**University of Washington**

**Department of Electrical Engineering**

**BEE 425 Microprocessor System Design**

**Design Project**



**Jamil Fakhouri**

**Adonay Lebeneh**

Table of Contents

[Introduction 4](#_Toc35386366)

[Specifications 4](#_Toc35386367)

[Research 5](#_Toc35386368)

[Theory of Operation 5](#_Toc35386369)

[Power Supply 5](#_Toc35386370)

[Control 5](#_Toc35386371)

[Interface 6](#_Toc35386372)

[Software 6](#_Toc35386373)

[Design & Prototyping 6](#_Toc35386374)

[Initial Tests 6](#_Toc35386375)

[Op-Amp Circuit 8](#_Toc35386376)

[Power Supply 10](#_Toc35386377)

[Control 11](#_Toc35386378)

[PCB Design 12](#_Toc35386379)

[Enclosure 13](#_Toc35386380)

[Test Plan 13](#_Toc35386381)

[Bill of Materials 14](#_Toc35386382)

[References 14](#_Toc35386383)

[Appendix A 15](#_Toc35386384)

[Code 15](#_Toc35386385)

[Appendix B 21](#_Toc35386386)

[Schematics 21](#_Toc35386387)

[Appendix C 23](#_Toc35386388)

[Main PCB 23](#_Toc35386389)

[Power Supply PCB 25](#_Toc35386390)

[Front PCB 27](#_Toc35386391)

[Appendix D 29](#_Toc35386392)

[Enclosure 29](#_Toc35386393)

[Appendix E 32](#_Toc35386394)

[Operation Test Results 32](#_Toc35386395)



# Introduction

As a final project for BEE 425, we were to design a function generator from scratch that satisfies the following criteria:

* Amplitude adjustable from 0 to ±5Vpp
* Frequency adjustable from 10 to 100kHz
* Wave type adjustable, Sine, Triangle, and Square waves.
* Housed in a box
* Has no connection to a PC
* Must use a microprocessor as a control element

A function generator is an essential electronic lab test device that outputs desired signals for circuit testing. Designing such project will allow us to apply our new acquired knowledge of microprocessor and PCB design to practical use to produce a product that we could use as Electrical Engineers.

# Specifications

General Specifications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Min | Typ | Max | Unit |
| Input voltage | 98 | 110 | 120 | VAC |
| Input voltage frequency | 50 |  | 60 | Hz |
| Output voltage amplitude | 0 |  | 20 | Vpp |
| Maximum frequency | 1 MHz | | | |
| Resolution | 28-bit | | | |
| Channels | 1 | | | |
| Waveforms | Sine, Triangle, Square | | | |
| Dimensions | 200 x 150 x 100 | | | mm |
| Weight | 1.18 | | | Kg |
| Interface | N/A | | | |
| Output connection | Standard BNC | | | |

Frequency Specification

|  |  |
| --- | --- |
| Wave Type | Frequency |
| Sine | 1Hz ~ 1MHz |
| Triangle | 1Hz ~ 300KHz |
| Square | 1Hz ~ 300KHz |

Amplitude Attenuation Specifications

|  |  |  |
| --- | --- | --- |
| Wave Type | Frequency | Max Amplitude |
| Sine | 10Hz ~ 1MHz | 20Vpp ~ 6.5Vpp |
| Triangle | 10Hz ~ 300KHz | 20Vpp ~ 16.4Vpp |
| Square | 10Hz ~ 300KHz | 20Vpp ~ 20Vpp |

# Research

Waveforms can be generated using microprocessor coded with mathematical function to output the desired signal using Digital to Analog Conversion (DAC). This however requires a high performing microprocessor to achieve clean and accurate wave types at high frequencies. As we are unfamiliar with the recommended microcontroller board TIVA, and knowing that an Arduino would not be fast enough to produce the needed results, we decided to research and learn basic information regarding the project. This led us to find a popular Direct Digital Synthesis (DDS) chips from Analog Devices (AD98\*\*) that satisfies all the requirements, and requires a microprocessor for communications and control.

The [AD9833](https://www.analog.com/media/en/technical-documentation/data-sheets/ad9833.pdf) variant can produce all three wave types required at a cost of $10, also available in a breakout board form Amazon, and requires SPI serial communications to control it, thus any Arduino can communicate with it, therefore, we decided to use the [ATmega328](http://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061A.pdf) DIP28 package microcontroller so we can install it on an Arduino Uno board and program it. The AD9833 chip outputs a wave of 0.65Vpp amplitude centered at 0.325V (DC offset), thus an amplifier circuit will need to be utilized to achieve the desired amplitude, and a High-Pass filter or an adjustable offset feature to remove the inherent DC offset from the AD9833 output.

To interface with the ATmega328 and the DDS chip, we decided to use a standard 16x2 LCD (since we had some laying around) that uses an I2C protocol, which again the ATmega328 can handle. Standard tactile buttons will be used to change frequency and wave type. And finally, two potentiometers to adjust amplitude and offset.

# Theory of Operation

## Power Supply

The device will be power using 110VAC wall power fed from a standard computer power cord to a fused switch adapter. The voltage will be stepped down to two 12VAC voltages using a 24VAC center tap transformer, then rectified to DC using a full bridge rectifier. Using a full bridge rectifier will result in a voltage level higher than 12VDC, usually about 15-18VDC. To regulate the rectified AC voltage, multiple types of fixed voltage regulators needed to be used. Since the rectifier produces +15 and –15VDC, these voltages are regulated using [LM7812](http://www.ti.com/lit/ds/symlink/lm340.pdf) and [LM7912](http://www.ti.com/lit/ds/snosbq7c/snosbq7c.pdf) to produce +12 and -12VDC respectively. To produce +5VDC, an [LM7805](http://www.ti.com/lit/ds/symlink/lm340.pdf) was used and fed from the regulated 12VDC supply.

## Control

The AD9833 uses SPI serial communication protocol to output the desired wave and frequency. The commands are sent from the ATmega328 microcontroller which is programmed using the Arduino IDE tool with C++ code (Appendix A). The code uses the standard Arduino SPI library to send HEX words to the AD9833.

AD9833 uses clock and data signals from the ATmega328. It also uses a chip select (CS) pin called (FSYNC) to let it know that a command is being sent. The ATmega328 supply the clock pin (SCLK) and the data pin (SDATA), and the FSYNC pin could be any digital output.

To send a command, the FSYNC pin will have to be LOW, then data will be sent in HEX form as MSW, followed by LSW, then FSYNC pin will have to HIGH. This tells the AD9833 that the data stream is finished, and allows it to process the data and output the wave form.

## Interface

The user can change the wave type using a button on the front panel. When the device is turned on, the wave type is set to default to a Sine wave. Every time the wave type button is pressed, the microcontroller is programed to cycle through the wave types as follows:

Sine 🡪 Triangle 🡪 Rectangle 🡪 Sine

Frequency is changed using a set of four buttons, Left, Right, +, -. The left and right buttons select the digit of the frequency range, 1, 10, 100, 1k, 10k, 100k. The +/- buttons are used to increment and decrement the value of the selected digit from 0-9.

Amplitude is adjusted using a knob, this knob is a potentiometer that when adjusted changes the gain on the amplifier circuit through the feedback resistor network. Finally, offset is adjusted using another knob, this knob is yet another potentiometer that is connected to the input of the amplifier circuit in which it adds either positive or negative DC voltage to the output of the AD9833.

## Software

The software is coded in C++ and is flashed onto the ATmega328 microcontroller using an Arduino Uno board. The software code controls the communications to the AD9833 and interpret the input from the user and output the necessary commands. See appendix A for code.

# Design & Prototyping

## Initial Tests

Using the AD9833 breakout board and an ATmega328 microcontroller (Figure 1.0), we attempted to communicate to the chip and learned how to send SPI commands to it. This allowed us to confirm the expected waveforms. The Sine and Triangle wave forms are indeed 0.65Vpp centered at 0.325V (Figure 1.1). However, the square wave is a surprising 0-5V wave (Figure 1.2).

A circuit board

Description automatically generated

Figure 1.0: ATmega328 and AD9833 coding test.

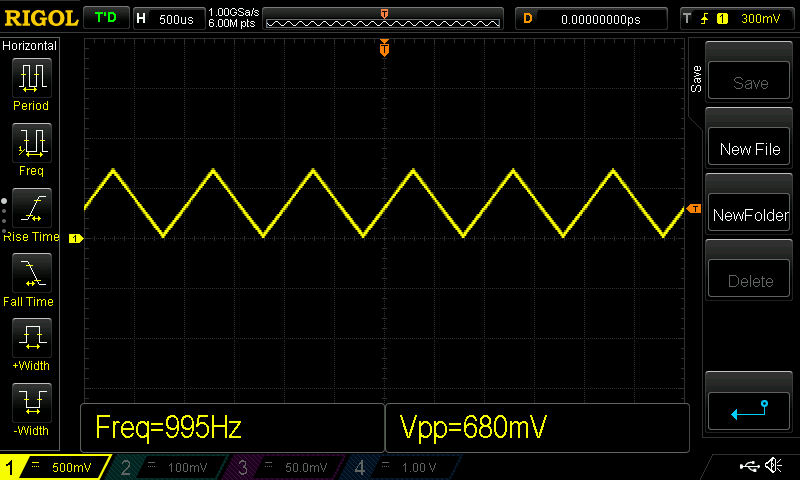
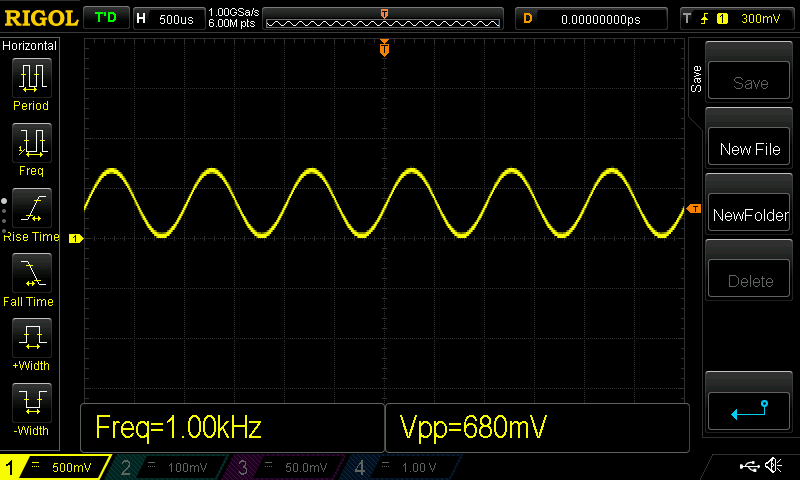


Figure 1.1: AD9833 Sine and Triangle output waveforms.

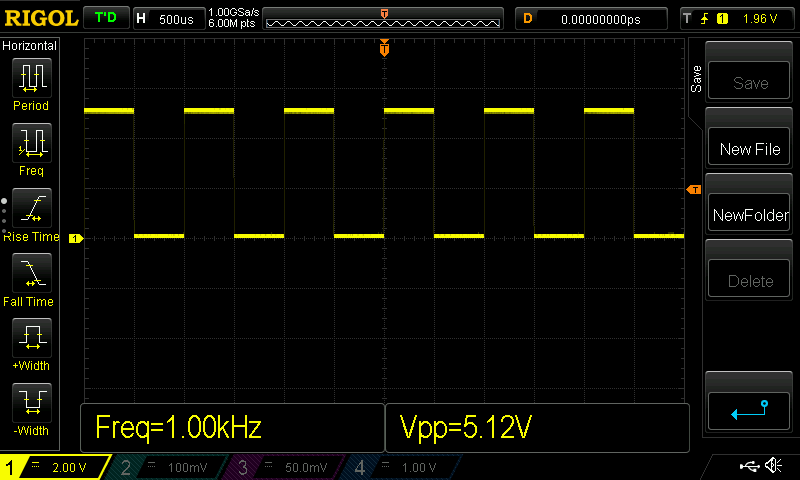


Figure 1.2: AD9833 Square wave output waveform.

## Op-Amp Circuit

To produce at least ±5Vpp waves, we designed a Non-Inverting amplifier circuit with a gain of at least 25.1dB (18 V/V) using LM318N Op-Amp (Figure 1.3). Also integrated the offset circuit to the amplifier (Figure 1.4).

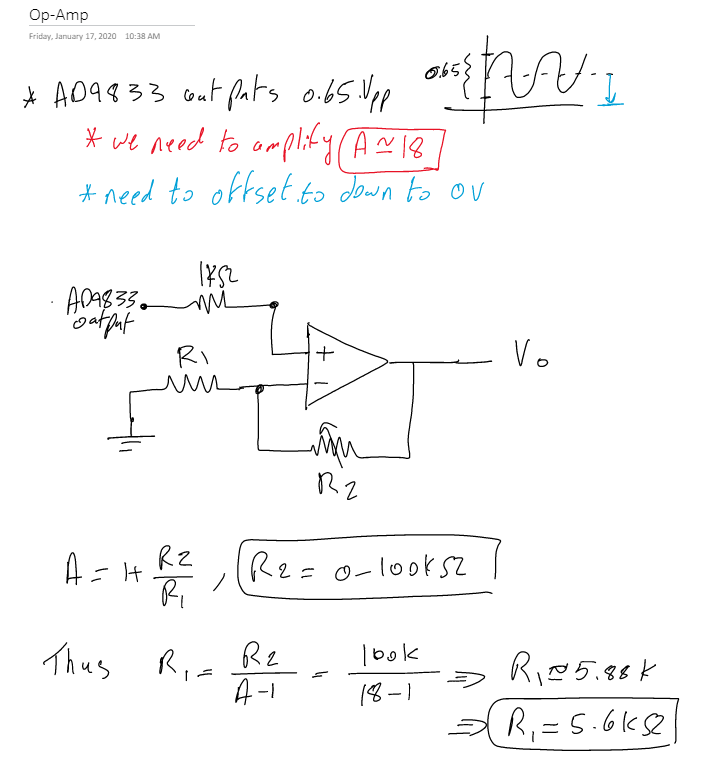


Figure 1.3: Amplifier circuit design.

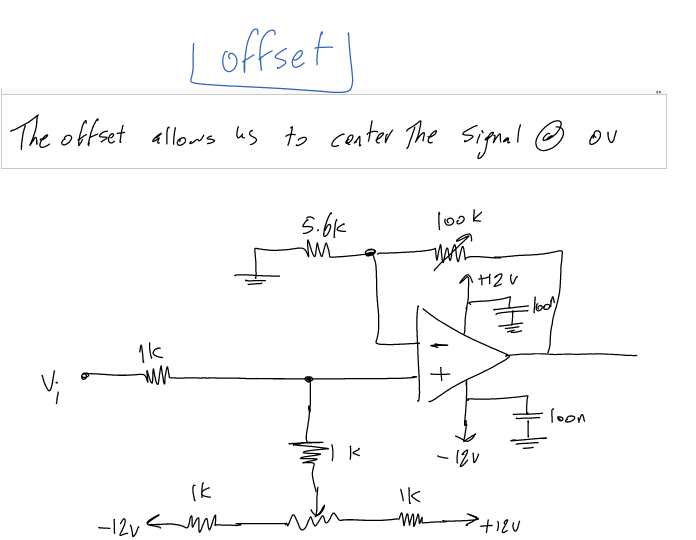


Figure 1.4: Offset Circuit.

To test the design of the amplifier circuit, it was assembled on a breadboard (Figure 1.5) and using a function generator producing a 0.65Vpp sine wave applied to the amplifier circuit. The output was tested (Figure 1.6).

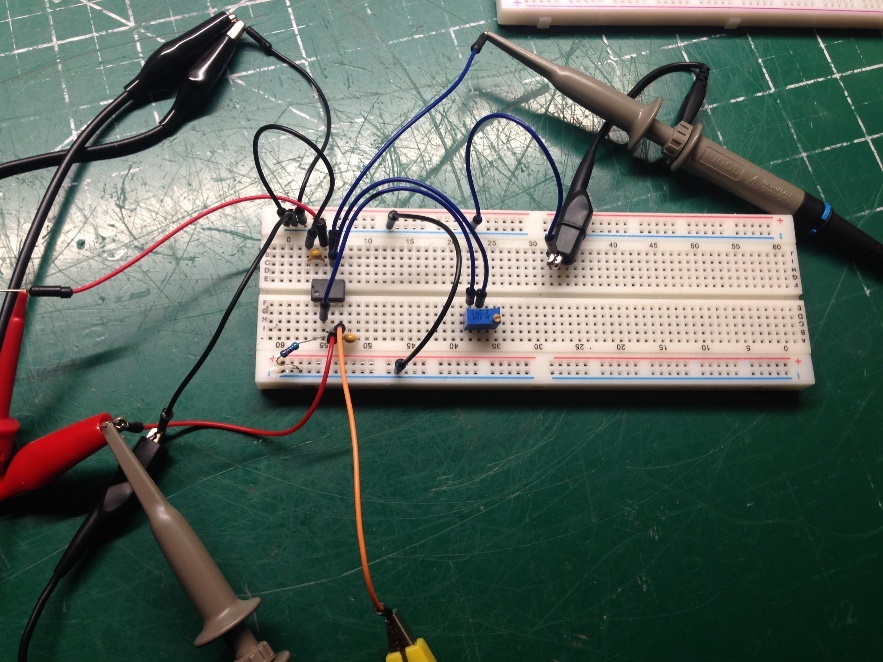


Figure 1.5: Amplifier circuit prototype.

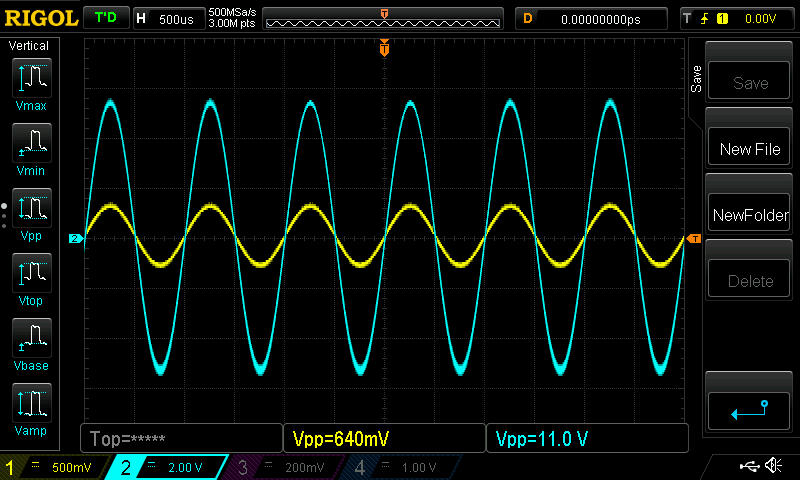


Figure 1.6: Amplifier circuit output.

## Power Supply

Though the AD9833 and ATmega328 use 5VDC to operate, this level is insufficient for the Op-Amp circuit. To operate an Op-Amp circuit we need a minimum of +12VDC and -12VDC so that a 25.1dB gain can be achieved. A simple 24V center tap transformer coupled with a full bridge rectifier and voltage regulators will provide the voltage and ample power to operate both the power and control circuits (Figure 1.7).

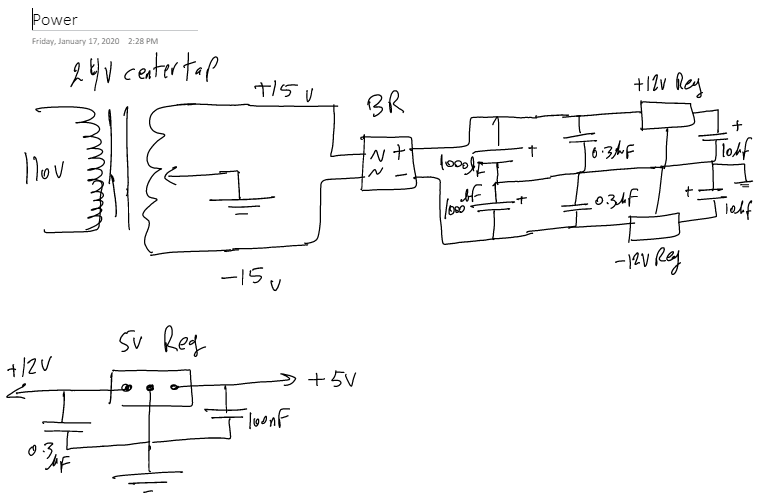


Figure 1.7: Power supply design circuit.

The circuit was then prototyped on a breadboard (Figure 1.8) and results were tested (Figure 1.9).

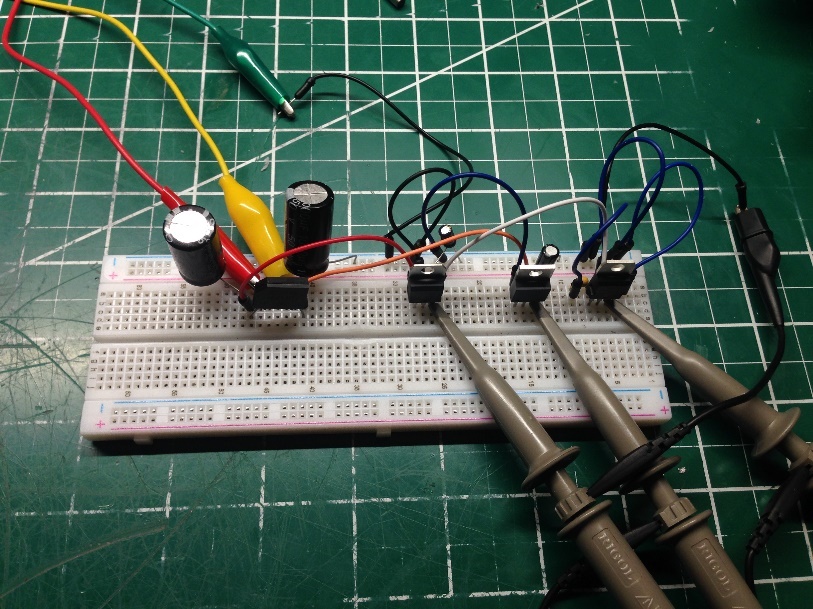


Figure 1.8: Power supply prototype circuit.

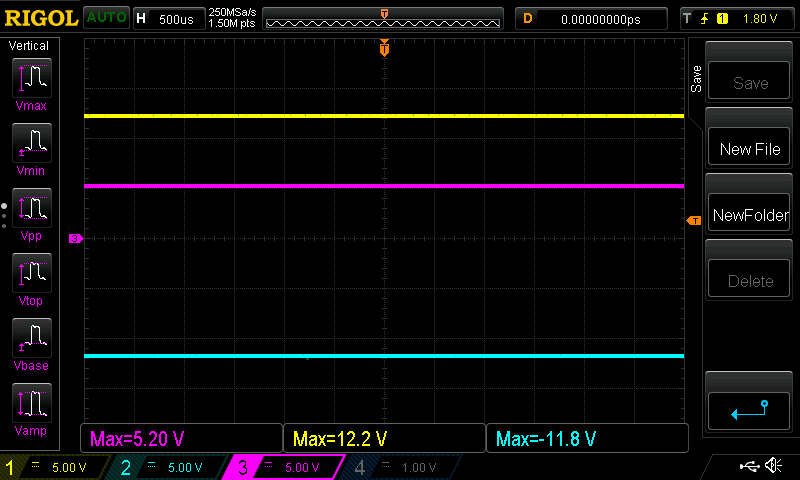


Figure 1.9: Power supply prototype test results.

## Control

The ATmega328 provides many digital IO for the user input buttons to be connected. We utilized the ATmega328 internal pull-up resistors so that the buttons are now active low. This allowed us to minimize the area needed on the PCB (less resistors). Six buttons were utilized to adjust frequency, wave type, and ON/OFF features (Figure 1.10).

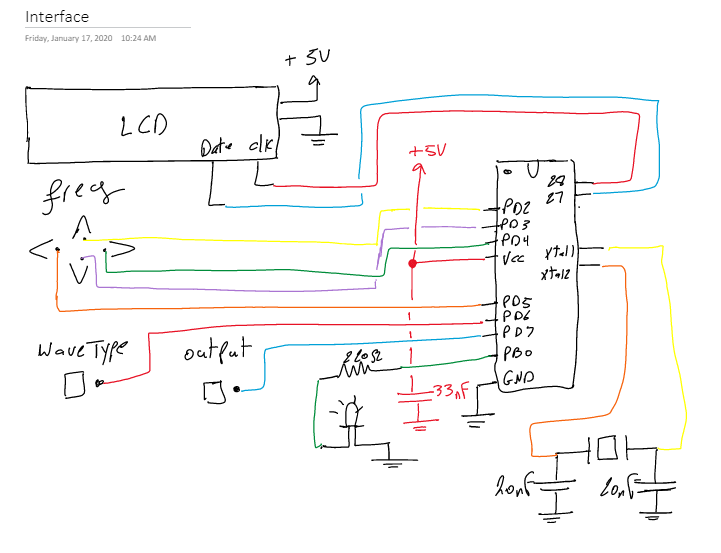


Figure 1.10: I/O utilization.

Since most commercial function generators use an output relay to turn the output ON and OFF, we incorporated such relay circuit in our design (Figure 1.11) for two reasons. One, it is a commercial requirement, and two, it allowed us to use the knowledge acquired from 425 class to run an inductive load using a microprocessor IO and a transistor circuit.



Figure 1.11: Output relay circuit design.

## PCB Design

After prototyping and testing the entire design for proper functionality, schematics and PCB were designed and implemented using Altium Designer (appendix B & C). The design is implemented on three PCBs, Main PCB, power supply PCB, and front panel PCB. The PCBs are connected together using custom ribbon cables. The multi PCB design separates the higher power AC PCB from the lower power DC control components located on the main PCB. The PCBs are designed as follows:

* The main PCB incorporates the AD9833, ATmega328 microcontroller, LM318N Op-Amp, Relays, NPN transistors, Oscillators, and various passive components.
* Power supply PCB which incorporates the full bridge rectifier, voltage regulators, and various filtering capacitors.
* Front panel PCB which mounts the LCD display, input buttons, potentiometers, and output BNC jack.

The PCBs were designed such that the bottom layer is a ground pour (plane) where all the ground pins are connected to. Also, the main PCB has two ground pours, AC and DC ground pours. This was recommended by the AD9833 datasheet to reduce interference.

During PCB design of main board, the footprint of the 25MHz oscillator required for the AD9833 chip was designed backwards, this error occurred due to the footprint layout in the datasheet being displayed from the bottom. Luckily, placing the oscillator on the back side of PCB and soldering it from the front corrected the issue.

See Appendix B for schematics and Appendix C for PCBs.

## Enclosure

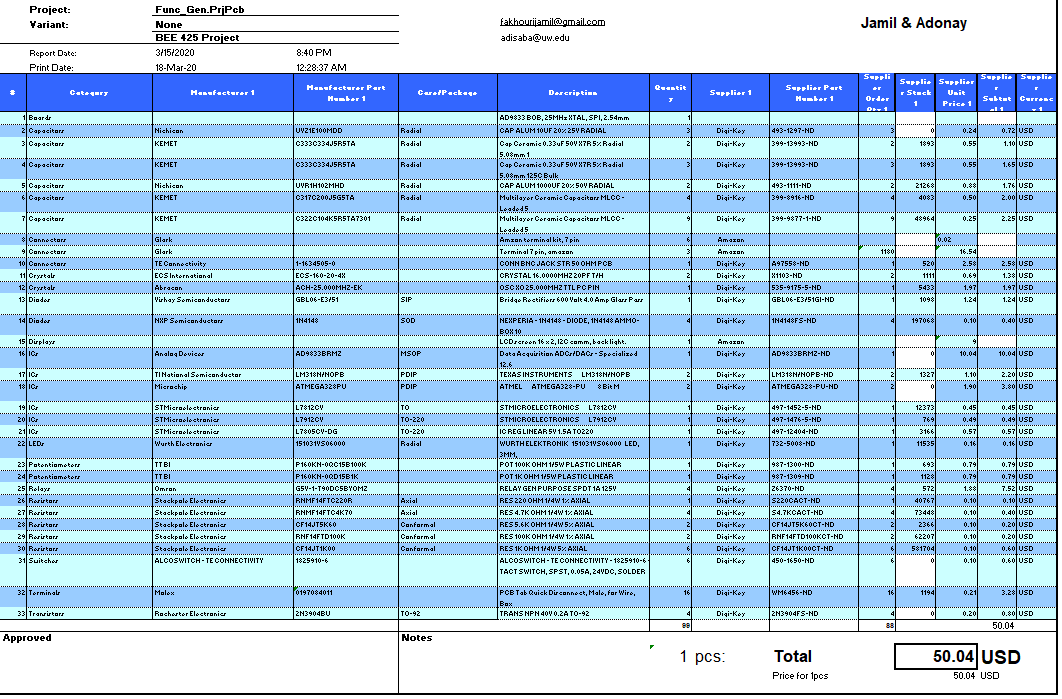
After the PCBs were designed, manufactured and tested, 3D models of each PCB were generated in Altium Designer, and the enclosure was modeled in Autodesk Inventor. Inventor allows for parts to be assembled virtually so that fitment and accuracy can be guaranteed to a high degree. This also ensured proper hole placement. The enclosure was then 3D printed using PLA material. See Appendix D

# Test Plan

Following PCB manufacturing and shipping, the following tests were conducted:

* Tested short between power and ground on each PCB.
* Continuity test of each trace.
* Continuity test after soldering to ensure no short on SMD components.
* Tested transformer output voltage (15, 0, 15 VAC).
* Tested power supply output (+12, -12, +5VDC).
* Continuity test on ribbon cables.
* Power up and tested current draw (<57mA AC).
* Tested UI for proper operation on ON/OFF, wave and frequency selection.
* Tested output on oscilloscope for frequency and amplitude ranges. (Appendix E)
* Tested proper fitment in enclosure.
* Retested operation after assembly in enclosure.

# Bill of Materials

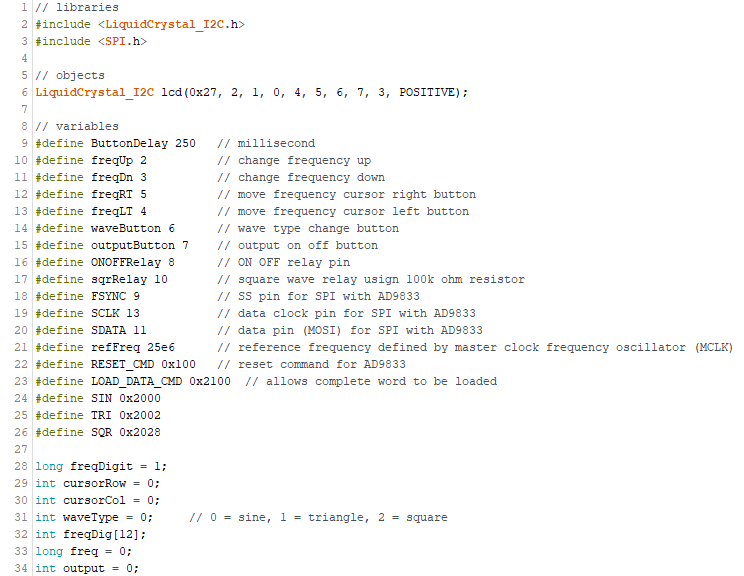


# References

* Julian Ilett, <https://www.youtube.com/watch?v=0uFb18z--lI>
* Bill Williams, <https://github.com/Billwilliams1952/AD9833-Library-Arduino>
* Cezar Chirila

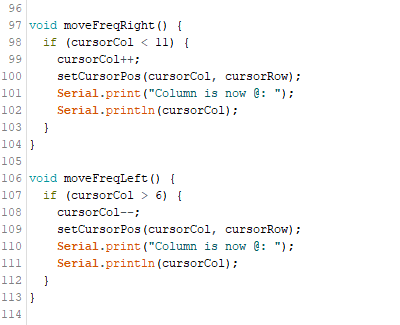
# Appendix A

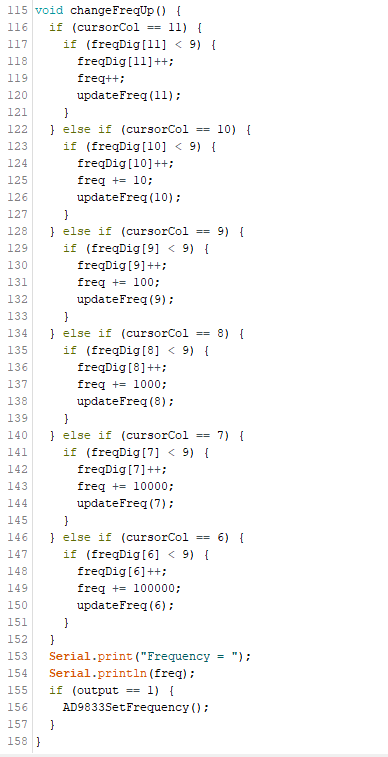
## Code

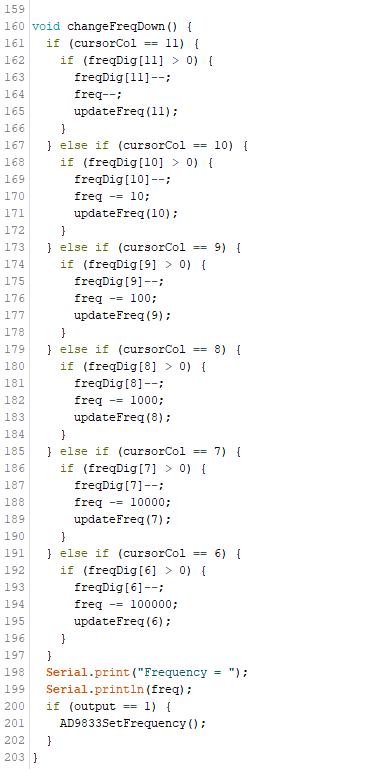


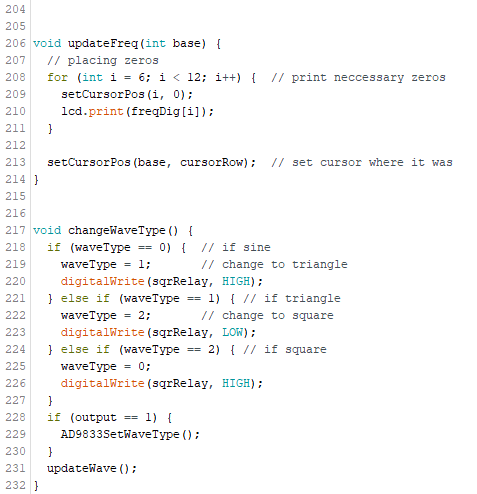




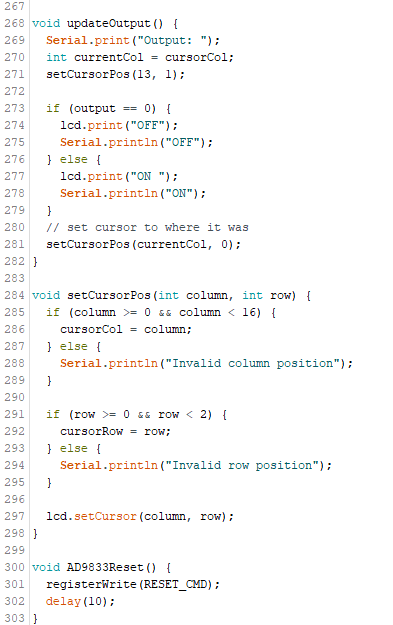










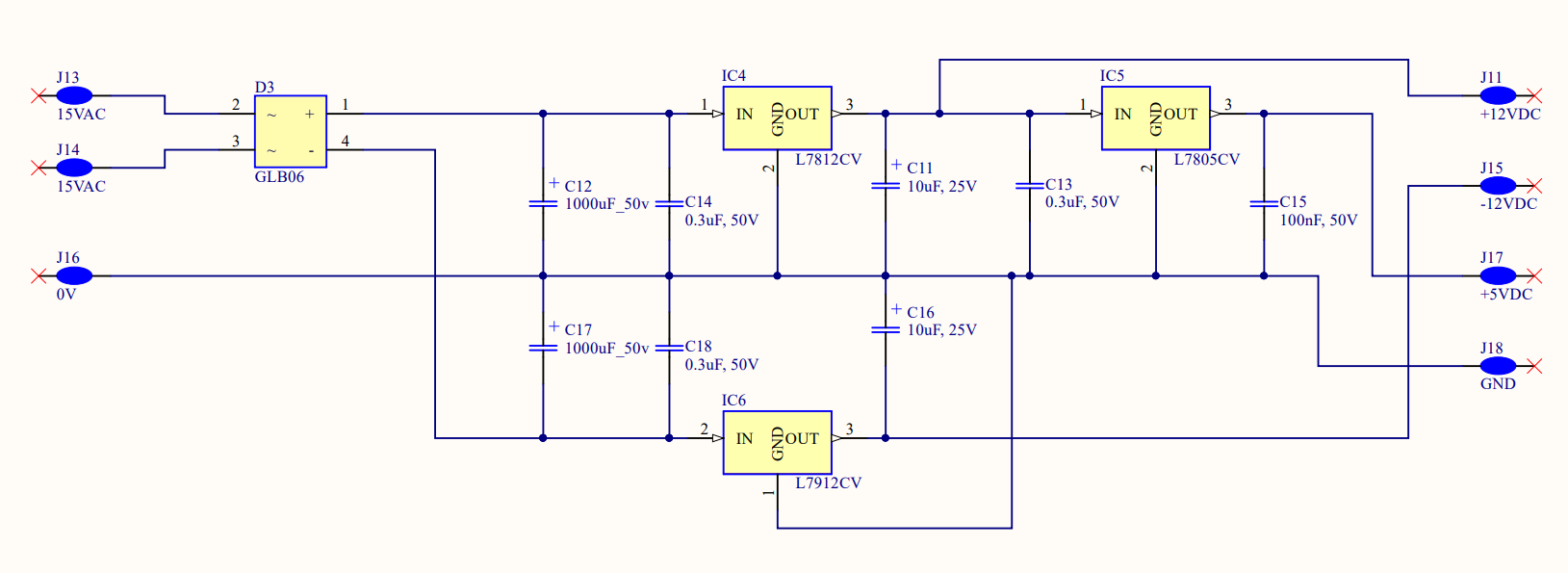




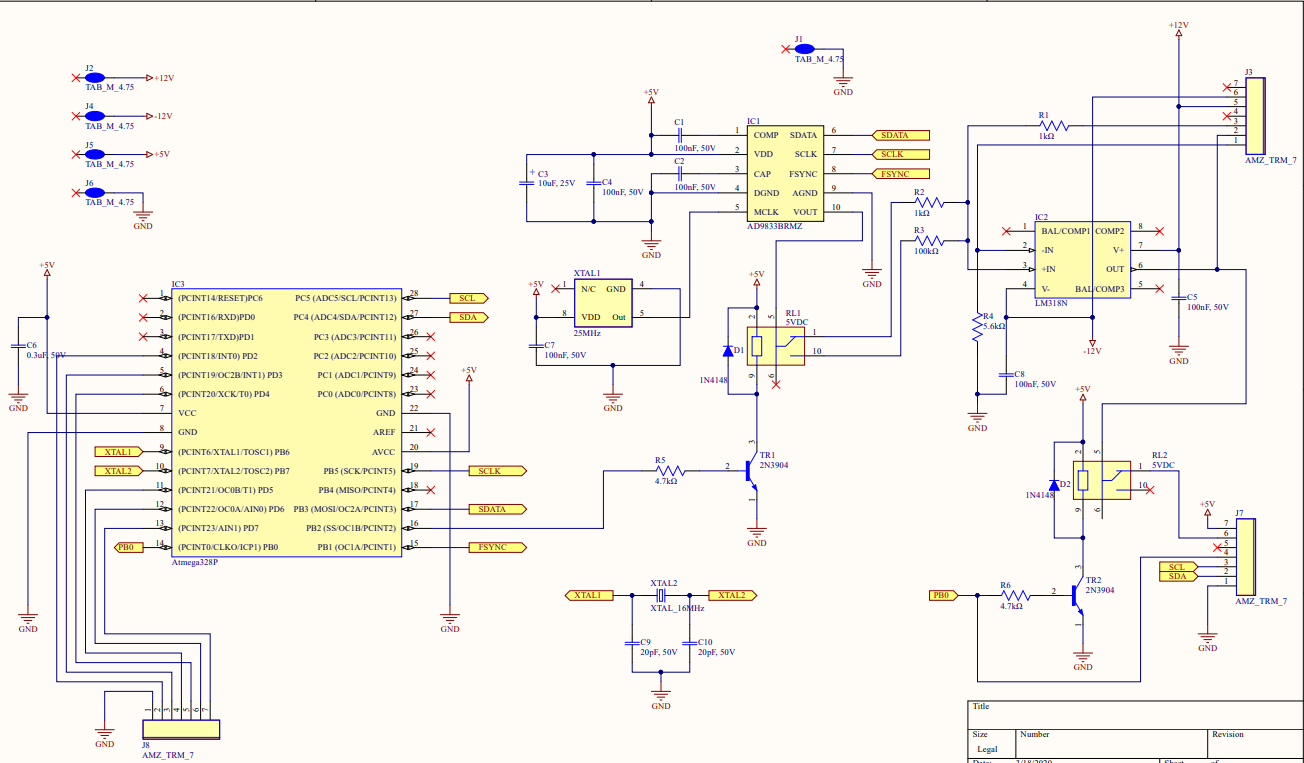
# Appendix B

## Schematics

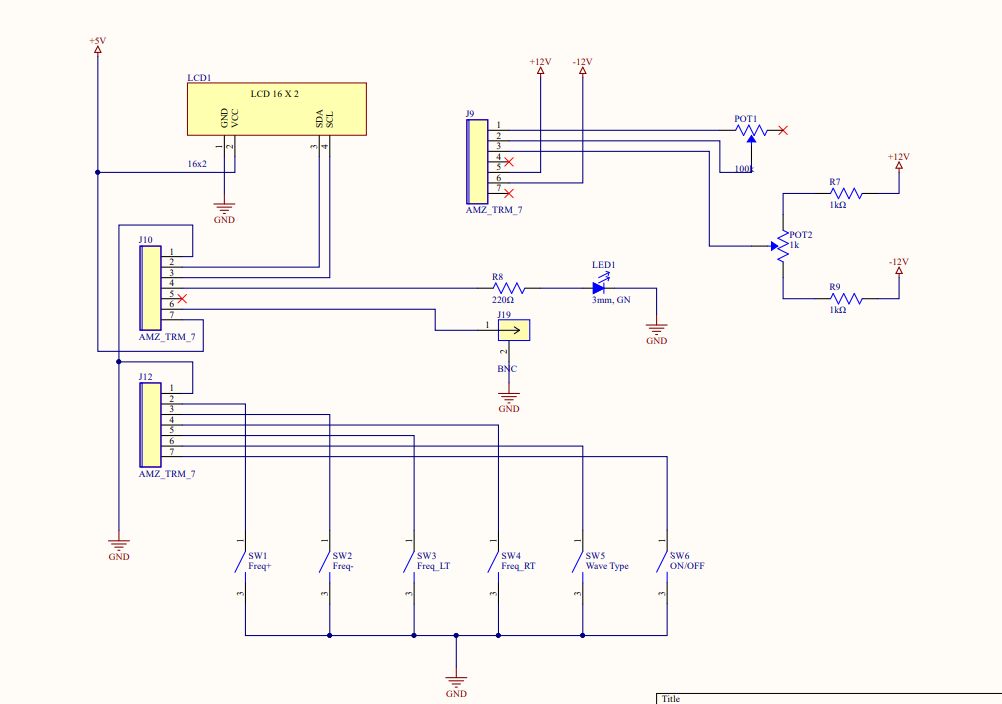
Power Supply Board



Main Board

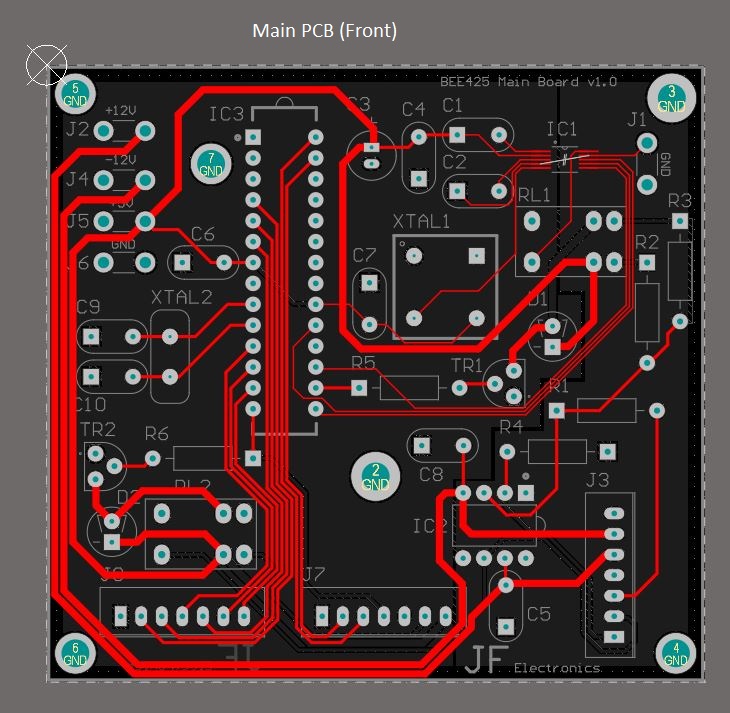


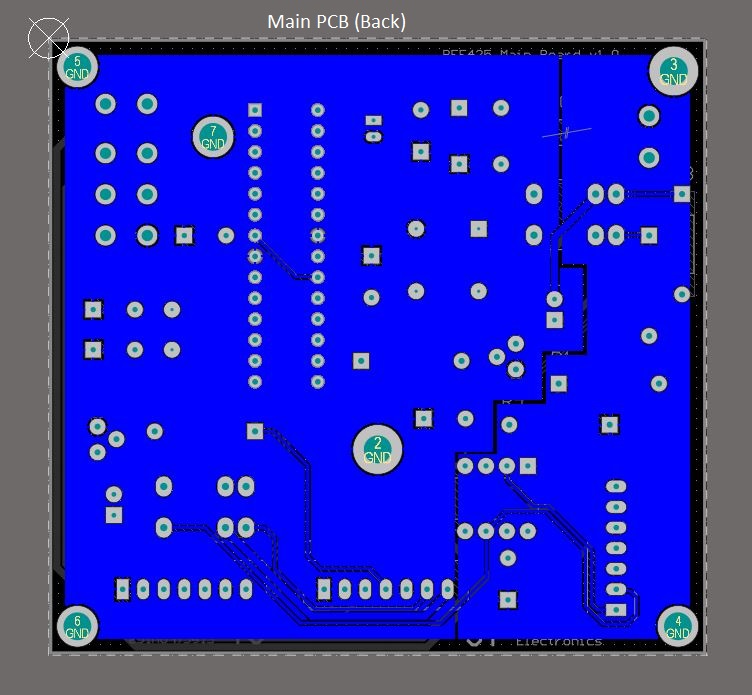
Front Panel Board

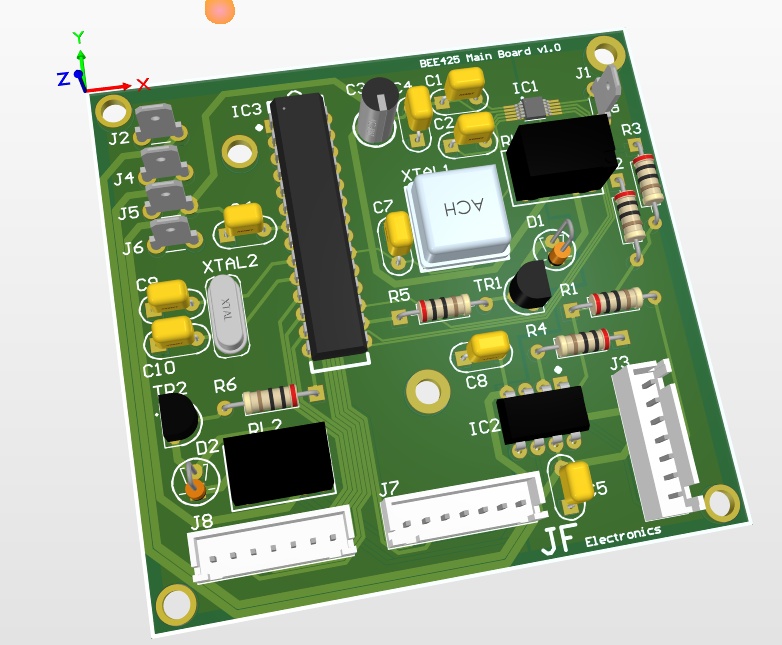


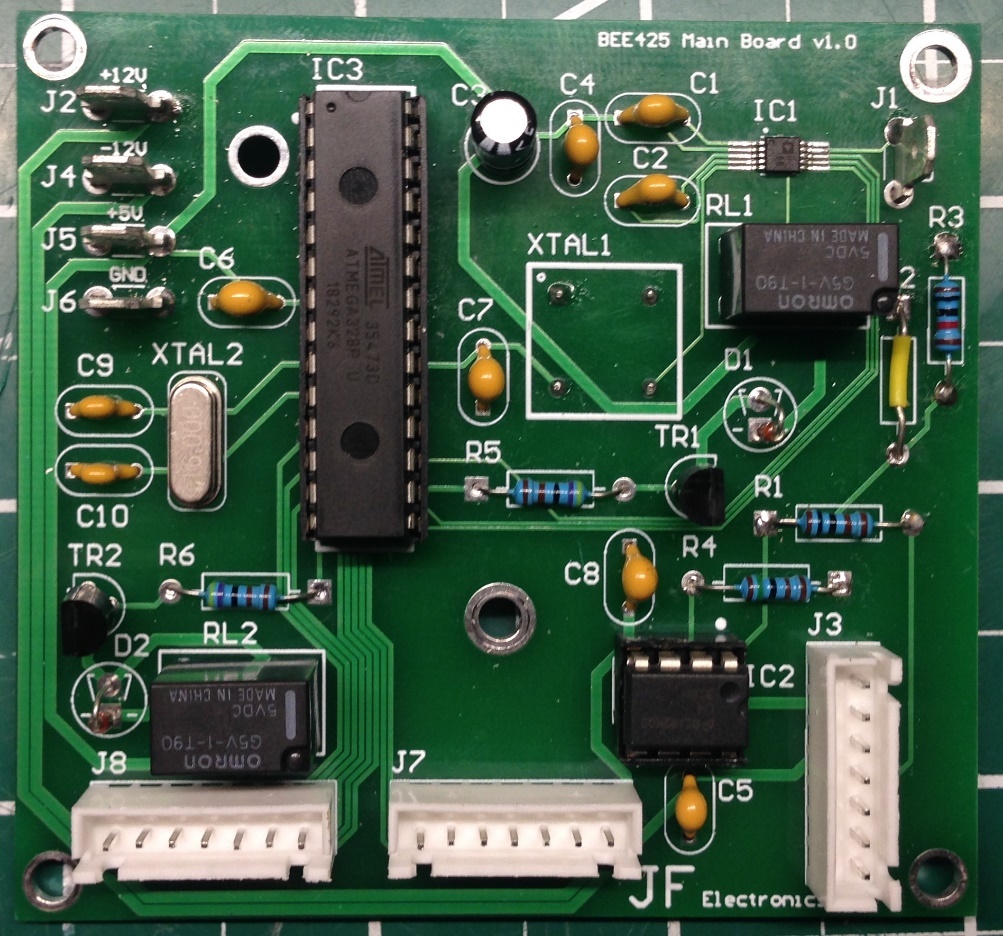
# Appendix C

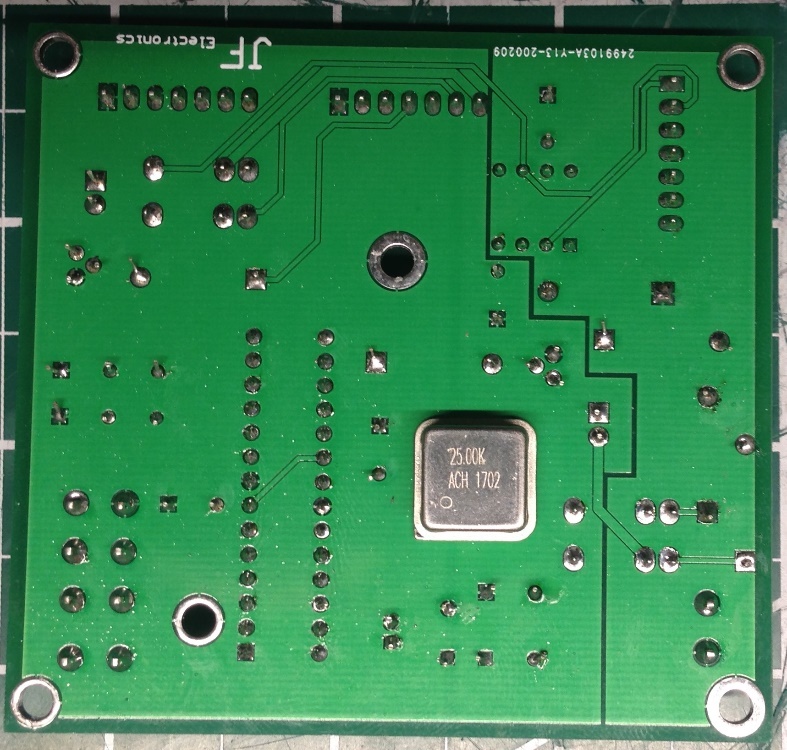
## Main PCB



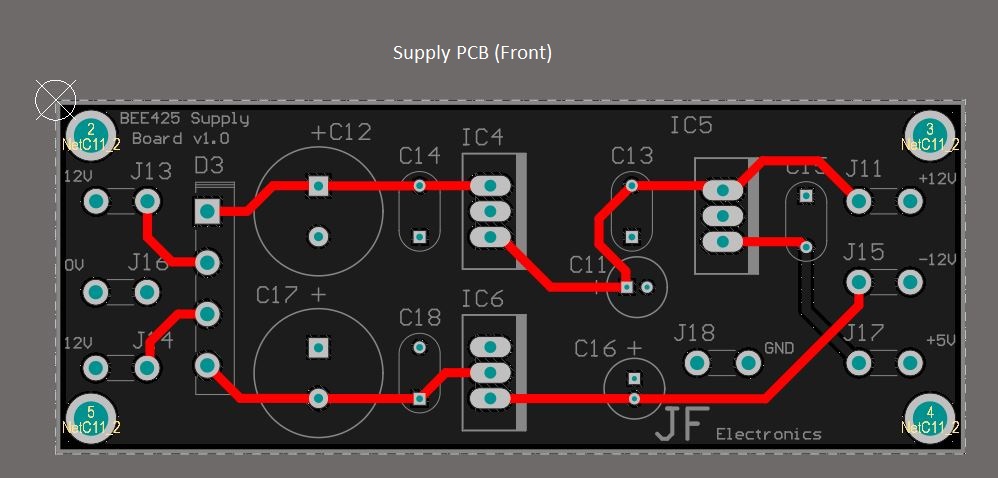


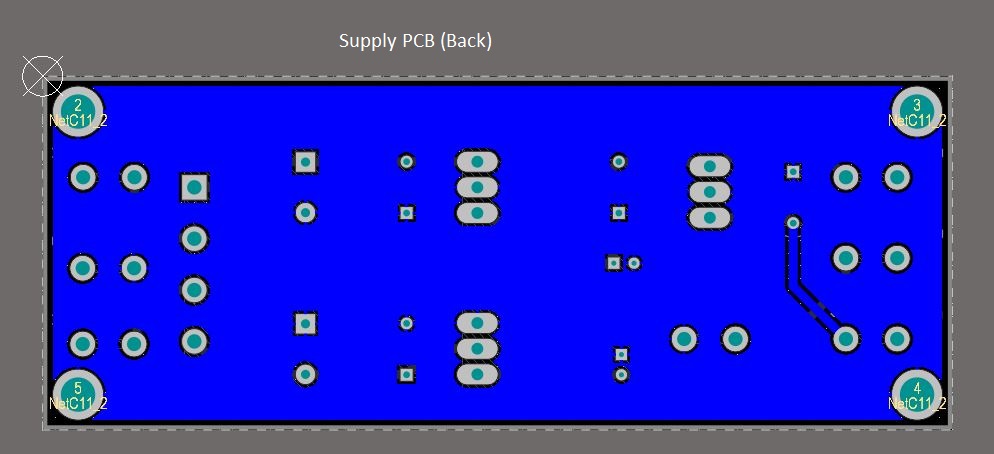


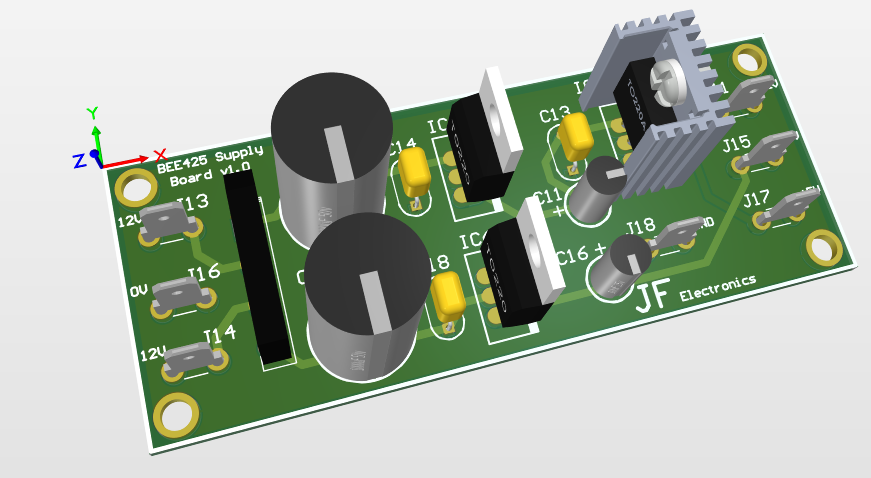


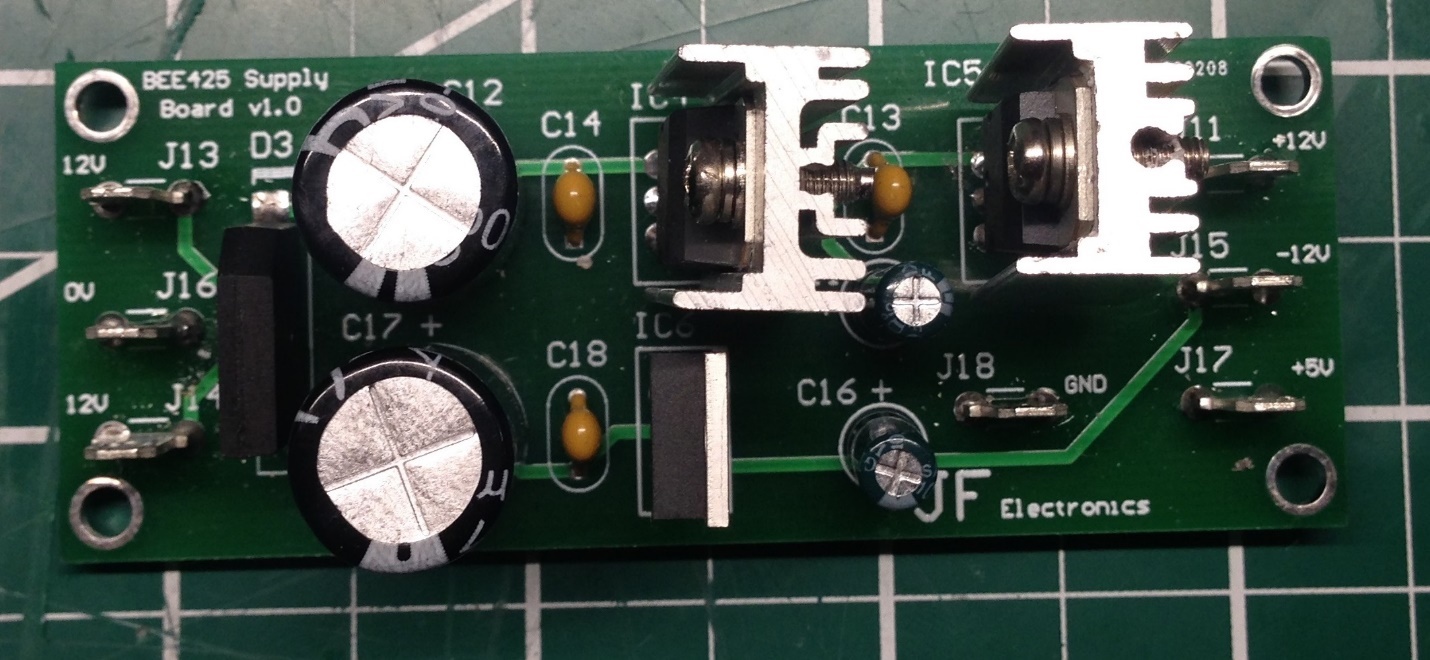


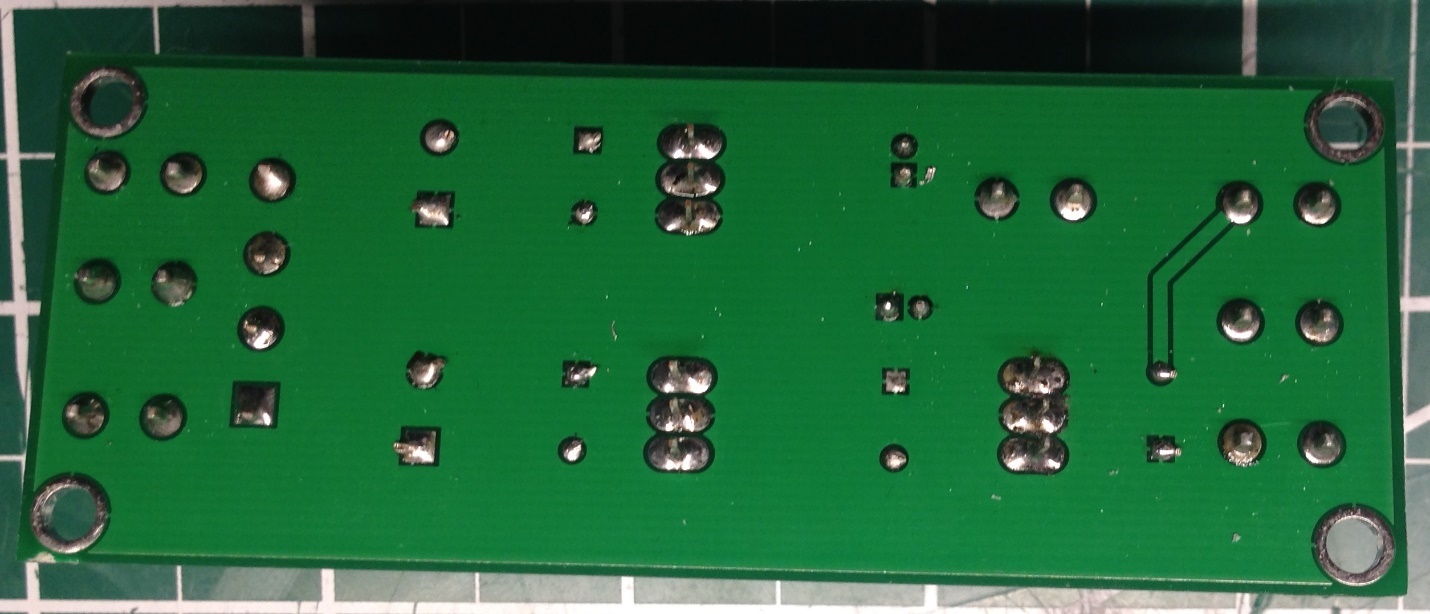
## Power Supply PCB



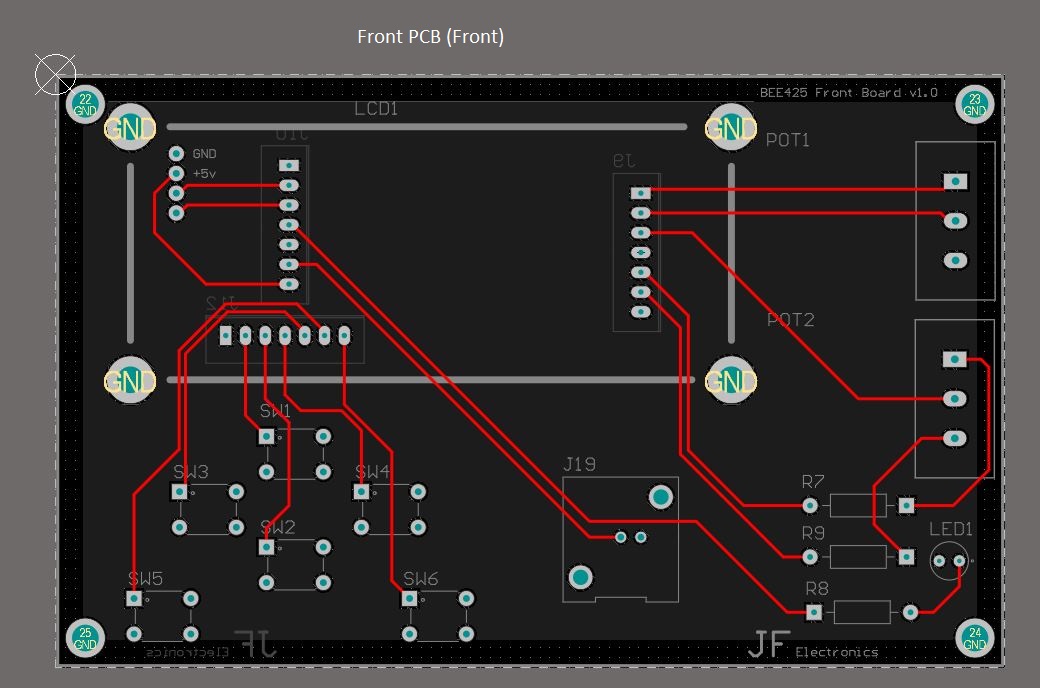


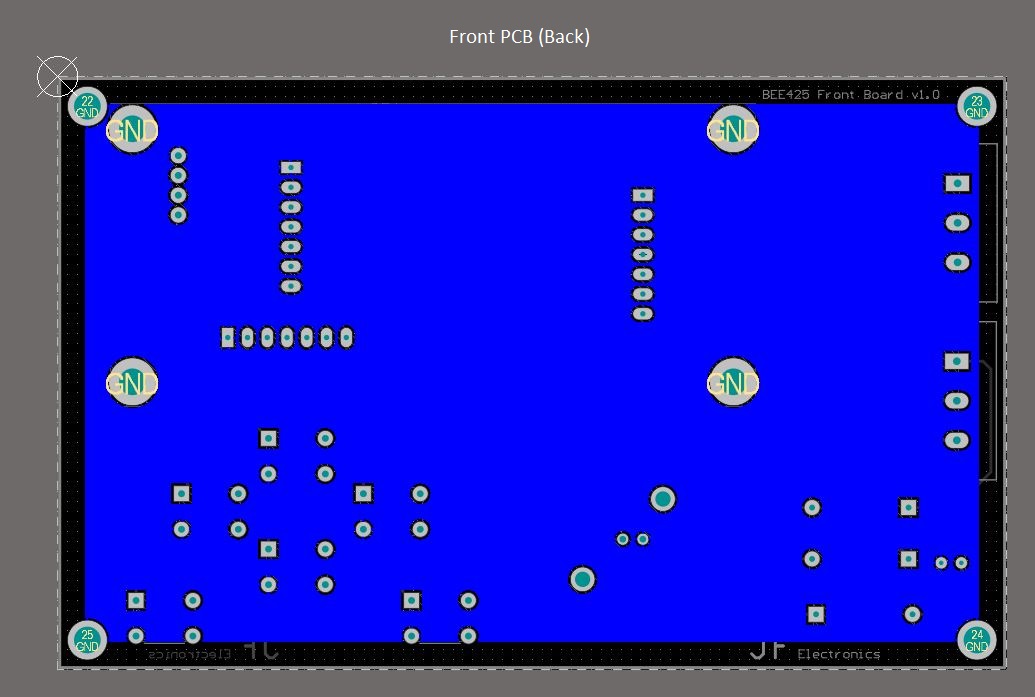


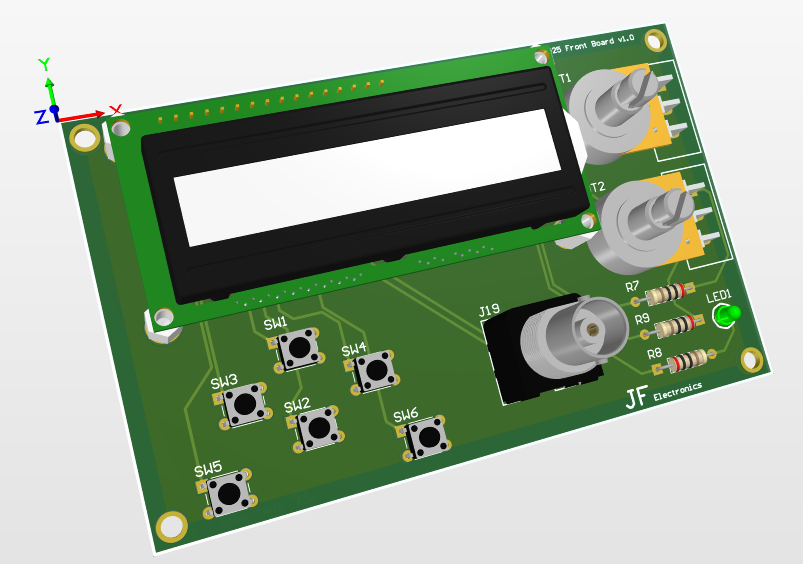


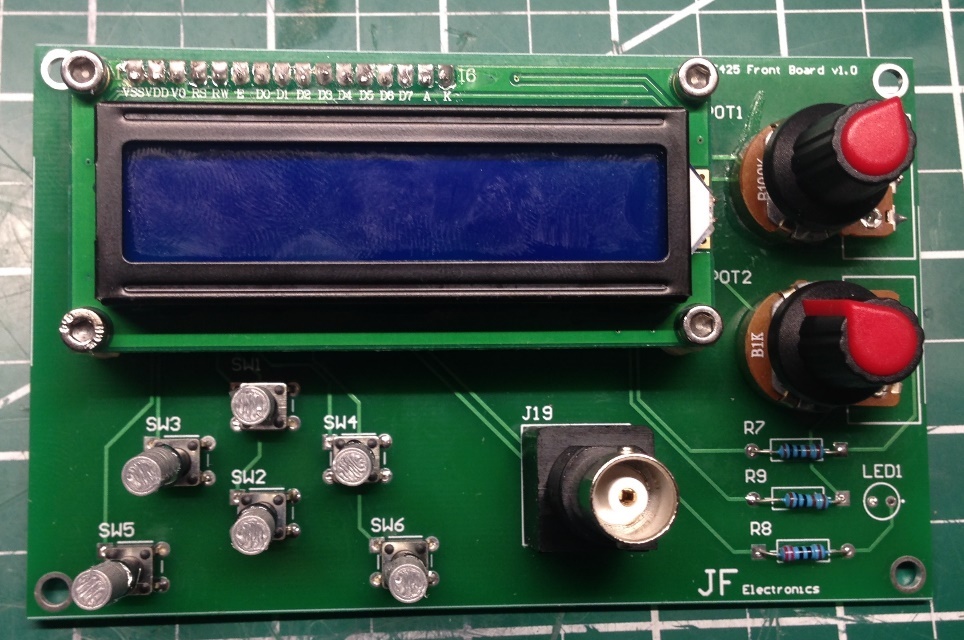


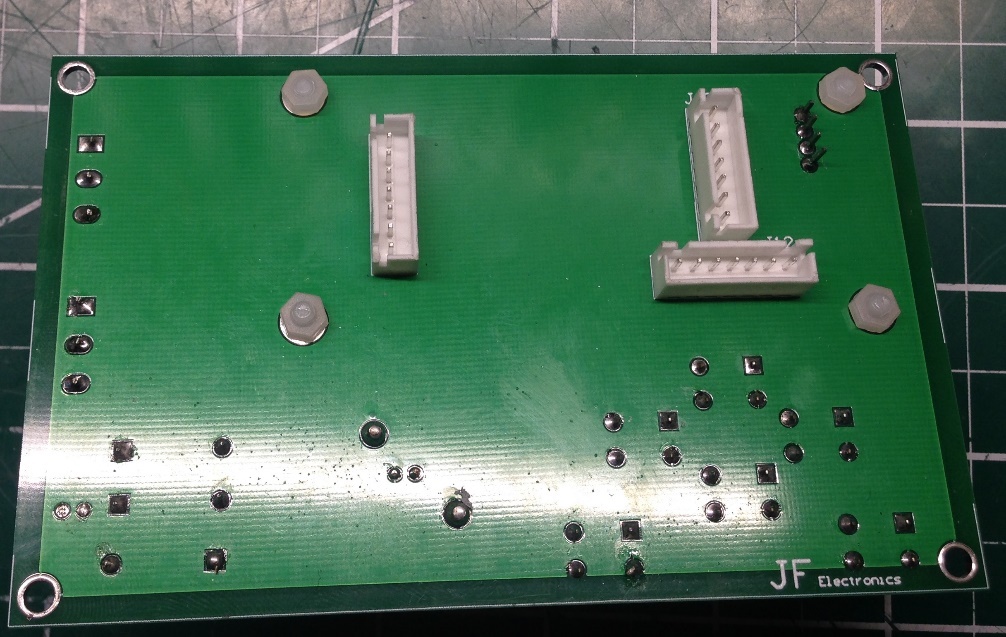
## Front PCB







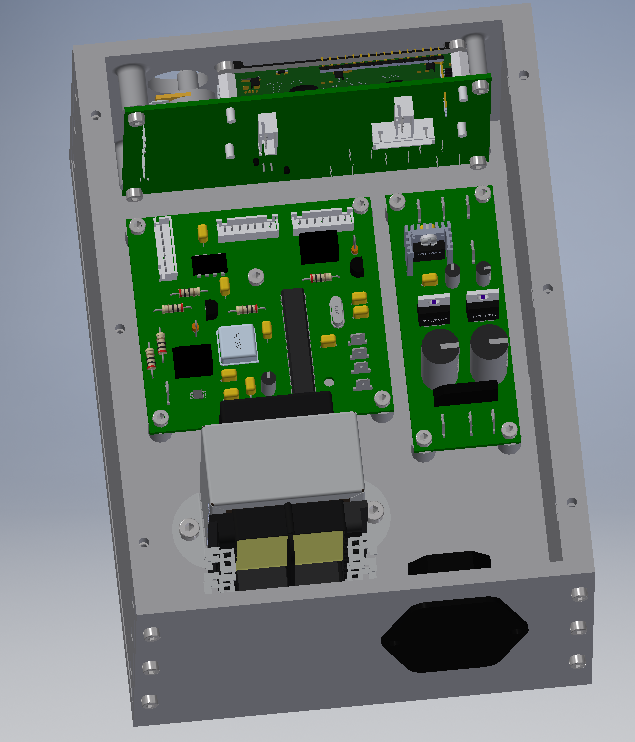


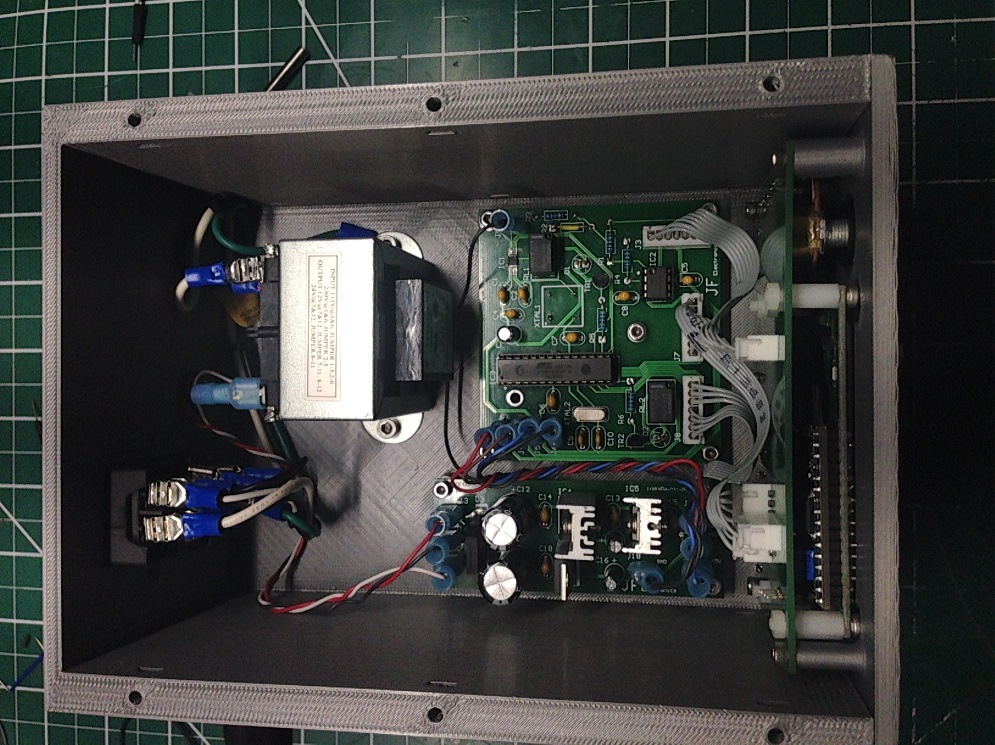


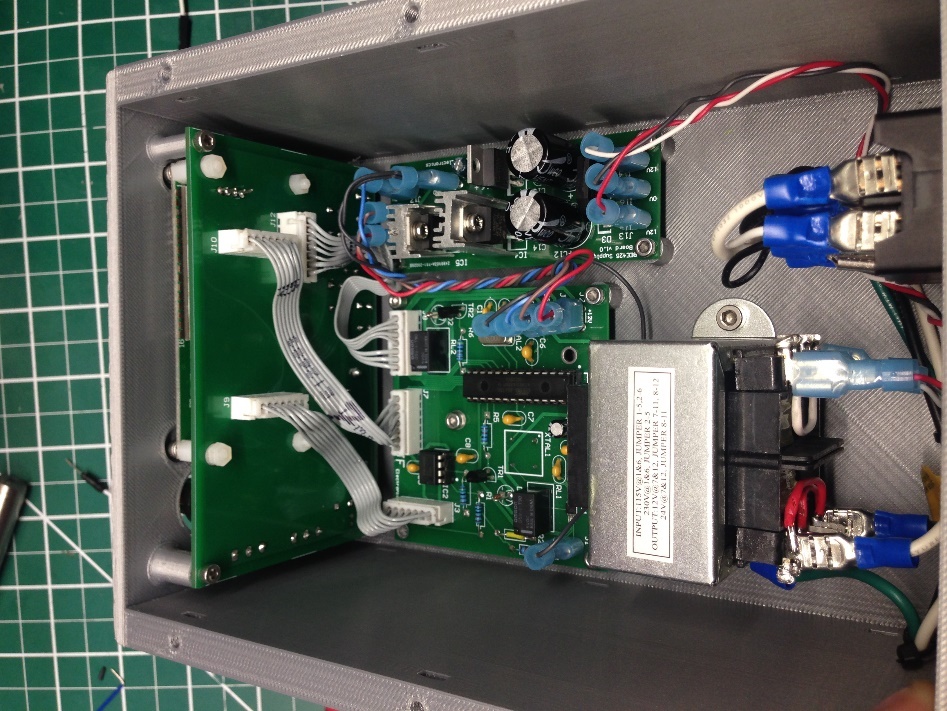
# Appendix D

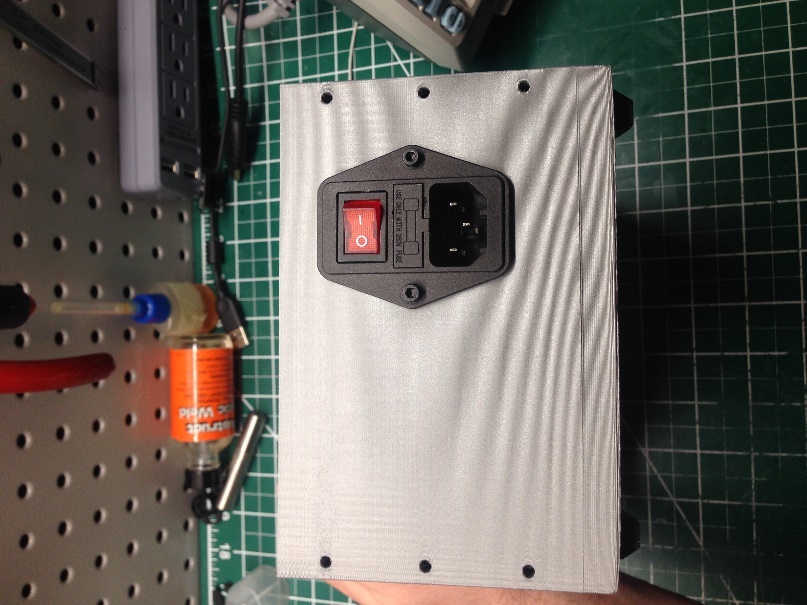
## Enclosure

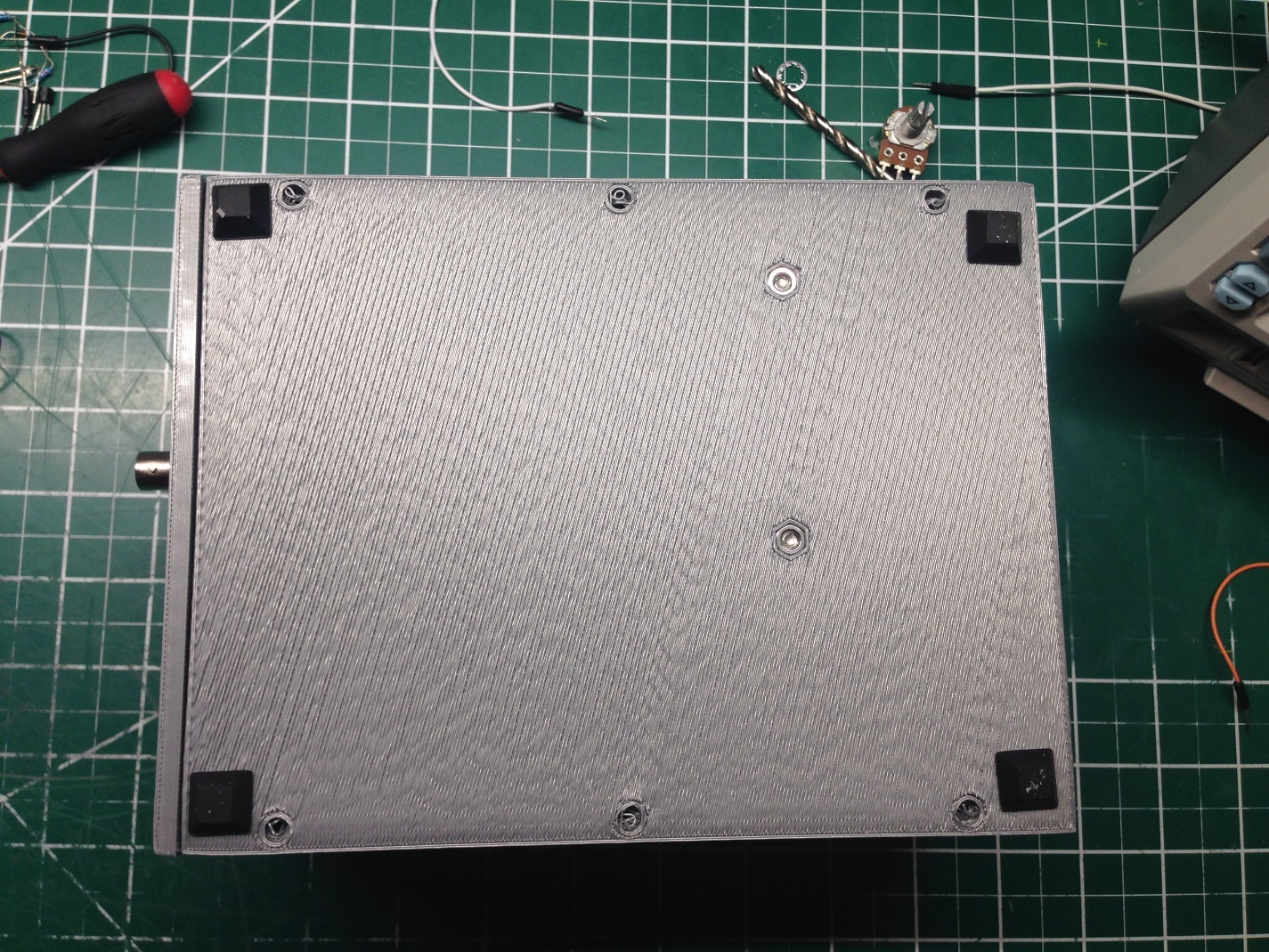












# Appendix E

## Operation Test Results

