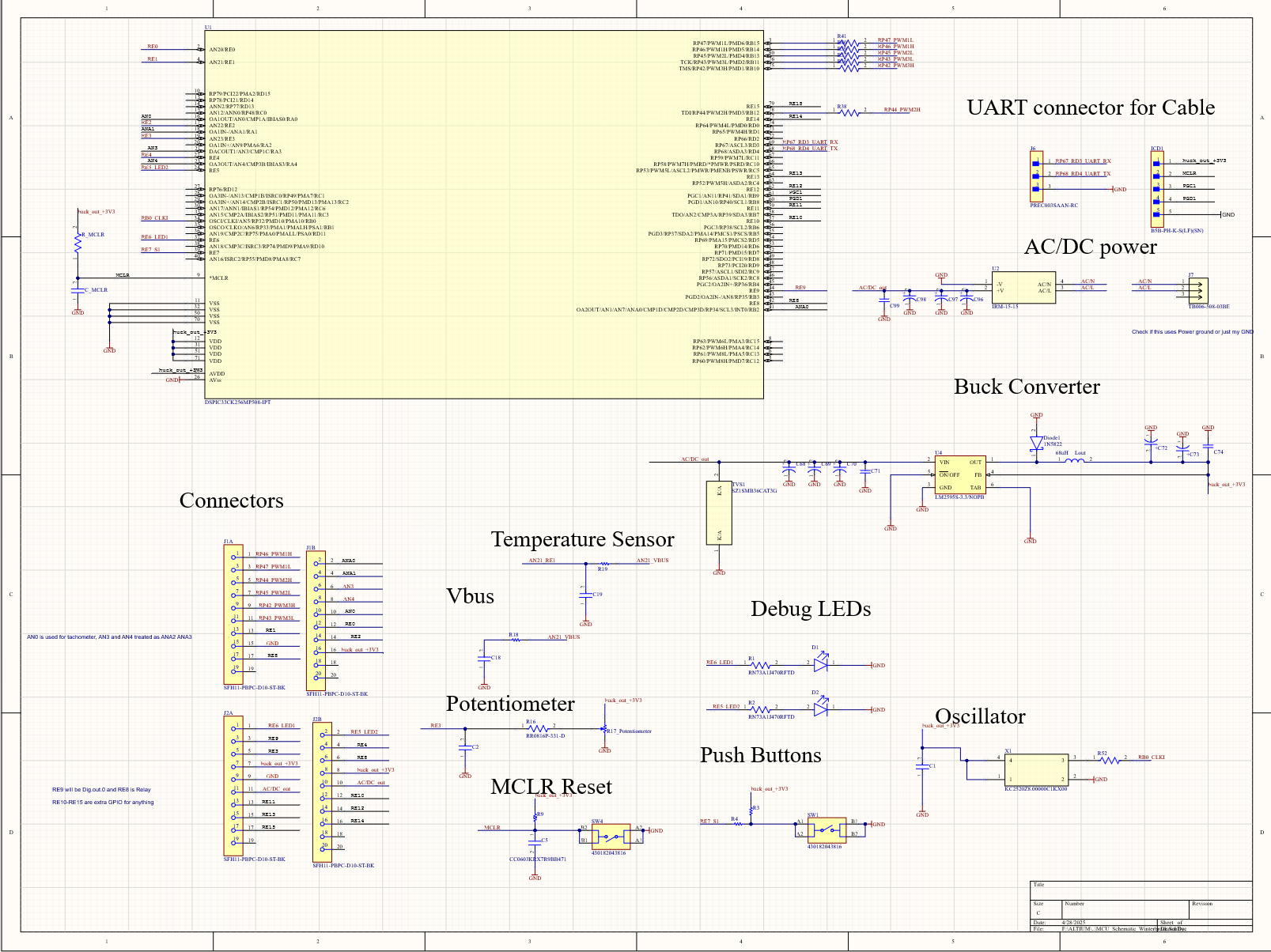
A diagram of a computer

AI-generated content may be incorrect.

*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*

## 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.

### 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 (LM2595s-3.3)*

|  |  |
| --- | --- |
| Vin (V) | Vout (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 (LM2595s-3.3)*

|  |  |
| --- | --- |
| Vin (VAC) | Vout (VDC) |
| 118 | 15.0 |

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

### 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 |

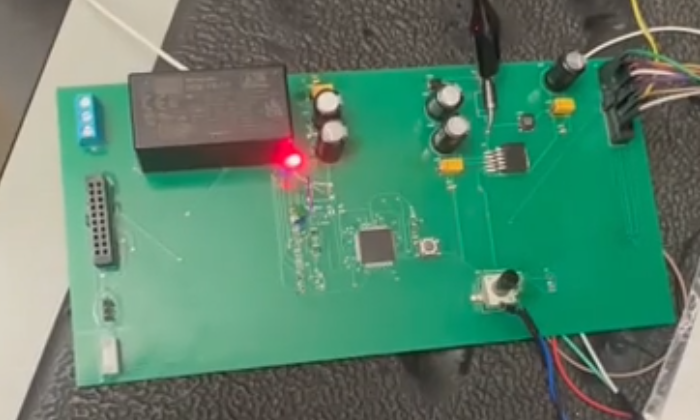
*Table 7: Button signal voltage when button is toggled*

|  |  |
| --- | --- |
| Relay |  |
| On | Off |
| 3.281V | -.045V |

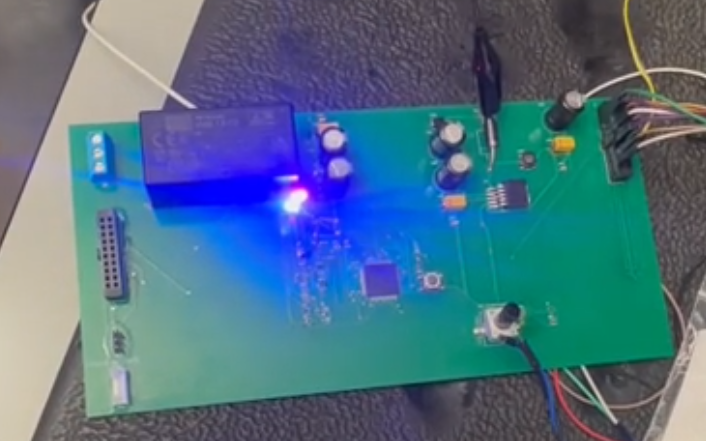
*Table 8****:*** *Relay signal voltage when toggled*

### LED Validation

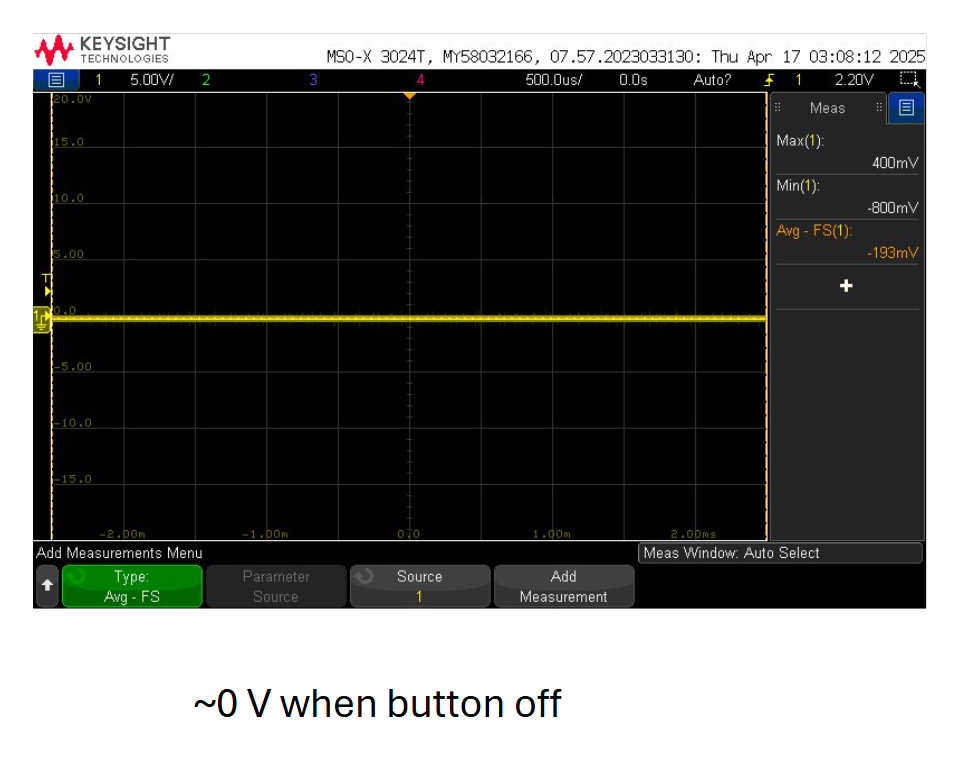
After ensuring the button and relay works the LEDs were validated. The original LEDs failed to function properly due to improper soldering, however the component was small and there was a risk of burning a pad when trying to get them off. The addition of two LEDs -- one red and one blue – was done as 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.



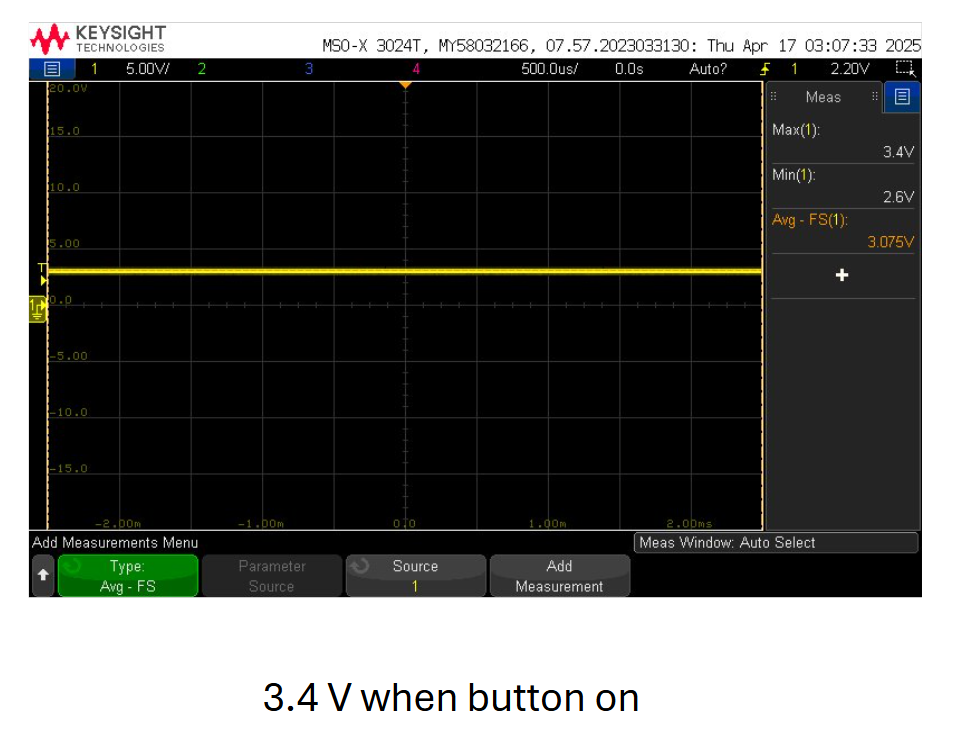
*Figure 14: Power On LED (red)*



*Figure 15: Motor On LED (blue)*



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



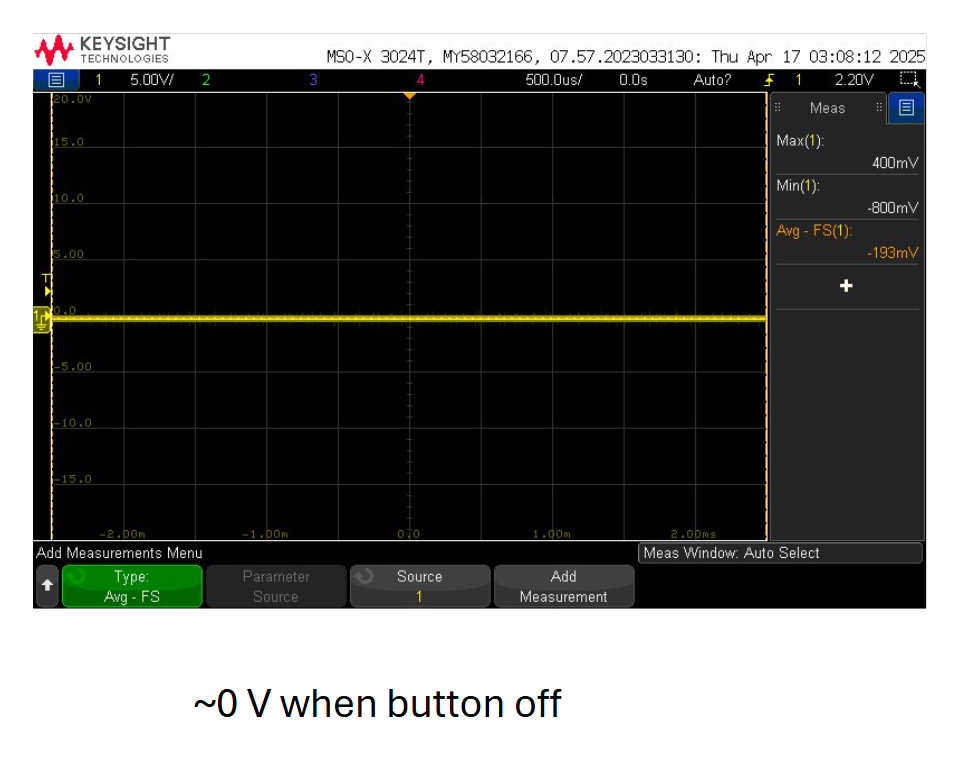
*Figure 17: MCU toggling motor on/off LED voltage (On)*

A screen shot of a graph

AI-generated content may be incorrect.

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

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



### 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*

## 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 LM2595s-3.3 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.

# Power Subsystem Report

## 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 VAC 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.

## Subsystem Validation

### 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.

A diagram of electrical components

AI-generated content may be incorrect.

*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.

A diagram of a circuit

Description automatically generated

*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 VAC, the validation testing was conducted at 10 VAC. 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 Vrms 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.

A circuit board with wires

Description automatically generated

*Figure 25: Rectifier and DC Link Circuit*

A screen shot of a computer

Description automatically generated

*Figure 26: Input Phase A and Phase B*

A screen shot of a computer

Description automatically generated

*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 VAC. Furthermore, the measured DC output is 9.6 VDC with 600 mV of noise. This differs slightly from the expected output of 12 VDC .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.

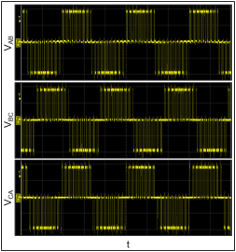
A screen shot of a computer

Description automatically generated

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

### 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*

### 3.3 VDC to Isolated 5 VDC Converter Validation

The 3.3 VDC to isolated 5 VDC converter converts 3.3 VDC to isolated 5 VDC. In Figure 30, the 3.3V/ISO\_5V converter schematic is shown. First, an input connector receives the input 3.3 VDC. 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 VDC to 5 VDC. Finally, another set of capacitors in parallel is connected to ground, to filter the voltage and reduce noise.

A diagram of a voltage converter

AI-generated content may be incorrect.

*Figure 30: 3.3V/ISO\_5V Converter Schematic*

The 3.3 VDC to isolated 5 VDC 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 VDC using a DC power supply and the varying currents using an electronic load. The resulting data was recorded in Table 9.

|  |  |
| --- | --- |
| Iout (mA) | Vout (V) |
| 12 | 5.381 |
| 50 | 5.188 |
| 100 | 4.981 |
| 150 | 4.782 |
| 200 | 4.581 |

*Table : 3.3V/ISO\_5V Converter Load Regulation Test*

Furthermore, the 3.3 VDC to isolated 5 VDC 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 VDC and was generated using a DC power supply. The resulting output voltages were recorded in Table #.

|  |  |  |  |
| --- | --- | --- | --- |
| Vin (V) | Vin,pp (mV) | Vout (V) | Vout,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 : 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.

### 15 VDC to Isolated 15 VDC Converter Validation

The 15 VDC to isolated 15 VDC converter converts 15 VDC to isolated 15 VDC. In Figure 31, the 15V/ISO\_15V converter schematic is shown. First, an input connector receives the input 15 VDC. 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 VDC to 15 VDC. Finally, another set of capacitors in parallel is connected to ground, to filter the voltage and reduce noise.

A diagram of a remote control

AI-generated content may be incorrect.

*Figure 31: 15V/ISO\_15V Converter Schematic*

The 15 VDC to isolated 15 VDC 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 VDC using a DC power supply and the varying currents using an electronic load. The resulting data was recorded in Table 11.

|  |  |
| --- | --- |
| Iout (mA) | Vout (V) |
| 12 | 14.959 |
| 25 | 14.951 |
| 50 | 14.937 |
| 75 | 14.924 |
| 100 | 14.912 |
| 125 | 14.898 |
| 134 | 14.893 |

*Table : 15V/ISO\_15V Converter Load Regulation Test*

Furthermore, the 15 VDC to isolated 15 VDC 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.

|  |  |  |  |
| --- | --- | --- | --- |
| Vin (V) | Vin,pp (mV) | Vout (V) | Vout,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 : 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.

### 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.

A diagram of a circuit

AI-generated content may be incorrect.

*Figure 32: Relay Schematic*

The relay was validated by supplying an input voltage of 15 VDC to the contact and connecting the 15 VDC input to the coil. Next, the 5 VDC relay signal was generated using a DC power supply to close the switch and allow the 5 VDC 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 VDC to pass through the relay as measured in Figure 34.

*A screen shot of a graph

AI-generated content may be incorrect.*

*Figure 33: Relay On-State*

A screen shot of a graph

AI-generated content may be incorrect.

*Figure 34: Relay Off-State*

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

### 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 VAC was only utilized at demo, this portion of the project fell out of the completed scope.

## Subsystem Conclusion

In conclusion the rectifier, DC link, power control, 3.3 VDC to isolated 5 VDC converter, 15 VDC to isolated 15 VDC 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 VAC, 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.

# Firmware Subsystem Report

## 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).

## 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.

## 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.

A screenshot of a computer

Description automatically generated

*Figure : Debug Print Statements (Potentiometer at 0)*

## 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.