# ECE 445 Final Presentation

Group 18: Breanne Warner, Michael Jamrozy, Yash Bhushappagala

December 6 2021

# Introduction

- Goal was to create a device that would be able to take input from MIDI keyboard and playback synth sounds

- Goal was to make it relatively affordable. Even basic analog synthesizers can cost a lot of money, so we wanted to limit the cost to about $100 so it is more accessible.

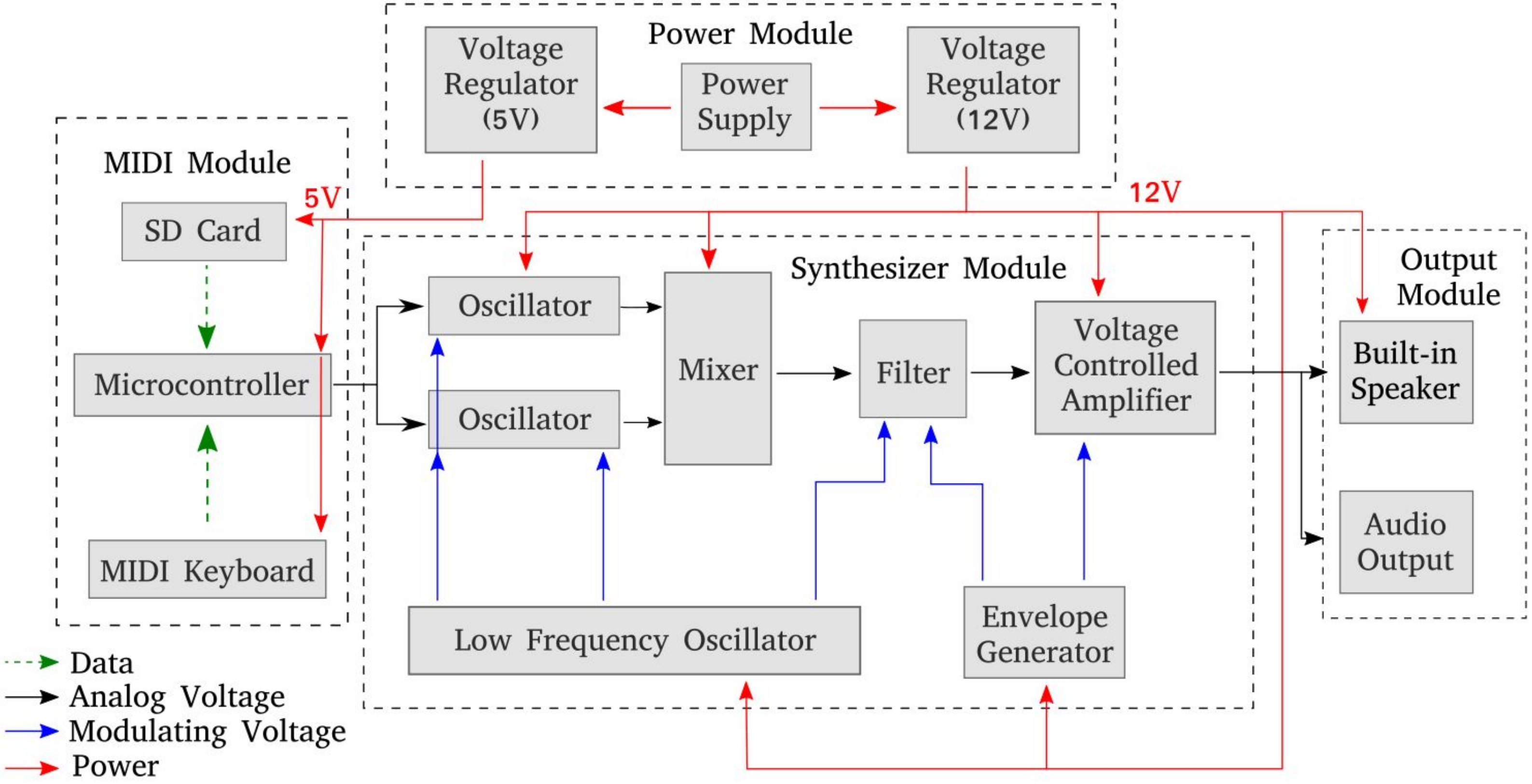
## High Level Requirements:

1. Ability to produce a square wave and sawtooth wave. These basic waveforms can be modulated by a low-frequency oscillator, for example the amplitude can be modulated (tremolo) or the frequency (vibrato). They go through a resonant low pass filter and are shaped by an envelope generator.

2. It can produce the correct pitches for at least 24 consecutive keys on an external MIDI keyboard.

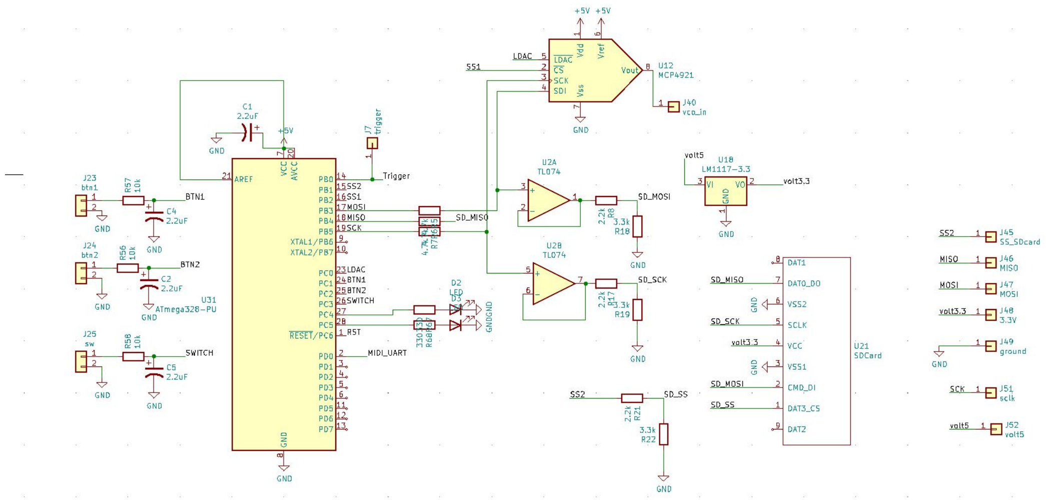
3. Synthesizer can also play back notes which are stored in files on an SD card.

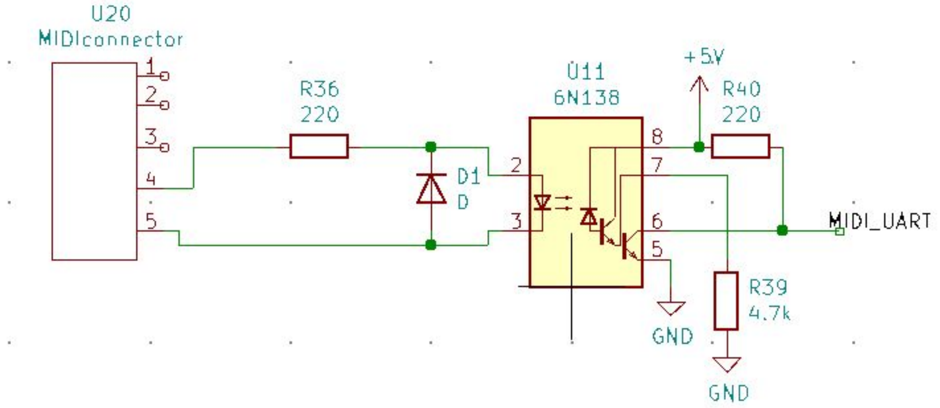
# Block Diagram



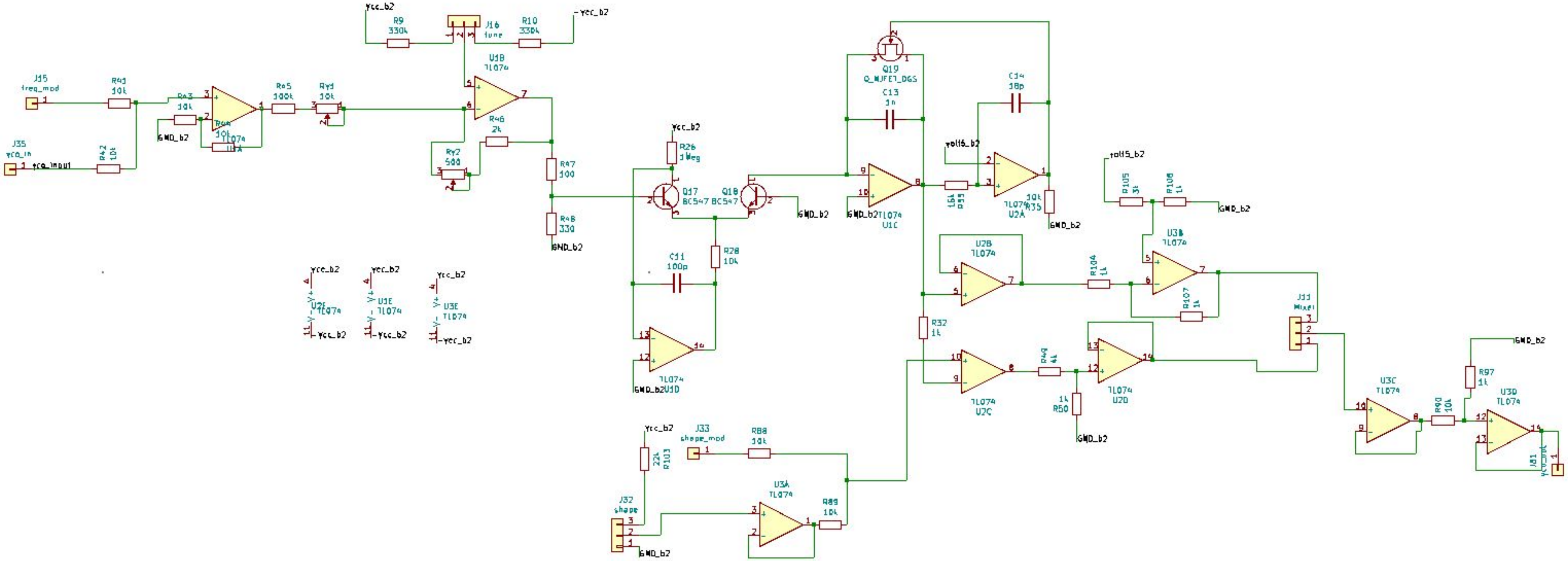

Block Diagram Design

# Design

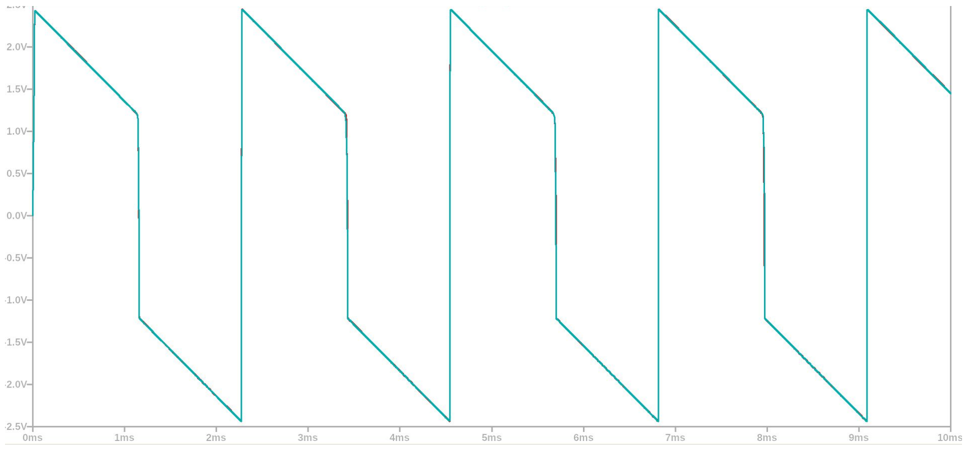


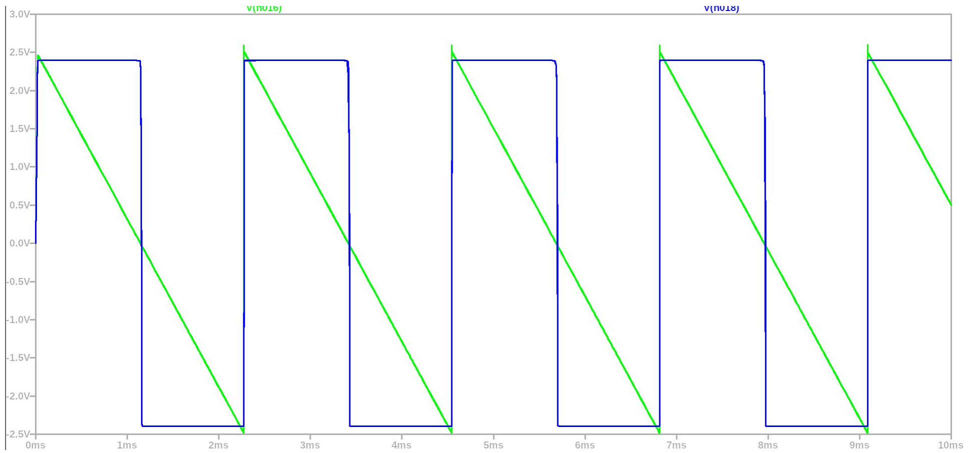




| Requirements | Verification |
|---|---|
| Produce voltage from 0-5V. | Probe output of the DAC |
| Produces a voltage corresponding to MIDI input | Probe the DAC as notes are played |
| Read key events from an SD card | Probe the DAC as notes are played |

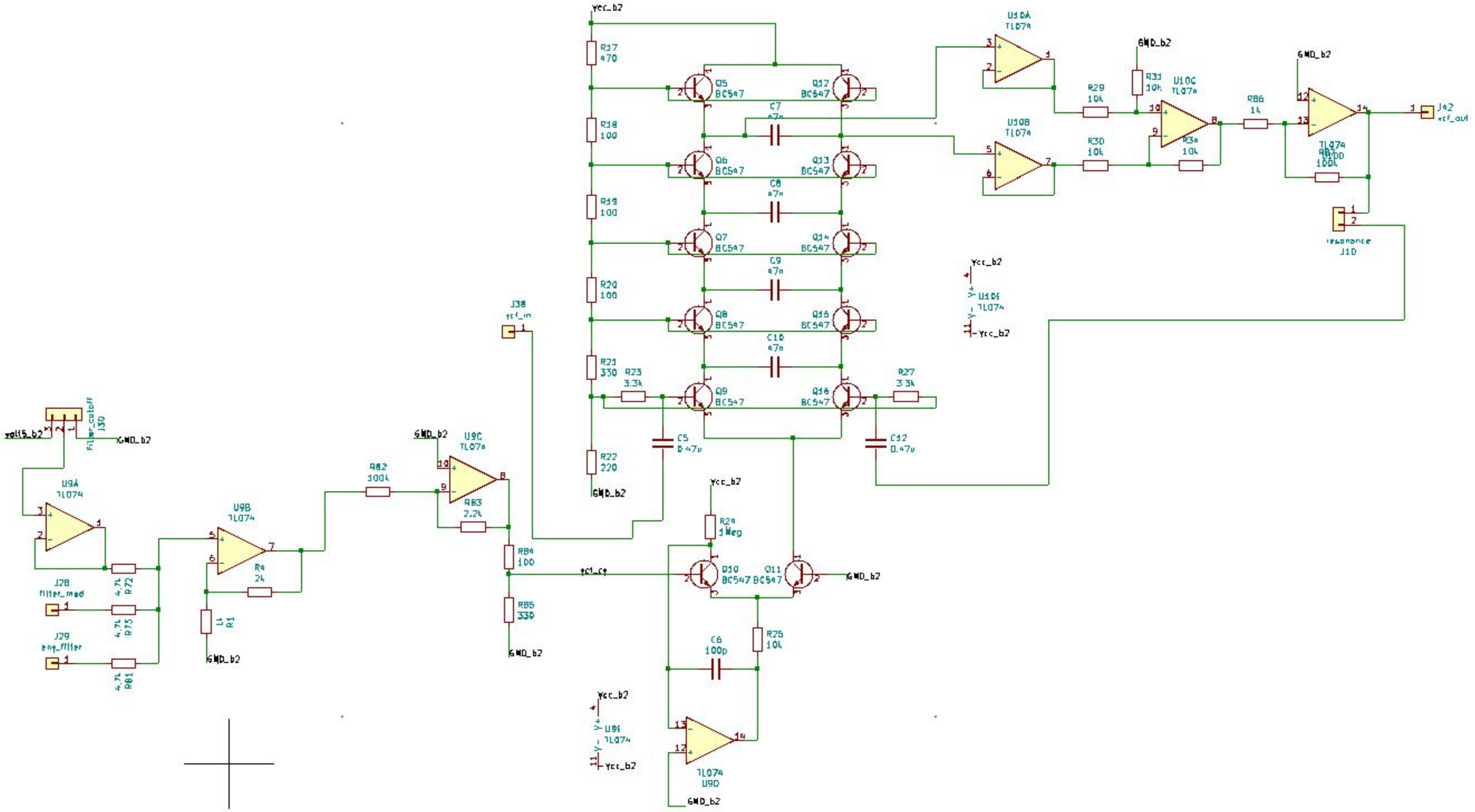



# Design: Voltage Controlled Oscillator



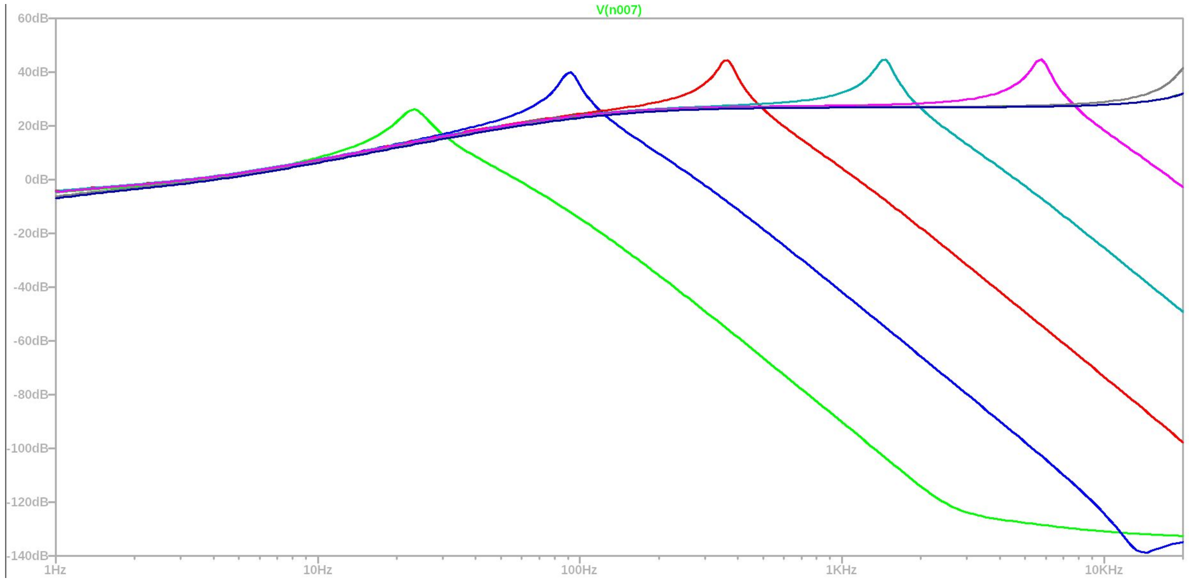




| Requirements | Verification |
| --- | --- |
| Produce square and sawtooth wave with frequency corresponding to voltage input. | Check the frequency with an oscilloscope. |

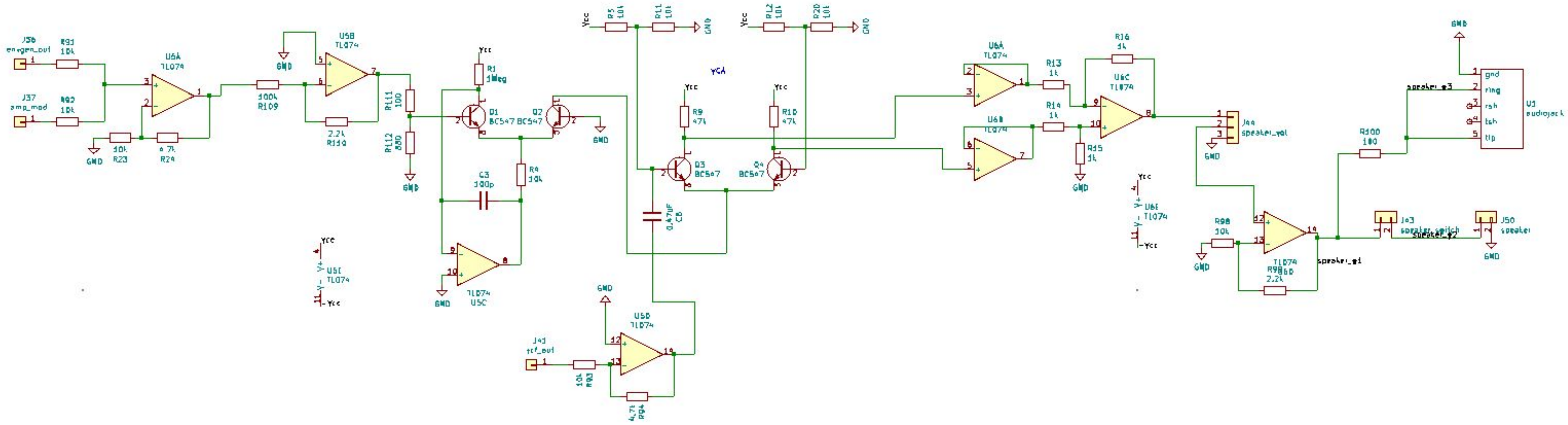


# Design: Voltage Controlled Filter

# Design: Voltage Controlled Filter

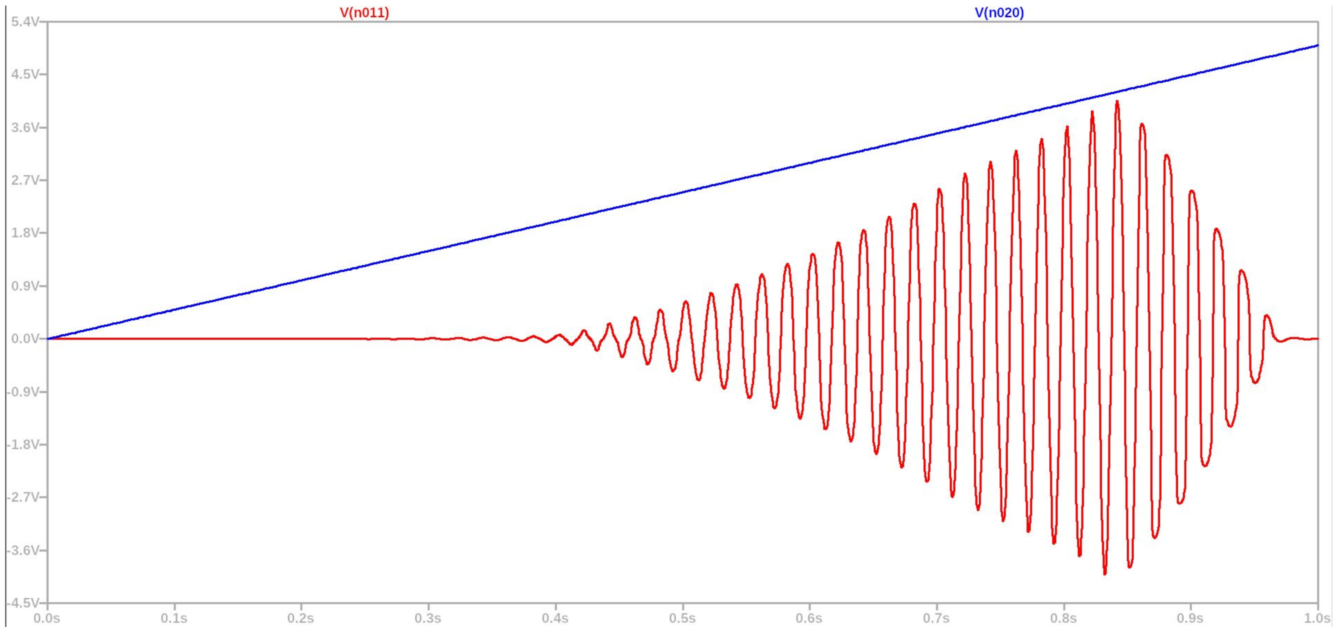



| Requirements | Verification |
|---|---|
| Low pass filter with controllable cutoff and resonance | Check oscilloscope with square wave input, vary resonance and cutoff. |

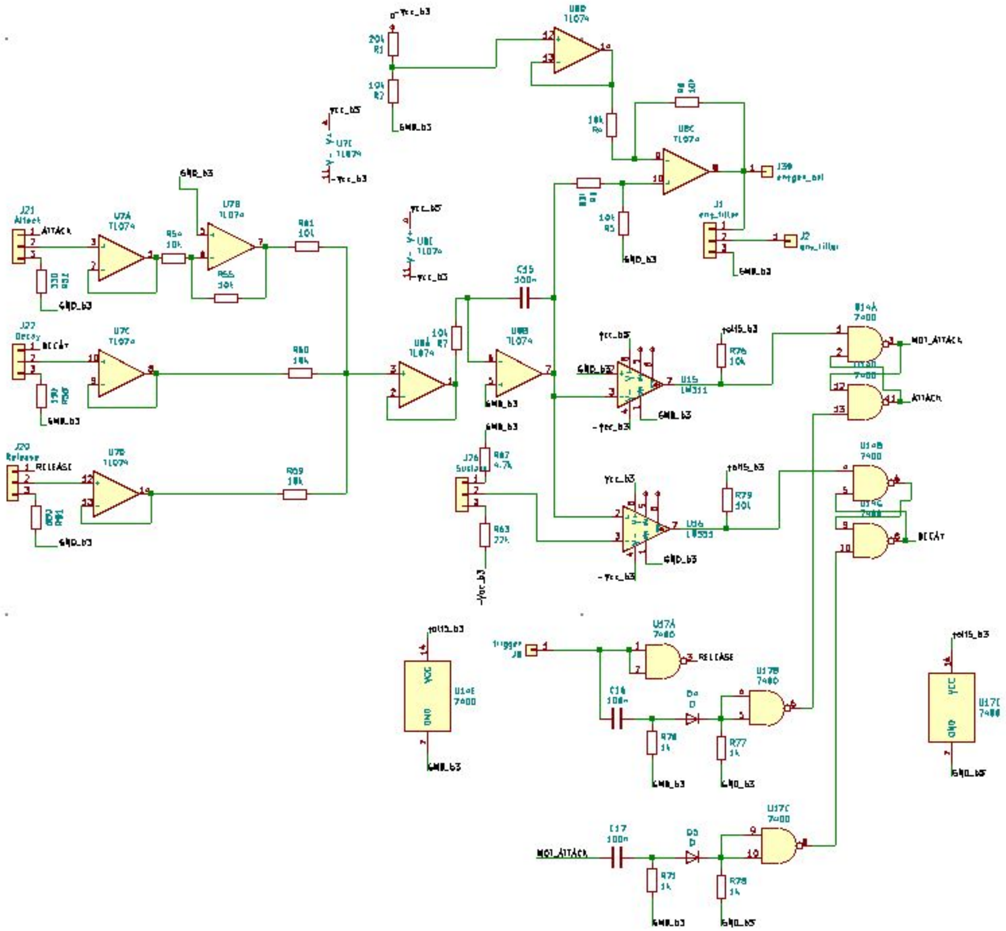


# Design: Voltage Controlled Amplifier

# Design: Voltage Controlled Amplifier

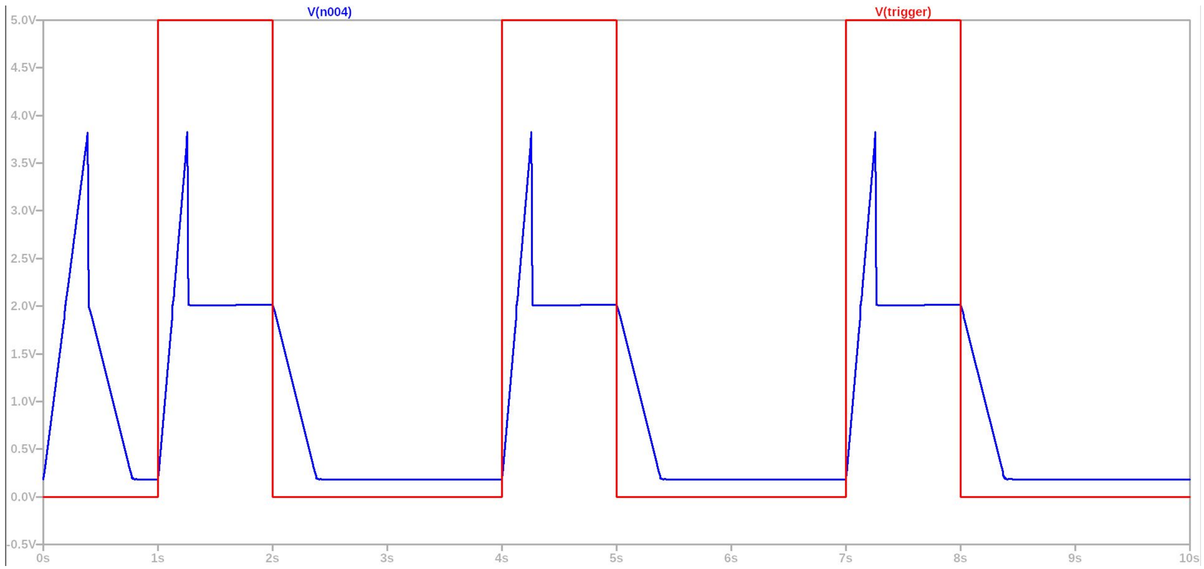



| Requirements | Verification |
|---|---|
| Input audio scaled in scaled by the control voltage, which comes from the envelope and LFO. | View the output with an oscilloscope. Slowly change control voltage and verify amplitude changes |

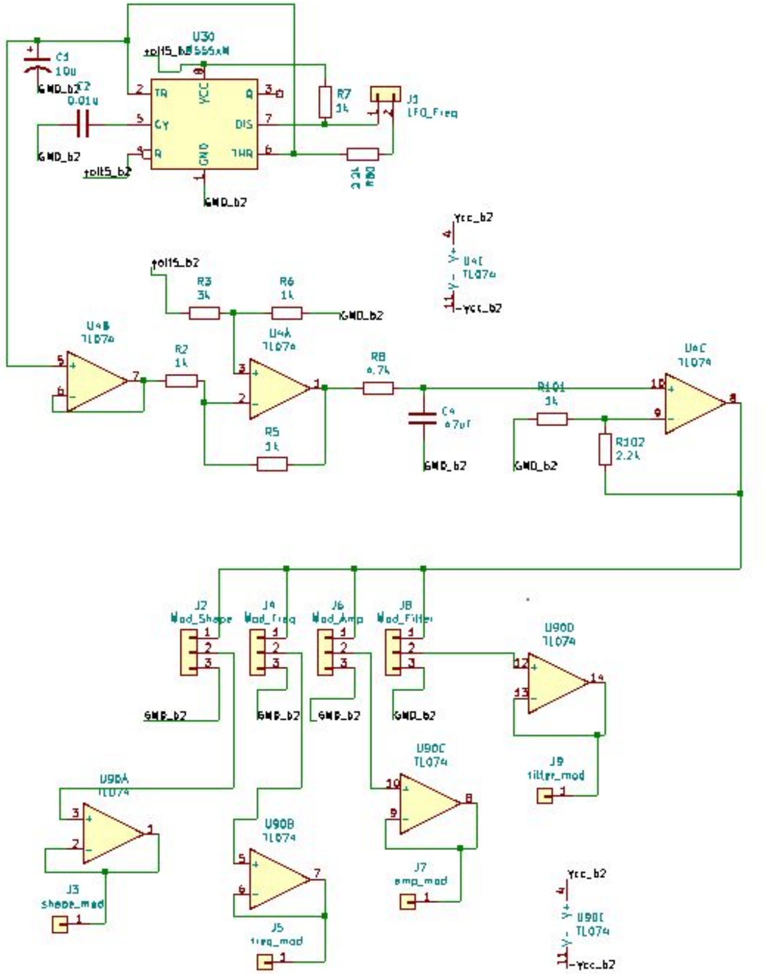


# Design: Envelope Generator

# Design: Envelope Generator

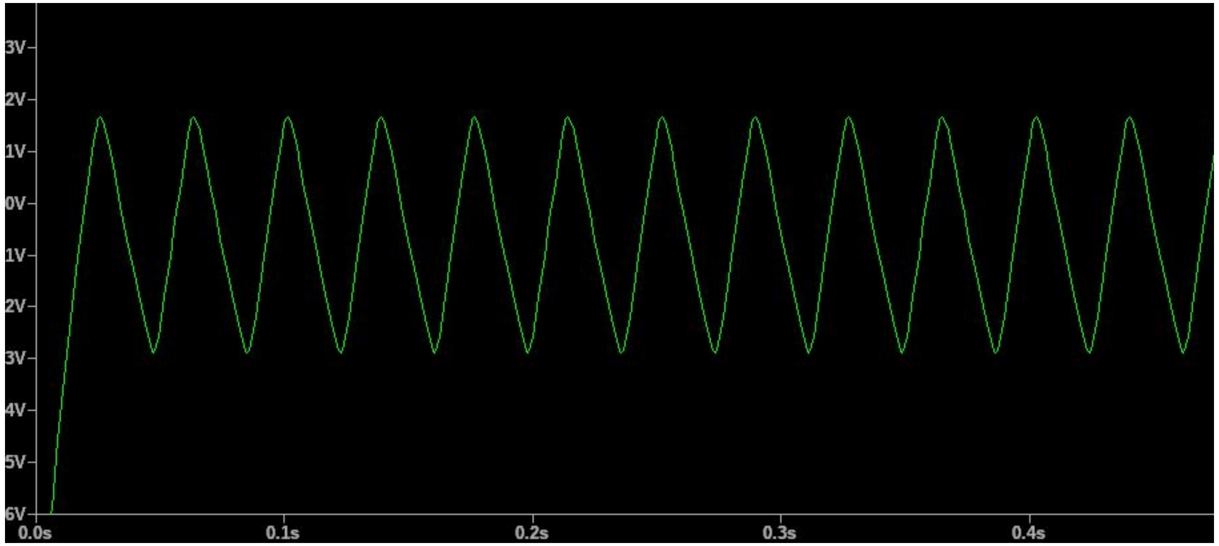



| Requirements | Verification |
|---|---|
| Envelope generator has attack, decay, sustain and release phases which can be controlled by potentiometers. | View the output with an oscilloscope. Change the potentiometers and verify the envelope changes correctly. |



# Design: Low Frequency Oscillator

# Design: Low Frequency Oscillator

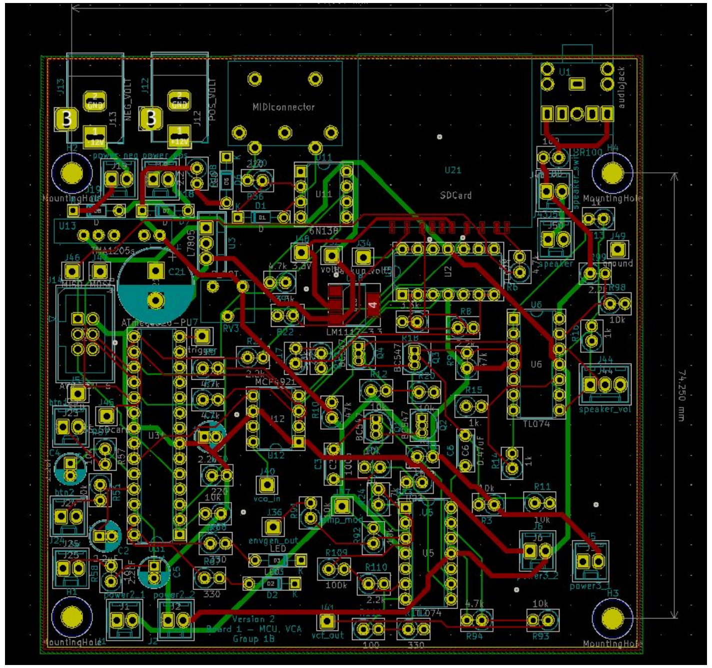



| Requirements | Verification |
| --- | --- |
| LFO produces a waveform from about -2V to +2V with a frequency of about 1Hz to 20Hz. | View output of oscilloscope. Change potentiometers and verify the signal changes appropriately. |

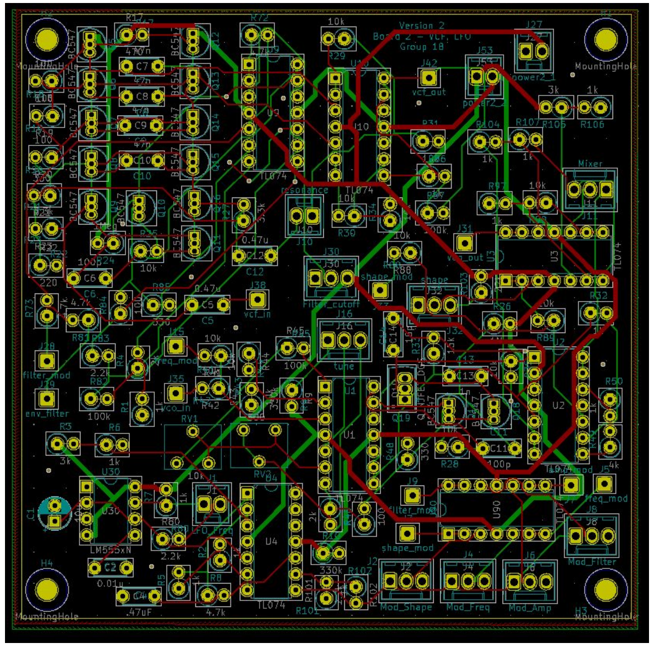
# Build

# PCB Designs

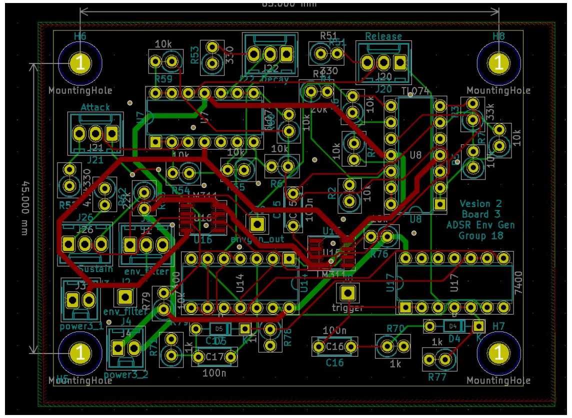
Power, Microcontroller, VCA



VCO, Low Pass Filter, LFO

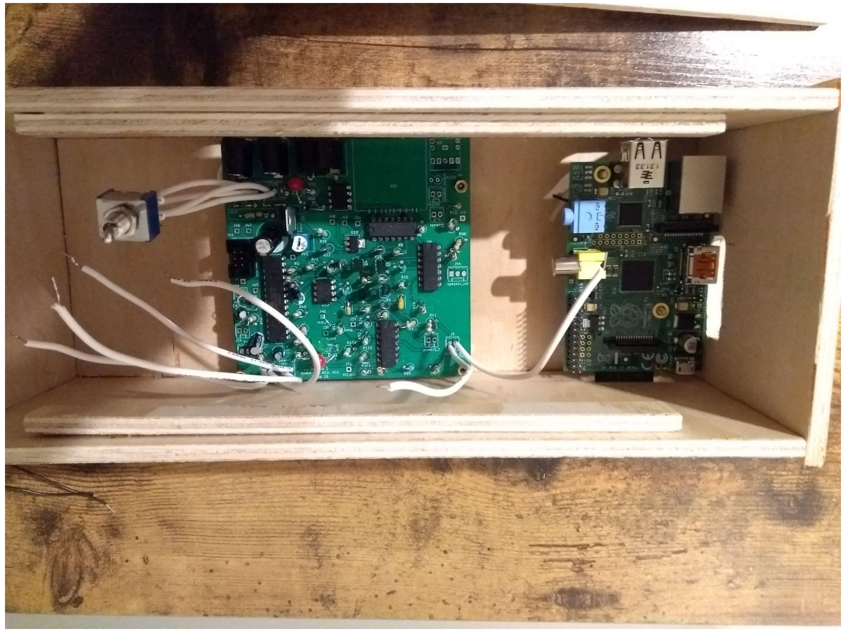


Envelope Generator



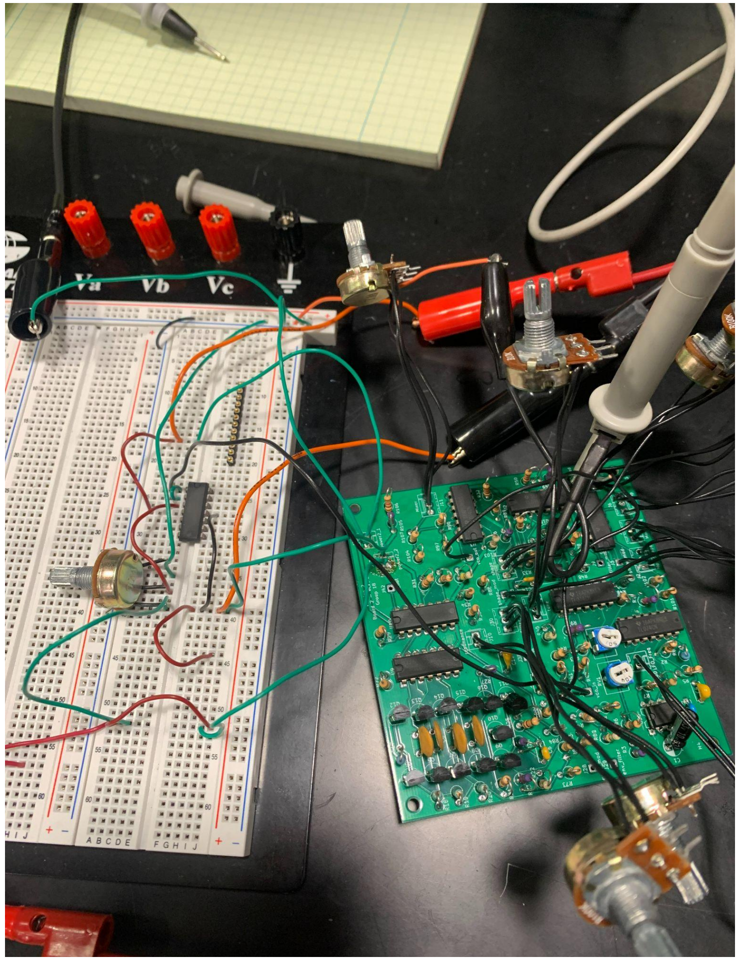
## Case

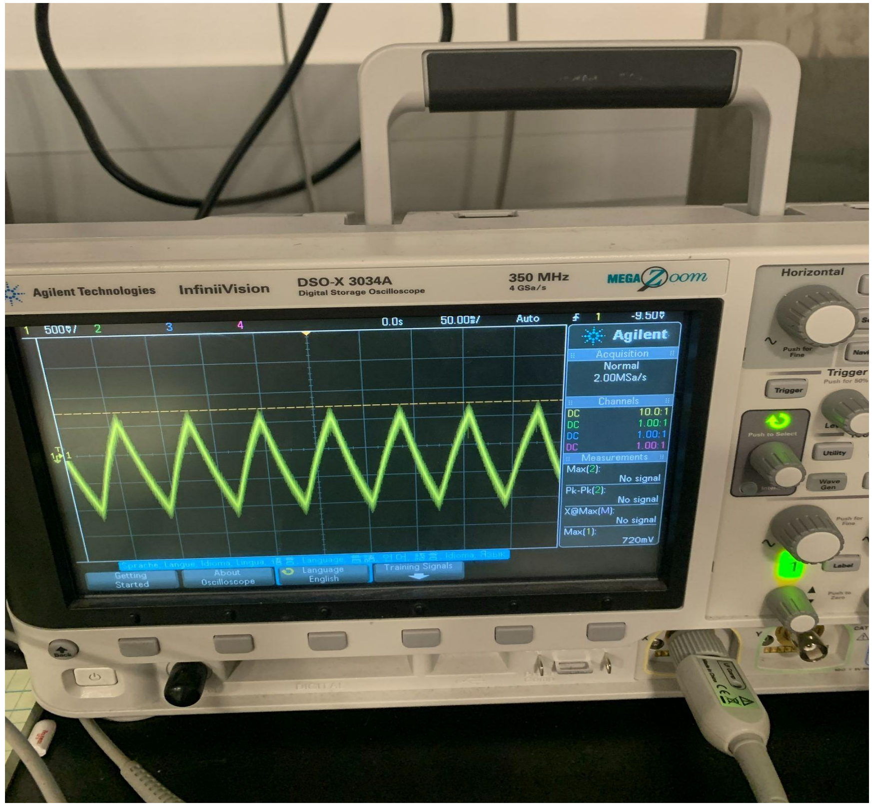
- Plywood case to hold the PCBs
- Holes in case for IO



## Low Frequency Oscillator

- Created the simulation on breadboard for VCO input (the input to LFO)
- Probed the PCB to result in a triangular wave on the Oscilloscope
- Output on Oscilloscope performed as expected: increasing amplitude when knob was turned and increasing frequency.






Low Frequency Oscillator Output on Oscilloscope

# Build

```
float vref = 5;

// see https://cyberblogspot.com/how-to-use-mcp4921-dac-with-arduino/
void setVoltage(uint16_t value)
{
  uint16_t data = 0x3000 | value;
  digitalWrite(SS_DAC, LOW);
  SPI.beginTransaction(SPISettings(16000000, MSBFIRST, SPI_MODE0));
  SPI.transfer((uint8_t)(data >> 8));
  SPI.transfer((uint8_t)(data & 0xFF));
  SPI.endTransaction();
  digitalWrite(SS_DAC, HIGH);
}

void loop() {

  float vout = 1;
  digitalWrite(LED1_PIN, HIGH);
  setVoltage( round(vout/vref * 4096) );

  delay(100);
  digitalWrite(LED1_PIN, LOW);
  vout = 3;
  setVoltage( round(vout/vref * 4096) );

  delay(100);


  // /dac output = vref*val/4096 --> val = round(out/vref * 4096)
}
```

## Microcontroller

- Managed to use the microcontroller to produce different voltages as intended.
- Were able to demonstrate this through an LED blinking switching the voltage output from 1V to  3V.

# Conclusions and Future Work

# Conclusions

## Problems:

- Mistakes in the PCB: we discovered two pins that were supposed to be connected but weren't, and there were probably other mistakes.
- Noisy power supply: Even if the whole circuit was working, the power supply was much noisier than we anticipated, with 300 mV peak-to-peak variation around 12V.
- Components: The switching 5V regulator we planned to use didn't work. We were also missing connectors which forced us to solder wires directly to the board, which made it harder to manage.

# Conclusions

## Success: Utilizing Raspberry Pi and External Speaker

- Playback synth sounds using keyboard press
- Used preloaded songs on SD card to play back synth songs
- Recreated triangle and square wave sounds
- Able to play 2 octaves of keys using keyboard press

# Conclusions

## Song Format on SD Card:

8e5

Duration
(eighth note)

Note (E)

Octave

T1.5

8e5 8D5 8b5 8G4 8e4 8D5 8b5 8G4

8e5 8D5 8b5 8G4 8e4 8D5 8b5 8G4

8e5 8C5 8a5 8G4 8e4 8C5 8a5 8G4

8e5 8C5 8a5 8G4 8e4 8C5 8a5 8G4

8e5 8D5 8b5 8G4 8e4 8D5 8b5 8G4

8e5 8D5 8b5 8G4 8e4 8D5 8b5 8G4

8e5 8C5 8a5 8G4 8e4 8C5 8a5 8G4

8e5 8C5 8a5 8G4 8e4 8C5 8a5 8G4

# Conclusions

## Future Work:

- Fix the PCB
- Further development of microcontroller design to allow for accurate MIDI input
- Design on bigger PCB and make bigger areas on PCB for easier probing
- Simplifying design of synthesizer
- Breadboard the design first, especially for unfamiliar components like the 6N138 optoisolator.

# Questions?

The Grainger College of Engineering

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN