SCFV-I: Synthesizer Controller With Ferrofluid Visual

Daniel Valentine

Abstract

This paper details the design process for the SCFV-I. The synthesizer features a mechanical keyboard to trigger notes as well as 3 potentiometers to control effects parameters. Users may cycle through 3 preset effects banks by means of the 4th potentiometer. The blue LEDs indicate which effect bank is in use. The controller also features a ferrofluid visual that animates the ferrofluid according user input by means of a custom made electromagnet.

Overview

The SCFV-I is a synthesizer with a ferrofluid visual. The SCFV-I makes use of an Arduino Uno Rev 3 microcontroller to communicate data, presented in the form of DC voltages, to the Arduino Interactive Developer Environment (IDE). This data is then used in Max, a visual programming language for music, to synthesize sound. The device also contains a ferrofluid visual that animates the fluid when a key on the keyboard is pressed. This as well as the lighting of the LEDs is done through the Arduino IDE and its digital output pins.

Ferrofluid & Electromagnet

Ferrofluid is a material composed of small ferromagnetic particles suspended in a fluid. The ferromagnetic properties make the material temporarily magnetic when in the presence of a magnetic field. This is useful for the purpose of this design as it serves as the basis for animating the fluid.

In order to create a magnetic field that can be controlled by a signal an electromagnet known as an inductor, an insulated wire wound in a coil, can be employed. As current begins to flow through a wire a small magnetic field is created around the wire in a collection of concentric circles. Creating a loop of wire amplifies the strength of the magnetic field in the center of the loop as the magnetic field adds up in the center. Stacking these loops by wrapping more and more copper wire around a fixed axis results in an even stronger magnetic field.

Figure 1 displays the relationship of the strength of the magnetic field (B) and the total current flowing through the inductor (I). From this relationship it is obvious the number of turns per unit of area (N) should be increased to create the strongest magnetic field while consuming the least amount of current.

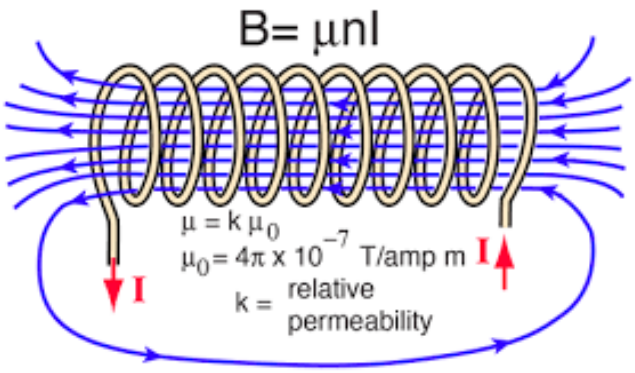

Figure 1

Design of Electromagnet

The electromagnet consists of a simple inductor. By passing current through its terminals a magnetic field strong enough to animate the ferrofluid is created. Placing a piece of iron in the center of the inductor amplifies the strength of the magnetic field, as the iron, which is ferromagnetic, begins to behave like a magnet as well. This contribution is significant, thus a model holding the copper windings as well as the iron core was 3D printed using FDM printing (Figure 2) by means of an Ultimaker S3. The iron core fit snuggly the holder and needed to be hammered into place. The model created in Figure 2 was created in Autodesk Fusion

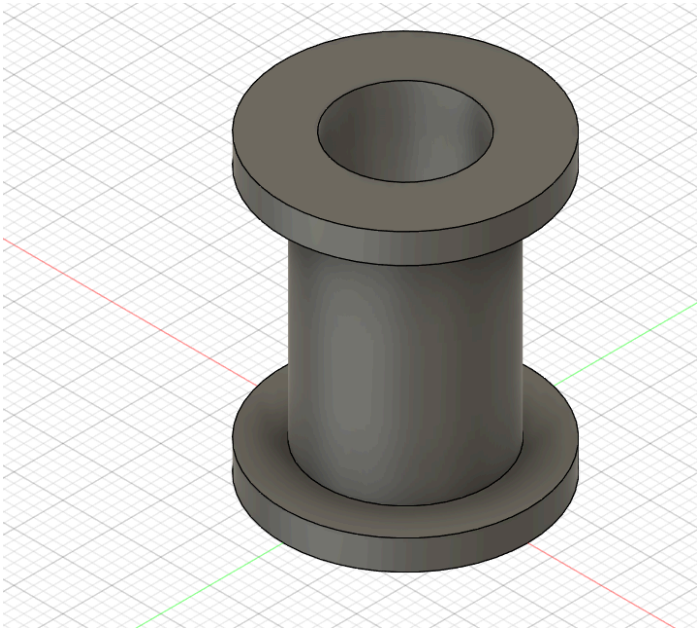

Figure 2

An initial prototype was created using 26 AWG (American Wire Gauge) enameled copper wire. A single layer of copper windings were spun by hand. The iron core was too wide to fit in any drill which conflicted with the initial plan to spin wire around the holder while rotating the core.

Hand spinning the windings resulted in a less compact and thus more inefficient design. The 26 AWG prototype inductor operated as expected, creating a magnetic field strong enough to move the ferrofluid. The 26 AWG prototype was then unwound in favor of 28 AWG wire. 28 AWG wire was chosen as it was thinner and yielded more turns in the same area and therefore theoretically a stronger field. The core and holder were rewound using 2 layers of 28 AWG wire which again operated as expected. The iron core had approximately an additional 5 inches in length. This material was removed using a hacksaw. The rough edges left by the hacksaw can be seen in Figure 3.

Figure 3

The final inductor presents heat as a concern. When passing 3 A of current through it the copper wire of the inductor begins to get too hot to touch, reaching temperatures above 70^{o} C in less than 30 seconds. The inductor also begins to emit a chemical smell at this amperage as well. Over 4.5 A and the inductor begins to smoke. Both the smell and smoking are likely the PLA plastic melting or burning. The copper wire is rated for 155^{o} C so it is doubtful that it is having issues with the heat. The best operating conditions for the inductor was at around 2 A as the magnetic field was strong enough to create noticeable effects in the ferrofluid while not getting hot too quickly.

Design of Electromagnet Control Circuit

The inductor designed in Figure 4 requires at least 900 mA to begin to start showing noticeable effects in the ferrofluid. The digital outputs of the Arduino can supply at most 40 mA, any more would risk damaging the Arduino. To supply the currents needed a PA305D3 power supply was used. This power supply is capable of 5 A at 30 V, more than enough power for this application. The Arduino's digital output is used to trigger the gate of a IRLZ24NPBF N-Channel MOSFET. When the voltage at the gate goes beyond a certain threshold, a larger current from the power supply is allowed to flow from drain to source. For the purpose of the design the threshold is irrelevant as the Arduino outputs only 5 V of 0 V. The 5 V output is always well above the threshold and 0 V output is always well below the threshold. The introduction of the MOSFET allows for the large current source to be quickly connected and disconnected from the inductor.

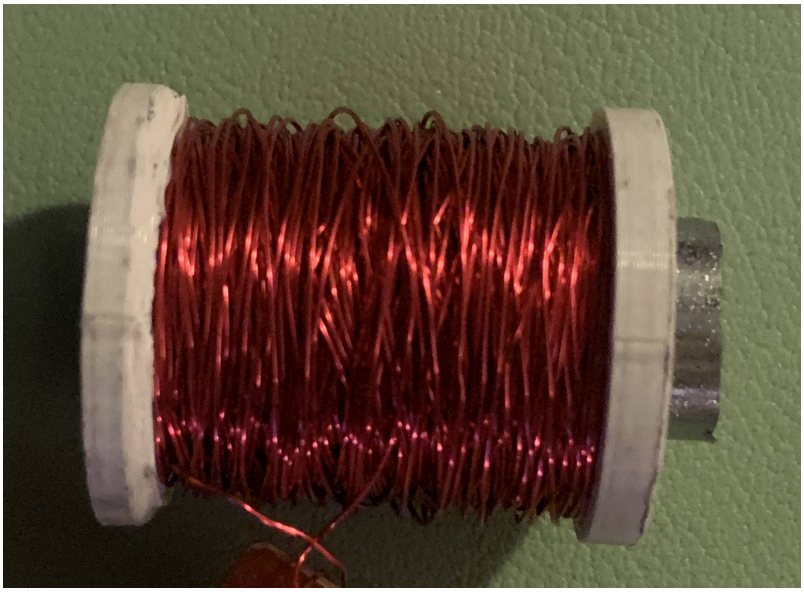

Figure 4

The electromagnetic control circuit (Figure 5) also features 2 resistors which serve to limit stray currents that would turn on the MOSFET. The diode placed in parallel to the inductor protects the MOSFET from the back EMF created by the inductor. The back EMF is a voltage generated by the inductor in the opposite direction of the applied voltage. The back EMF occurs because the magnetic field generated by the inductor induces a current in the opposite direction of the supply voltage. The MOSFET can only handle current flow in 1 direction and a reverse of the current flow could break the component. The diode blocks this current and is thus necessary.

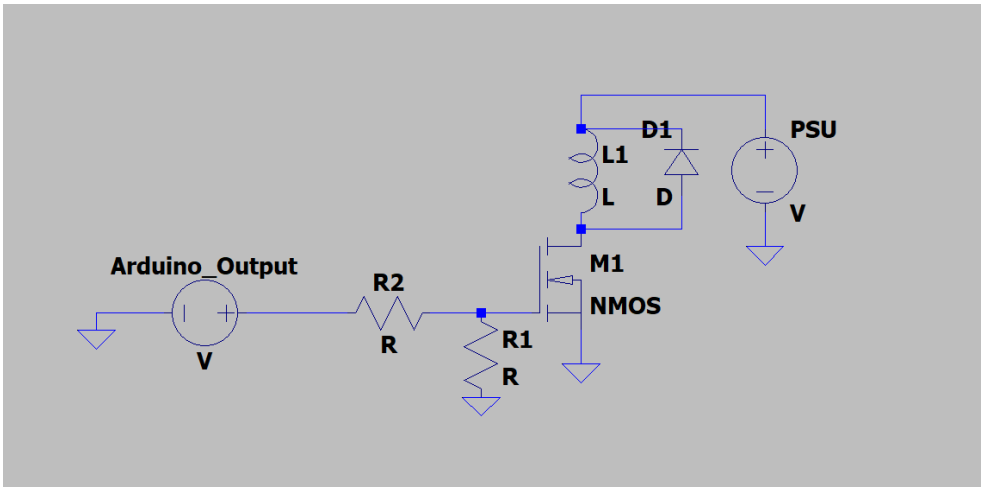

Figure 5

Before the diodes or MOSFETs were purchased a circuit simulation was created in LTSpice. The initial motivation of this circuit simulation was to create a 1-1 simulation of the physical circuit before construction. The first step in this process was to find the inductance of the inductor. The method employed makes use of a Koolertron GH-CJDS66-A function generator and a FNIRSI-1014D oscilloscope. First, the output impedance of the function generator was measured. This was done by connecting the function generator to the oscilloscope and observing the amplitude of an arbitrary waveform, in this case a sine wave was used. A load resistor was then connected in series with the function generator and a different smaller amplitude was observed. Using Equation 1 the output impedance of the function generator was calculated. R_0 corresponds to the output impedance, R_L corresponds to the resistance of the load resistor, V_0 corresponds to unloaded voltage, and V_L corresponds to the unloaded voltage.

$$\text{Equation 1: } R_0 = R_L\left(\frac{V_0}{V_L} - 1\right)$$

Then a 20 kHz sine wave was applied to the inductor. The frequency output from the function generator was then lowered until a waveform ½ the amplitude of the original 20 kHz waveform was observed. Using this and Equation 2 the inductance (L) was calculated to be around 5 mH. The f term corresponds to the frequency of the ½ amplitude waveform.

$$\text{Equation 2: } L = \frac{R_0}{2\pi f}\sqrt{\frac{1}{3}}$$

This relationship makes use of the reactive (opposition to alternating current) properties of the inductor to calculate its inductance. The inductance of the inductor gives the current voltage relationship of the component (Equation 3) similar to how Ohm's law gives the current voltage relationship for the resistor. Thus this parameter is imperative to the circuit simulation.

$$\text{Equation 3: } V = L\frac{di}{dt}$$

LTSpice comes with a large library of components whose parameters have already been measured and documented. This library of components, in particular the MOSFET library, did not have any cheap parts with quick shipping times. This led to using the circuit simulation as a tool to test component parameters similar to components that could be delivered cheaply and quickly. Eventually the IRLZ24NPBF N-Channel MOSFET was chosen as it could handle 18 A at 55 V. The diode chosen was a Tnisesm 20 A diode. This diode is capable of handling 20 A at 1,000 V. Both of these components have ratings far beyond this design's applications (the limit being 5 A at 30 V of power supply) but were chosen just to be safe.

Once components were in hand a prototype circuit was created on breadboard. In this prototype the values of R_1 and R_2 in Figure 5 were adjusted. If these resistances were too low the MOSFET would stay on after being triggered as the stray currents would keep it in its conducting state. If these resistances were too high the MOSFET would sometimes not trigger. The final values for the resistors were approximately 680 Ω and 68 kΩ for R_2 and R_1 respectively. All the resistors used were rated for a ¼ W and had no issues with heat under all operating conditions. The MOSFET also had potential to become hot due to the large current flowing through it, however

in all testing the MOSFET did not get hot to the touch. This meant that the MOSFET did not need an additional heat sink. Once the circuit was operating as expected on bread board the final circuit was soldered together in the EXP Makerspace.

The diode did present an issue in the fabrication process of the final electromagnet control circuit. The width of the diode's leads was too wide to fit into the holes on the perfboard. This resulted in needing to bore through the hole to make it larger using a screwdriver. The solder joint connecting power to the inductor did break once in transit from Northeastern University. Resoldering it was a simple process but does raise concerns in regard to the integrity of that joint. The addition of the MOSFET also saw substantial decreases in efficiency. With the inductor connected to just the DC power supply an 10 V was needed to drive around 4.5 A, however once the MOSFET and other components were added 27 V was needed to drive 4.5 A.

Design of Input Controls

The input controller is made up of 3 distinct elements, the keyboard, potentiometers, and the Arduino. The potentiometers take in 5 V from the Arduino's 5 V pin. Turning the knob of the potentiometers sweeps through the 0 V and 5 V and changes the voltage read at the analog read pins 1-4. This voltage is scaled in the Arduino IDE to values between 0 1023.

The keyboard operates by means of 12 switches placed in parallel with each other. Each switch is connected in series with a trimmer potentiometers and is fed 5 V from the Arduino's 5 V pin. The trimmer potentiometers serve to finely tune the output voltage of each switch. The output voltage can vary between 0 V and 5 V and is read by analog read pin 0. It is then scaled in the Arduino IDE to values between 0 1023. The voltages output by these switches serve to control the frequency of the synthesizer in Max. A design of this model was first prototyped on breadboard (Figure 6). The keyboard also makes use of tactile mechanical keyboard switches (Figure 7). These switches have a unique footprint and cannot be fit on perfboard, requiring the design of a printed circuit board (PCB).

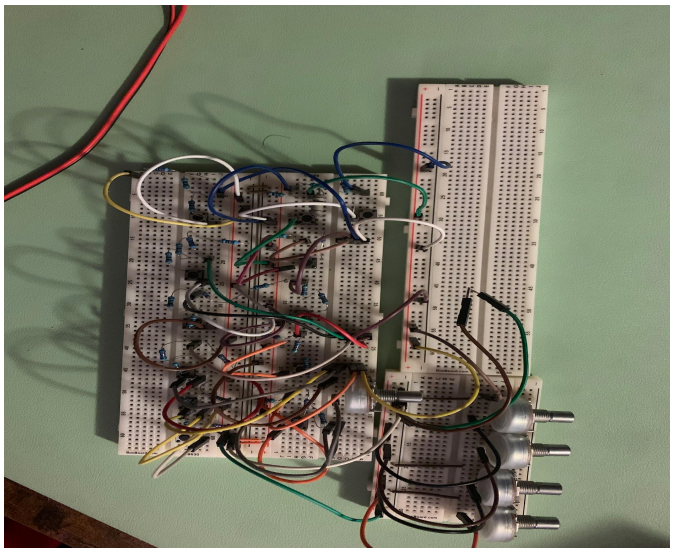

Figure 6

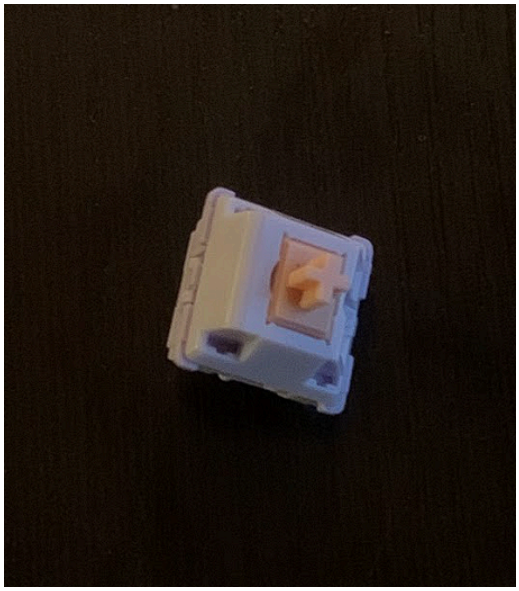

Figure 7

A first attempt at PCB to hold these switches was designed and milled using a Bantam Tools Desktop CNC Milling Machine. The design and gcode (instructions for the mill) were created in Autodesk Fusion (Figure 8). The dimensions of the switches, header pins, and trimmer potentiometers that were to be placed on the board were carefully measured beforehand to ensure the footprint (which is the physical outlines of its leads and supports).

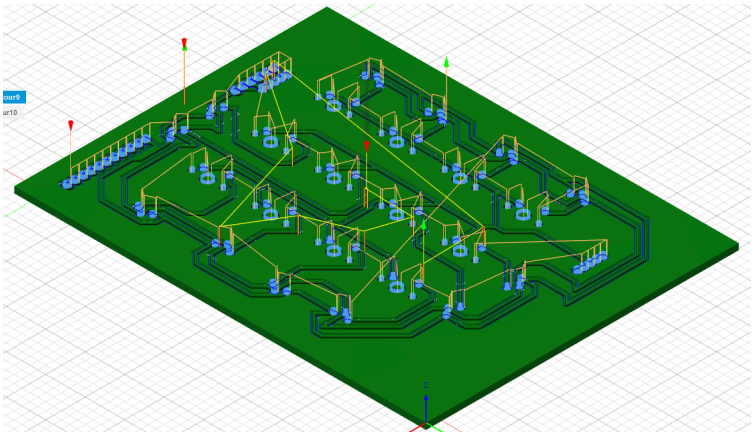

Figure 8

The mill operates by removing a thin copper layer from the rest of the stock. The material is removed also to form isolated traces of copper that can carry current between components. This first PCB took around 3 hours to mill and had several failure points. Firstly the mill was not able to mill out much of the bottom right of the stock (Figure 9).

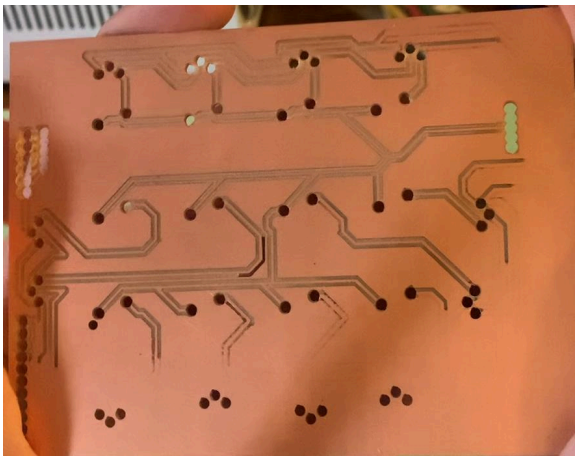

Figure 9

This likely occurred due to an uneven pressure applied to the board while milling. The mill secured the stock on the top and bottom edges of the stock, while no pressure was placed on the sides of the stock. The uneven pressure likely bowed the stock slightly causing the drill bit not to make contact with the stock. Secondly, while nearing the end of the milling process the stock suddenly shot out several inches to the right, immediately causing the mill to shut down. Again the uneven pressure applied to the board may have forced it out of position. Finally, the mill lacked the precision needed for the design as some of the copper traces appear to overlap on each other.

The failure and time it took to mill the first board led to the design of a multilayer PCB. This board has 2 layers of conductive material which are separated by an insulator. Each conductive layer is also covered with a finish to protect it. The presence of 2 layers of conductive material makes designing the PCB significantly easier as traces can be drawn through each other without them actually intersecting as they are on separate layers. Figure 10 displays the new PCB design, which was created in KiCAD.

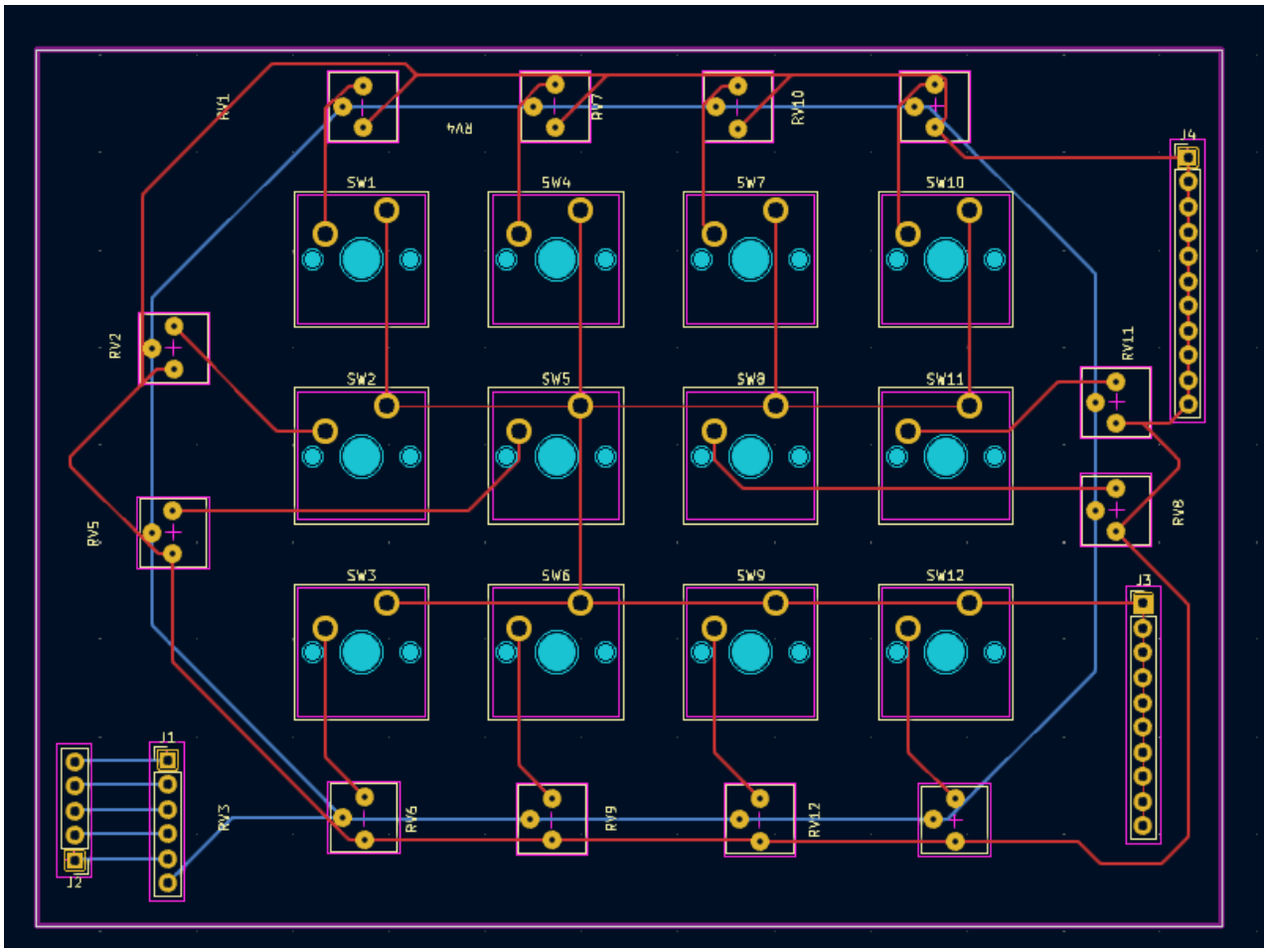

Figure 10

The red lines connecting vias represent traces placed on the front face of the PCB and the blue lines represent traces placed on the back of PCB. The header J1 connects the potentiometers to the keyboard PCB. The J2 header is connected to J1 and serves to route cables to the Arduino's analog read pins. Pin headers J3 and J4 connect to power and ground respectively. The small boxes with 3 golden holes are footprints of the trimmer potentiometers and the larger boxes with the 3 blue holes are the footprints of the mechanical keyboard switches. The 2-layer was also simpler to design as it came with the footprints preloaded in KiCAD.

The PCB was designed according to the specifications outlined by the PCB manufacturer, PCBWay. These specifications allowed for finer spatial resolution when designing the PCB than the mill, again making drawing traces easier. The Gerber file of the PCB was sent to PCBWay and fabricated somewhere in China. Turnaround time for this process was surprisingly quick as the PCB was delivered in about a week.

While sending the PCB to a professional manufacturer yielded a functional result it was much more costly than the mill. PCBWay requires at least 5 copies of a single PCB be ordered. This combined with the shipping cost made the fabrication of the 2-layer PCB one of the most expensive parts of the final product. Soldering the electrical components onto the 2-layer PCB was trivial and was done at the EXP Makerspace. The first completed PCB worked as expected when tested.

Digital Output Design

The SCFV-I makes use of the Arduino's digital output pins to control both the LEDs as well as send 5 V pulses to the gate pin of the MOSFET. Whether an LED is on or off is dictated by the potentiometer connected to analog read pin 4. The output of the potentiometer is read by a 3-way conditional statement in the Arduino IDE. These conditional statements set three ranges of values and depending on which value the voltage input at pin 4, a different digital output pin will output 5 V, thus lighting a different LED. Each LED has it's own 5 kΩ current limiting resistor connected in series with it. The LEDs are connected to digital output pins 11-13. Digital output pin 10 is connected to the gate of the MOSFET. This pin outputs a voltage when the voltage at analog read pin 0 reaches beyond a certain threshold voltage. This is again controlled by a conditional statement in the Arduino IDE. The threshold voltage needed to be tweaked in order to ensure that noise at analog read pin 0 would not trigger a pulse from digital output pin 10. Furthermore, the threshold voltage needed to be raised so that the MOSFET would not continue to conduct after the Arduino ended the 5 V pulse.

Housing Design - 3D Printing

Designing the housing proved to be one of the most challenging aspects of this whole project. To fabricate the housing FDM 3D printing was employed by means of an Ultimaker S3. Figure 11 displays the first prototype which was designed in Autodesk Fusion. This model had several issues, which can be split into 2 categories; issues with the design and issues with the 3D printer.

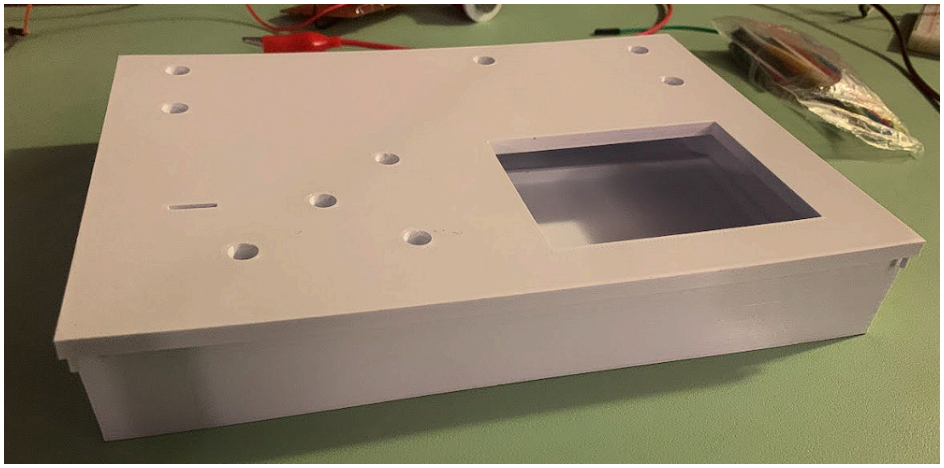

Figure 11

The first design issue was the placement of the dowels meant to support the Arduino. The dowels were several millimeters off from where they needed to be to fit the Arduino. This portion of the design was iterated upon and worked well in the 2nd print (Figure 12).

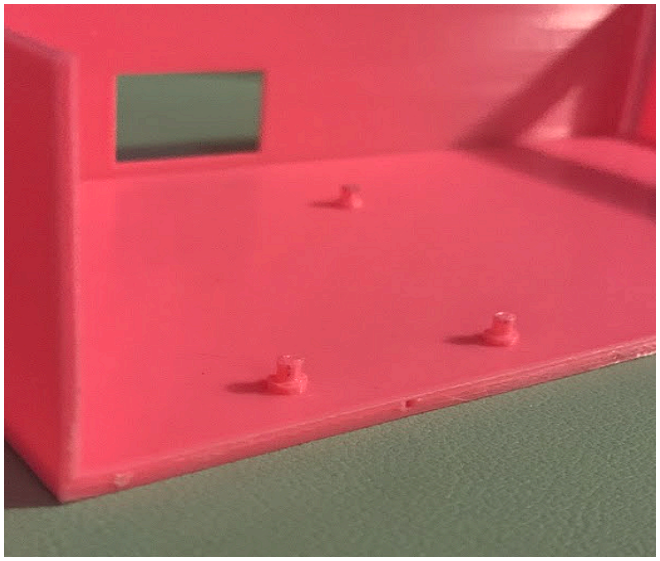

Figure 12

The keyboard also failed to fit in the lip printed to hold it. This was again addressed in the 2nd print. The final issue was that the keyboard and Arduino could not fit comfortably in the housing. This was addressed by making the housing slightly wider. Unfortunately this increase of width meant the design was too wide to fit on a single 3D printer. This led to the design being split in half, with each halved being printed separately.

The 3D printer faced two main issues, poor resolution quality as well as deformation. The poor resolution quality means that the 3D printer struggled to print pieces exactly according to the specification in the model. Take for example the dowel and hole combination in Figure 13.

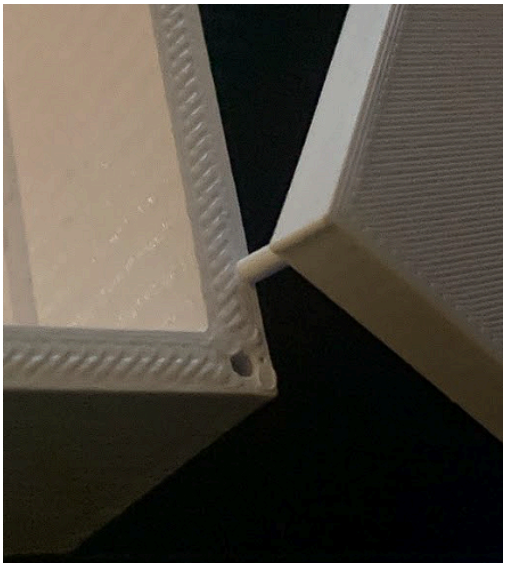

Figure 13

The dowel measures 0.5 mm in diameter and is narrower than the hole, so it should fit. Due to the poor resolution quality it did not fit. This happened despite setting the printer to *Fine - 0.15 mm* setting. To circumvent this poor resolution the dowel and hole mechanism was redesigned (Figure 14).

Figure 14

Longer thinner prints always came deformed as well (Figure 15). This occurs because as the plastic cooled, it began to shrink and pull in on itself, warping the print. A solution to this issue was not found in time for the final product.

Figure 15

The 2nd iteration of prints was also modeled in Autodesk Fusion. Some of the improvements included the aforementioned dowel-hole re-design and an adjustment to the lip holding the keyboard. To accommodate the splitting of the housing into 2 parts an additional holder was added. This holder proved to be unnecessary in the final product as the wires connecting the Arduino to the keyboard PCB held the device together. Additionally the top part of housing was made thinner to allow for the thread of the potentiometer to be exposed (Figure 16). Figure 17 displays the final housing design in Autodesk Fusion.

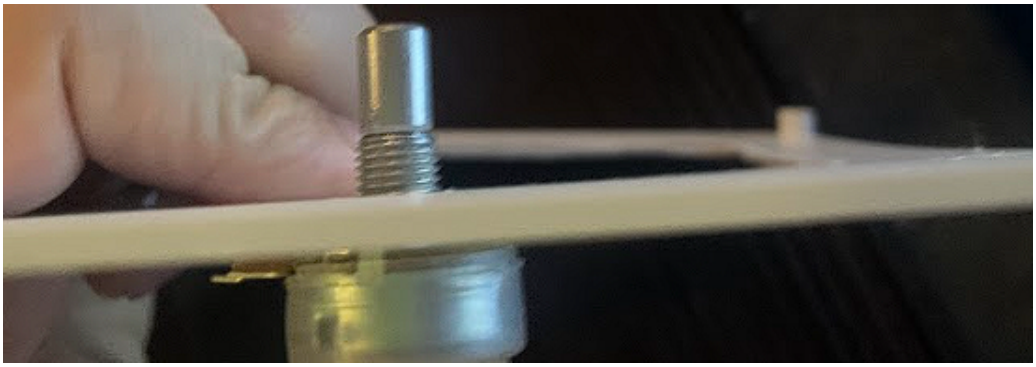

Figure 16

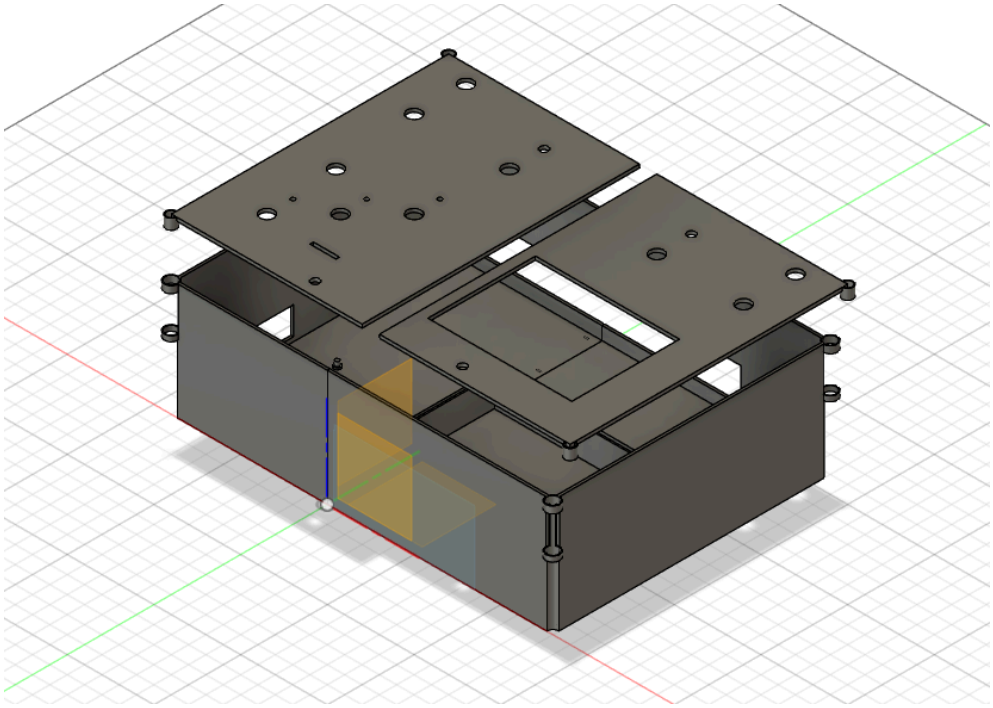

Figure 17

Figure 18 displays the holder means to hold the ferrofluid and inductor. This print saw deformation in the rings on its side. This occurred due to lack of support during the printing process causing the hot plastic to sag. Other than the sagging of the side rings the holder came out as expected.

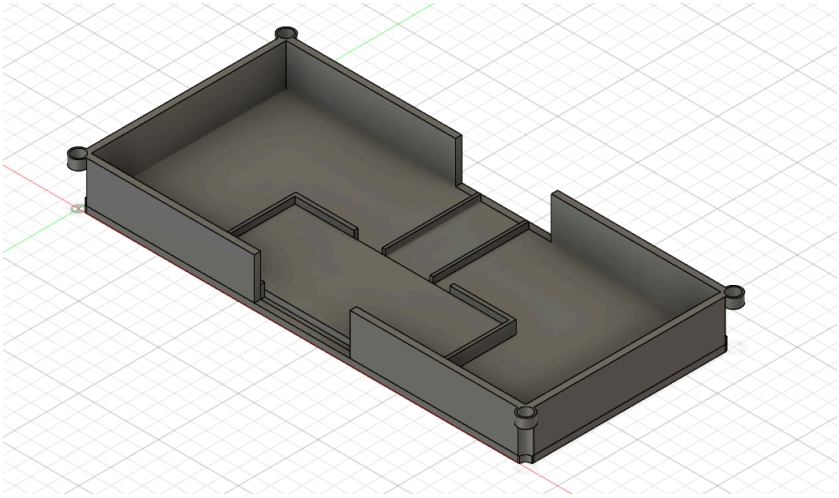

Figure 18

Printing also proved to be a somewhat costly and very time consuming process. A typical 40 gram print would cost around $5 and would typically take around 5 hours to print. This coupled with the consistent poor quality of the prints made finishing the housing very difficult.

Housing Design - Wiring

To wire the potentiometers and LEDs with the keyboard PCB and Arduino, the components were first mounted to by the top part of the housing. The potentiometers were secured via nut and washer combination and LEDs leads were bent outward to secure it in place. Solid core wires and resistors were then soldered onto the components and connected to jumper cables to connect to the PCB and Arduino. Solid core wire is not ideal for this application and it runs a much greater risk of being split under mechanical strain than stranded wire, meaning more care was needed in bending the wires into place. Additional jumper cables were connected from PCB to Arduino to transmit data from the potentiometers. Zip ties were also employed to make the connection of 5 V wires and ground wires to the keyboard PCB more manageable. The grounds of the electromagnet control circuit and the power supply were also connected to the keyboard PCB's ground.

Code

The code driving the synthesis and audio process made use of 2 pieces of software, Max and the Arduino IDE. The Arduino IDE code reads data from the Arduino's analog read pins. It then uses serial communication to communicate that data to Max. The Arduino IDE code also outputs pulses to its digital output in accordance with data received at the analog read pins.

Some additional code is needed in Max to allow for serial communication. This code makes the data from the Arduino IDE into a list of 5 integers. These integers are used to control synthesis parameters. The data from the potentiometer connected to analog read pin 4 feeds into a scale object and then a *gate* object allowing for data to only be passed to a certain effect. Data from analog read pin 0 is mapped to the frequency of 3 sawtooth oscillators and its functionality remains the same when the voltage at analog read pin 4 is changed. Analog read pins 1-3 pass the data to be gated by the *gate* objects.

Each effect bank has three parameters that can be manipulated by the potentiometers. The first effect bank contains amplitude control for 2 of the sawtooth oscillators. These oscillators are tuned to be a 5th and octave above the core oscillator. The final effect in this bank allows for amplitude modulation of all 3 oscillators. A sine wave serves as the signal modulating the amplitude. The second effect bank contains a resonant low pass filter (lores~) as well as a cverb~ (built in reverb in Max) object. Manipulating the potentiometers allows for changing of the cutout frequency, Q factor, and reverberation time. All the integers fed into these objects are scaled to better fit the effects parameters. For example, the data output from the Arduino IDE ranges from 0 1023 and is scaled to 0. 1. in Max to make it a more suitable range for the Q factor. The final effect bank contains another lores~ object however in addition to both the Q factor and cutoff frequency being adjustable another sine oscillator allows for the manipulation of the cutoff frequency over time. This sine wave's frequency can be manipulated by means of

one of the potentiometers. The data from the analog read pins was a bit noisy with fluctuations of around ± 5 mV. This results in the frequency as well as other parameters slightly oscillating in their value in Max.

Conclusion

The SCFV-I is a novel synthesizer controller that comes with a ferrofluid display. The synthesizer comes equipped with its own Max patch that can be used to create a wide array of textures and be easily edited to its user's preferences. The SCFV-I still has plenty of room to be further developed. Further development could include the design of a dedicated non-variable power adapter meant to work with 120 V outlets as well as the implementation of an IC to make the data output to the Arduino more reliable.

.