**9/13/2021**

**Summary**

During the pandemic I ended up working with plenty of electronics and code because electronic parts are cheap and all you need to code is a computer. Winter my freshman year I built a simple synthesizer for ASME’s Fabrication and Design Essentials program. It used an Arduino to detect the position of my hand on a membrane potentiometer to generate a tone and several potentiometers to modify the tone’s properties. It’s been my favorite project so far and I ended learning a lot about electronics and signal processing.

**Inspiration**

This was not the first time someone has built a synthesizer using a membrane potentiometer. I got my idea from Wintergatans modulin. Wintergatan is a musician and engineer who creates cool musical instruments. I like their work which is partly why I want to be a mechanical engineer. When researching synthesizer design I also found Moritz Klein’s guide on active filters. I thought the way he explained signal processing super cool so I included an active low pass filter in my design.

**The Design**

I produced this video before the housing was done.

Stats

>84 playable notes from A0 to A flat 7

>Using the tuning potentiometer, the synthesizer can be tuned to any 2 octave interval within the 84 note range.

>Active analog low pass filter for mellow sound (sine wave) or nasal sound (square wave)

>Vibrato can change pitch from 0 - 4 hertz and occurs at a frequency of 5 hertz

>Portamento or slide between notes can go from air raid siren smooth to none at all >Local Gain control has range of 5 to 0 V

I used CAD to create the housing of the synthesizer but ended up modifying it to be simpler to accommodate my very blunt hand saw.

I also created a circuit diagram

**Challenges**

Getting to carried away with coding

I was super excited to start building my synth but it took a while for some of the electrical components to come in the mail so I ended up coding all the software in one night. As you would expect when I did assemble everything my code was a buggy mess! I had to spend another couple hours debugging the software which wasn’t so fun. This was also my first time programming in C++ so there were of errors I didn’t understand. Next time I do a project I will definitely try to assemble the components first and then methodically build my code instead of doing it all at once and hoping that it will work.

Membrane Potentiometer Signal Noise

Once I received my membrane potentiometer in the mail and hooked it up to my Arduino, I realized the signal was very noisy which made the speaker alternate between notes in not so nice sounding ways. I ended programming a smoothing function within my code to average out the past 10 readings in order to smooth out my signal.

Membrane Potentiometer sending Signal when not depressed

When I removed my hand from the membrane potentiometer the speaker would still play. Having space between notes in music is vital so I added a pulldown resistor to the membrane potentiometer which had it send a signal of 0 when not being played. I found this solution online from a blog by Russel Smith. The drawback of this solution is that the readings from the membrane potentiometer were no longer linear. To fix this I took readings of the potentiometer at 2 cm intervals. Using excel, I calculated a polynomial that did a decent job of linearizing the data and incorporated it into the readpot() function in my code.