Affordable Analog Synthesizer

Design Document

ECE 445 Design Document

Team 18: Yash Bhushappagala, Michael Jamrozy, Breanne Warner
TA: Feiyu Zhang

Fall 2021

Table of Contents

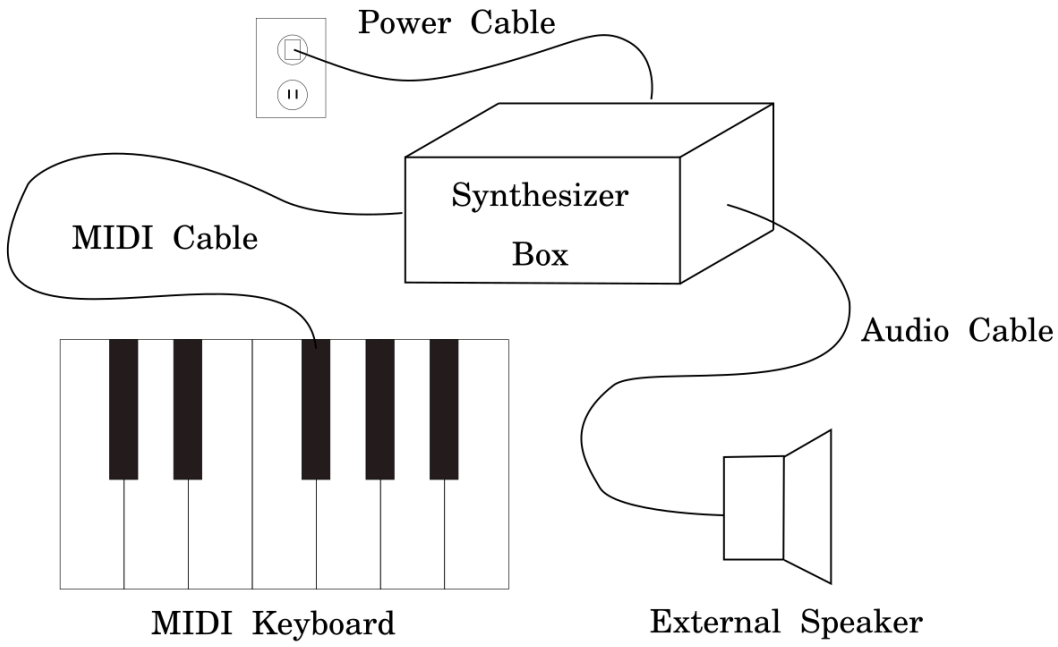
1. Introduction

1.1 Problem and Solution

To further understand the purpose that the project holds, our team did research on the interest in music synthesizers in the market. Music synthesizers are extremely expensive at market value and for most people, it is not reasonable to own a music synthesizer due to the high cost. As many people are interested in creating music, or using synths but may not have the budget to own one, the objective of the project is to create an affordable analog synthesizer. Also, According to Technavio, “the music synthesizers market is poised to grow by USD 62.90 million during 2021-2025, progressing at a CAGR of over 2% during the forecast period” [10]. Being able to create an affordable model holds values with the growth in market and demand for music synthesizers, as well as documentation for the homemade solution that we make.

In order to solve the demand for an affordable music synthesizer, creating the synthesizer from scratch and utilizing a cost analysis to obtain cheaper parts will help in implementing an effective and cost effective approach.

1.2 Visual Aid

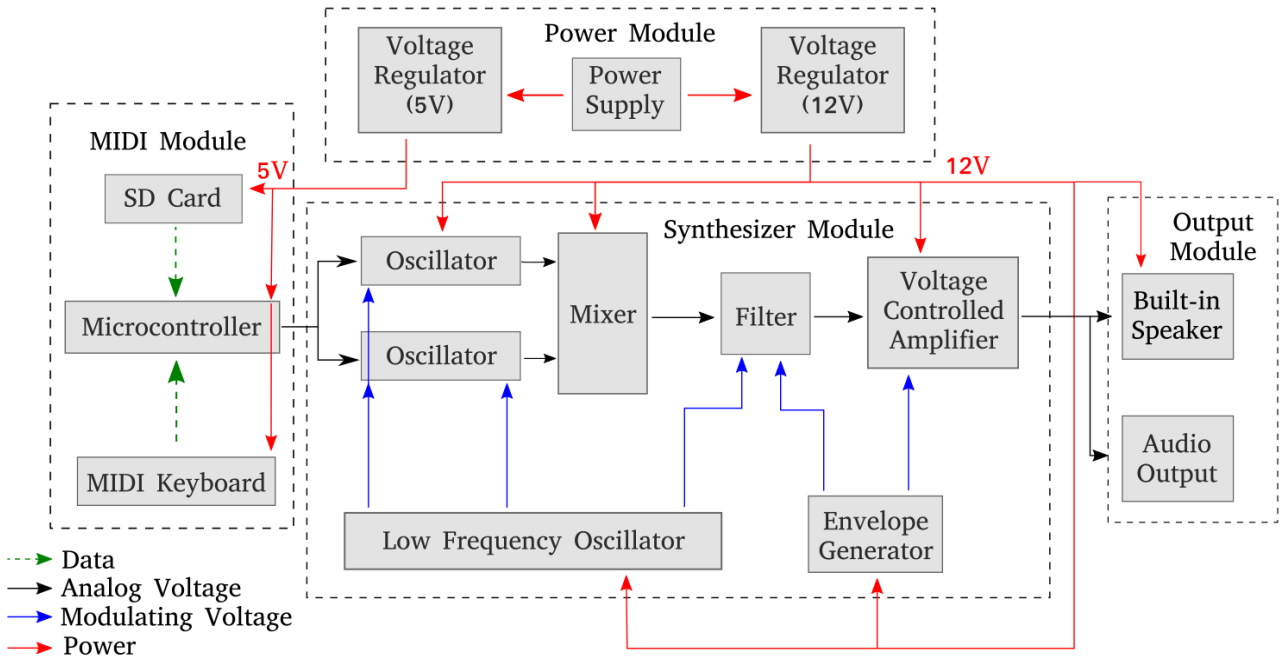


1.3 High-Level Requirements

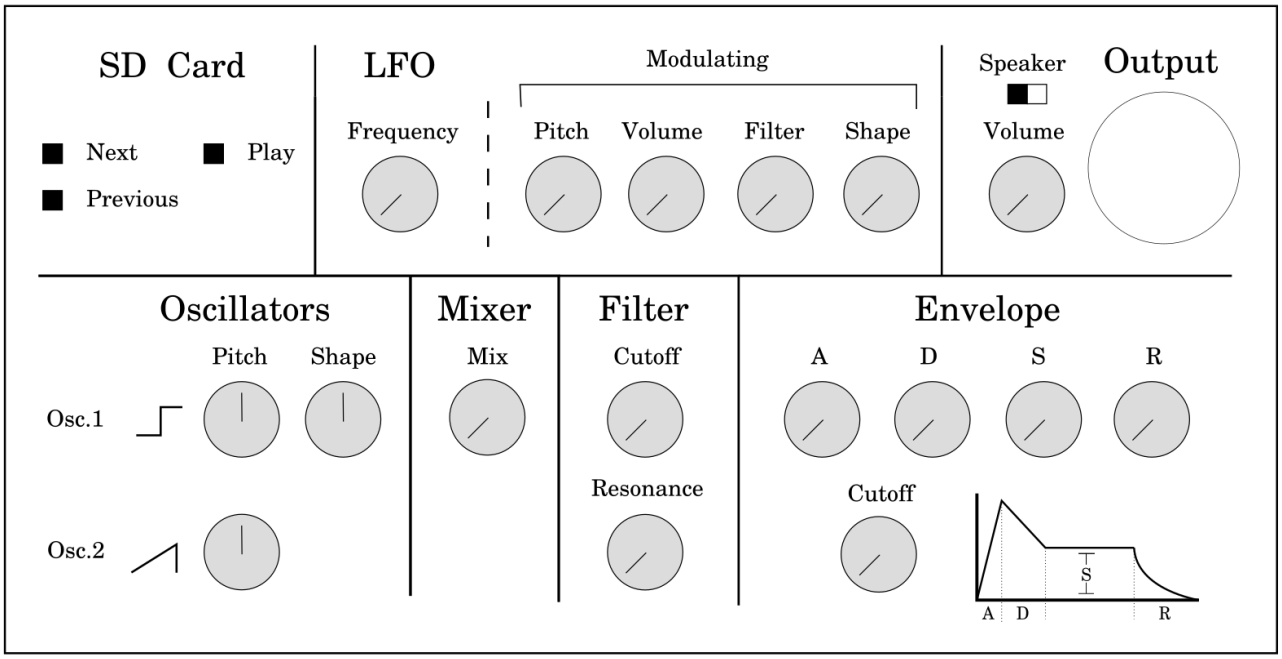
1. The basic sounds it can produce are a sawtooth wave and square wave with a controllable duty cycle. It should be capable of recreating at least the following kinds of effects: tremolo (variations in volume), vibrato (variations in pitch), sweeping cutoff filter, and resonance.
2. The synthesizer will be able to produce the correct pitches for at least 24 consecutive keys, from the MIDI keyboard centered around A4 (440Hz), so it will range from 220Hz to 880Hz [9].
3. Have the ability to read key inputs from a file containing a sequence of key events on an SD card and play them back through the synthesizer as if they were notes being played on the keyboard.

2 Design

2.1 Block Diagram



2.2 Physical Design



2.3 Subsystem Requirement and Verification

MIDI Subsystem

Requirement	Verification
1. Microcontroller can produce an analog voltage in the range from 0-5V that acts as the input to the voltage controlled oscillator.	Probe the voltage-controlled oscillator's input voltage, which is connected to the microcontroller's PWM output through a filter. Write test code to loop through every key in the keyboard and check that the voltage level matches what is expected for the given key press.
2. The microcontroller will send a specific voltage to the synthesizer subsystem oscillators depending on the key pressed on the MIDI keyboard	Use an oscilloscope in the testing phase to check that each voltage level outputting from the microcontroller is correct within the tolerance threshold before passing into the

	synthesizer module.
3. Ability to read key events from files on an SD card and send these signals to the synthesizer subsystem. The format of the file will be a sequence of key events, each event contains a number identifying what key to press, time in milliseconds it should be held down, and the time delay between the end of the note and the beginning of the next note. This is a monophonic synth so it is only capable of playing one note at a time.	Utilize SD card input into the front panel, be able to select taking inputs from SD card or the keyboard. Testing will be done with a switch and will check if the output signal from switch is either a 1 or a 0 depending on which switch is being displayed. Preload some test songs onto the SD card with certain characteristics that are easy to test. For example, have one which plays a scale and keeps each note pressed for one second and released for two seconds. Verify that the timing is correct.
4. The synthesizer has the ability to cycle through multiple files loaded onto the SD card and play them individually.	Preload the SD card with songs that are easy to tell apart. Then play the first song, stop it, switch to the next song, and repeat until it is verified that all songs on the SD card were able to play. At the end it should cycle back to the first song.

Synthesizer Subsystem

Requirement	Verification
1. Two voltage-controlled oscillators, with a voltage input in the range of 0-5V. The first oscillator produces a square wave with an adjustable duty cycle, and the second oscillator produces a sawtooth wave.	a) Hook the output of the oscillator #1 to an oscilloscope and speaker. Then vary the voltage input between 0 and 5 volts. Make sure that the output is a clean square wave, and also that the frequency increases exponentially when voltage increases linearly. b) Do the same for oscillator #2 and check that the output is a sawtooth wave.
2. A mixer combines the outputs of the two oscillators into any ratio.	a) Turn the mixer potentiometer knob and view its output on the oscilloscope while both oscillators are working. Verify that the oscilloscope shows a square wave at one extreme and a sawtooth wave at the other extreme.

3. Low-pass filter with controllable cutoff and resonance. The cutoff is voltage-controlled within a range of 0 to 5 volts. Resonance is not voltage controlled; it is controlled simply by a variable resistor which adjusts the feedback into the filter.	a) Verify that the voltage controlled cutoff works: Connect the input of the filter to a function generator's square wave. View the output of the filter on an oscilloscope. Use a power supply as an input to the control voltage, sweep the cutoff, and make sure the resulting square wave becomes smoother as the cutoff decreases. b) Now keep the voltage fixed and vary the resonance to the filter. Check to see that the resonance appears on the oscilloscope. On the oscilloscope, resonance appears as ripples after steep transitions.
4. The envelope generator creates an envelope that is used to modulate the sound's amplitude as well as the cutoff frequency of the filter. The degree to which it affects the filter is controllable with a potentiometer. The envelope first increases during the attack phase, then decreases during the decay, then stays at a constant sustain level, and then drops to 0 during the release phase.	a) Connect the output of the envelope generator to the oscilloscope and program the microcontroller to set the trigger signal every few seconds. View the envelope on the oscilloscope and make sure it has distinct phases for the attack, decay, sustain and release. Turn the attack potentiometer and verify that the attack time increases. Similarly verify the decay and release increase when those knobs are turned. Verify that the sustain level increases as the sustain knob is turned.
5. The voltage controlled amplifier has two inputs, the audio signal coming from the filter as well as the amplitude envelope from the envelope generator. The envelope controls the gain of the amplifier.	a) View the output of the amplifier with an oscilloscope. Set the control voltage with a power supply and use the audio output from the other parts of the synth as the other input. Verify that the audio signal being sent to the amplifiers is viewable on the oscilloscope connected to the output of the VCA. b) Change the control voltage with the power supply. Check that as voltage increases the amplitude on the oscilloscope increases and as voltage decreases the amplitude decreases.
6. The low frequency oscillator will generate a triangle wave from about 1Hz to 20Hz. It's output will be in the range from -2V to 2V. It can be used to modulate other parameters of the synthesizer in varying amounts.	a) Verify that the waveform is a triangle wave with the oscilloscope and that its frequency changes as the potentiometer is changed. b) Connect the synthesizer output to a speaker. Verify that for each knob (volume, pitch, filter cutoff, square wave duty cycle), turning it increases the modulation of that

	particular sound. Also check with the the oscilloscope.

Power Subsystem

Requirement	Verification
1. Sending +12 V and -12 V to the Oscillator, Mixer, Voltage Controlled Amplifier, and Audio Output	a) Measure voltage drop across positive and ground and the drop across ground and negative 12V. Make sure they are both 12V.

2.4 Plots

Below is a simulation of the VCO (schematic in Fig 7.1). It produces a sawtooth wave, which can also be used to generate a square wave using a comparator. The duty cycle of the square wave can be controlled by setting the threshold voltage of the comparator. In this case it was set to $V_{cc}/3$, so it is low for ⅓ the cycle and high for ⅔ of it.

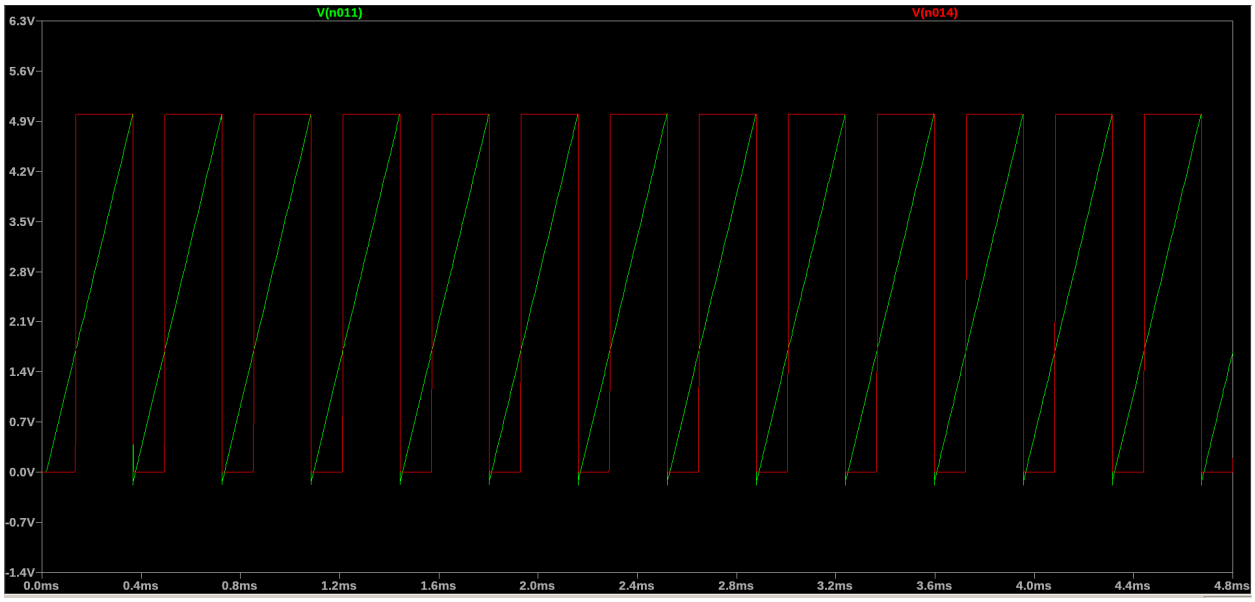


Figure 1: Sawtooth and Square Waves

2.5 Circuit Schematics

This design for this voltage-controlled oscillator was based on a lecture by Aaron Lanterman at Georgia Tech [4] and the book *Musical Applications of Microprocessors* [5]. It consists of an integrator with a constant current input to generate a ramp up. When a threshold of 5V is reached, a comparator turns on, turning on a JFET which allows the capacitor to discharge quickly back down to 0V. The input current to the integrator is created by a pair of BJTs. The current has an exponential relationship with the control voltage, which is ideal for musical applications.

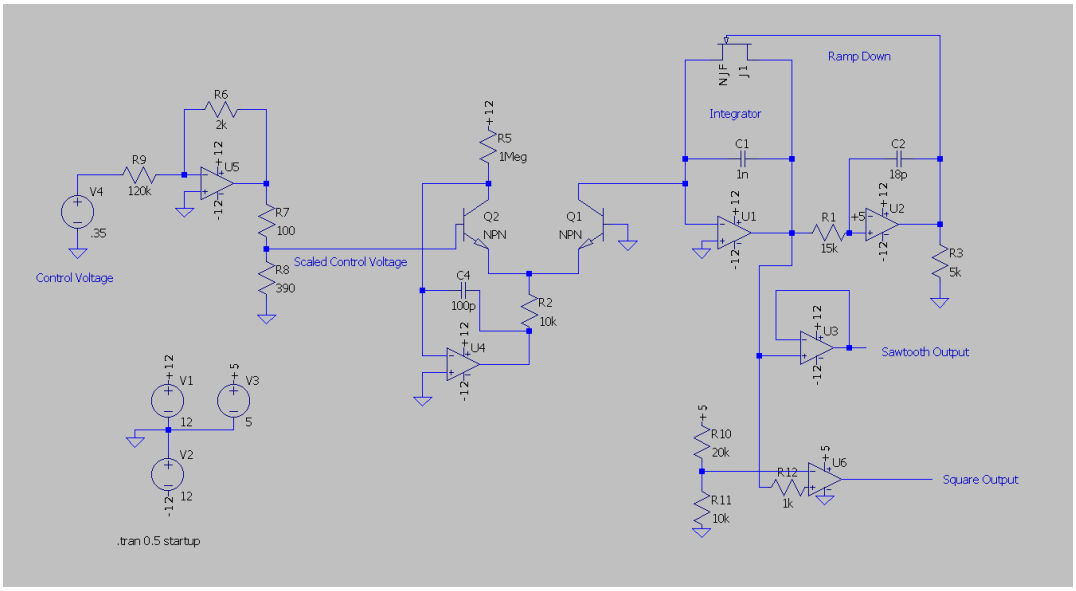


Figure 2: VCO

For the voltage-controlled filter, we will use the topology of the Moog ladder filter [3]. This was patented in 1969 and so the patent is now expired. The control voltage controls the bias current (using the same subcircuit as in the VCO) to all the transistors, which effectively changes their small-signal resistance, changing the cutoff of the filter. The output is differential, so a circuit to calculate the difference is used on the right side to get the output relative to ground.

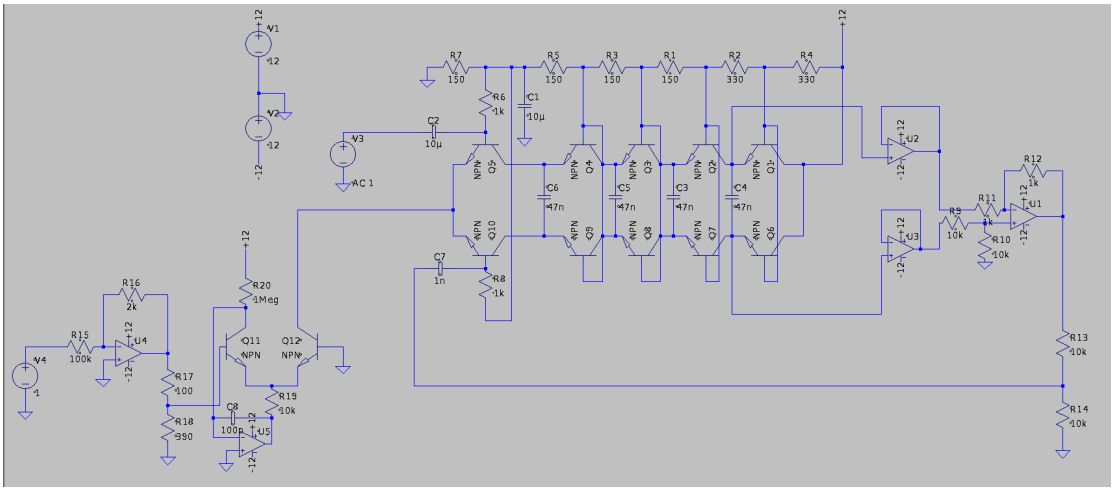


Figure 3: Moog Ladder Filter

The keyboard sends data to the synthesizer using MIDI. There is some work to convert the MIDI signal to something that can be used by the microcontroller, which in this case is UART. This is because the MIDI connector was designed to avoid ground loops and so the connector needs an optoisolator. This circuit was adapted from a Sparkfun article [6] which itself gets the it from the official MIDI specification, but it was modified to use a more modern optoisolator as the chip in the MIDI spec is no longer made.

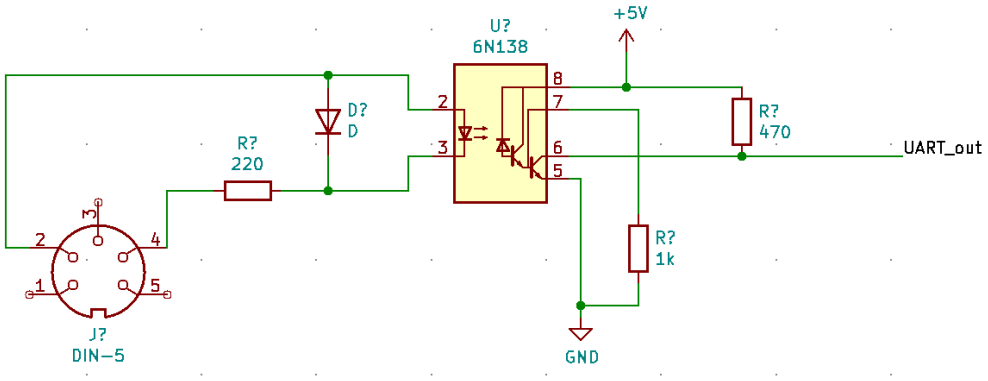


Figure 4: MIDI to UART

The microcontroller needs to be programmed, so we will include an ISP on the board. There will be a spot on the board to connect a USB programmer.

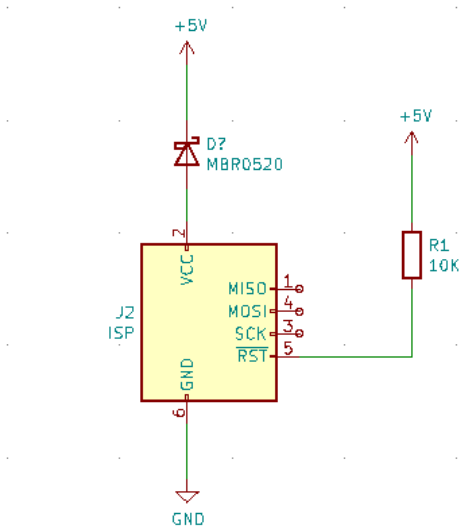


Figure 5: ISP connection

An amplifier adjustable with a volume knob. It goes to an audio output and to a built-in speaker. There is a switch on the speaker so that it can be disabled.

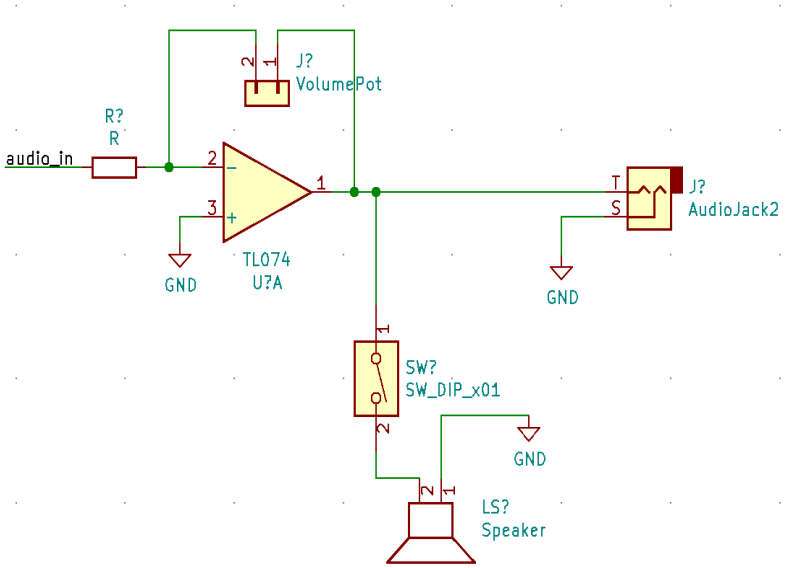


Figure 6: Speaker and Audio Output

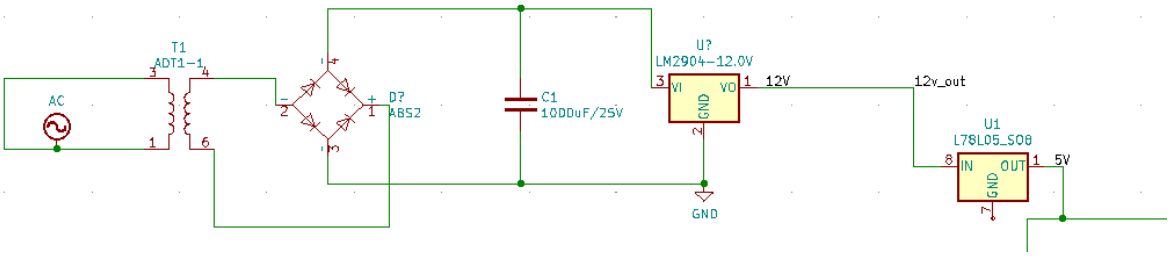


Figure 7: Power Supply

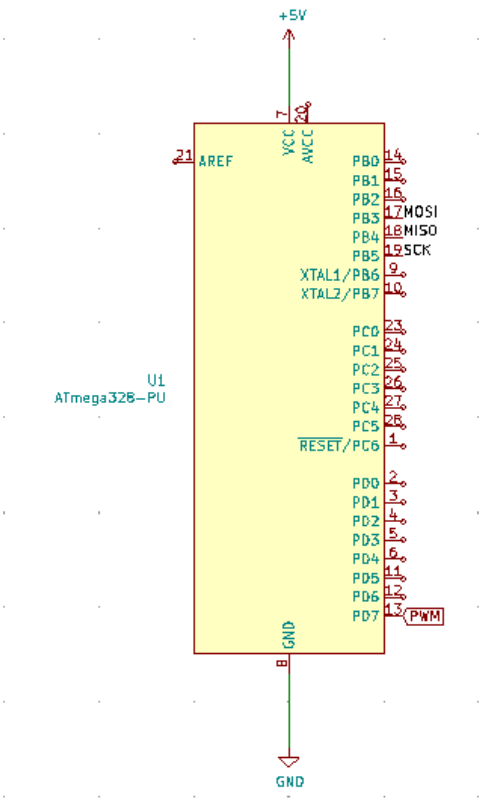


Figure 8: Atmega328P

2.6 Tolerance Analysis

The critical part of tolerance analysis is understanding that through each component of the subsystem, there is a threshold for what is required above or below the expected output and we are setting the accuracy tolerance to 0.5% on the frequency of the oscillators.

The target we are primarily trying to achieve would be sound of a particular frequency. It is very important that the frequency produced is as close to the real frequency as possible, or the synthesizer will sound out of tune, so this is where we will focus our tolerance analysis. We aim to get at least within 0.5% of the desired pitch. The frequency produced depends on the functioning of the synthesizer subsystem and the precision of the voltage outputted by the DAC and microcontroller.

We have two options to produce the voltage input for the voltage-controlled oscillator. We can either use the ATMega to communicate with some additional DAC chip, or we could use the PWM pins on the ATMega and filter them to create a constant voltage. We chose to use the PWM pins as this is simpler and lower cost. Pins 4 and 13 of the ATMega support PWM at 980HZ, which should be fast enough that the filtered voltage stabilizes quickly enough for key presses [8]. The ATMega PWM supports 256 different duty cycles, which means we can get voltage outputs with a resolution of 5V/256 = 0.0195V = 19.5mV.

It is important that the voltage-controlled oscillator produces a frequency with an exponential relationship to the voltage so that equally spaced voltages can map to the pitches of keys. We will choose a voltage to frequency relationship where 2.5V maps to 440Hz (the A4 note) and every increase/decrease by 1.2V will result in a doubling/halving of the frequency. This way a half step corresponds with 100mV changes. As shown above, we can get a resolution of 19.5mV, so we can get to the nearest 5*19.5mV = 97.7mV, which is very close to the desired 100mV. Now we will check how close the frequencies can be.

Frequency Relationship -

$2.5\,V\ produces\ 440Hz\ and\ every\ increase\ or\ decrease\ by\ 1.2V\ will\ double\ or\ half\ the\ frequency$

$$Frequency(V) = 440 * 2^{(V-2.5)/1.2}$$

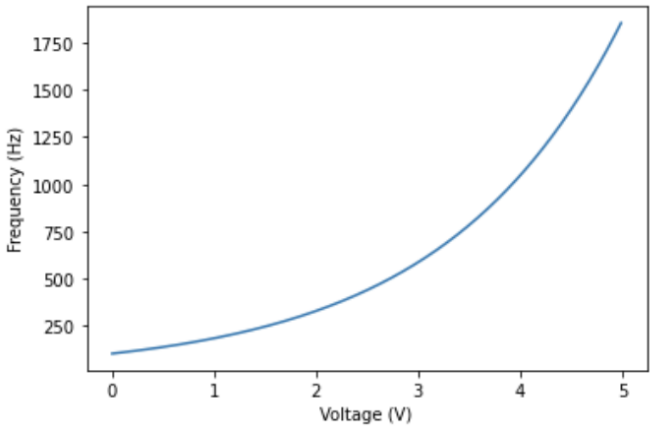


Figure 8.1: Frequency vs Voltage V that can be produced by microcontroller

The graph above has all the frequencies that can be produced by our microcontroller. We calculated our error for each musical note based on the difference between its frequency and the frequency we can produce with our microcontroller. The code for that is shown below.

```
from math import log2
a3 = 220
root = 2**(1/12)

freq = a3

print("freq\tvreal\tvclose\tfreqclose\terror (%)")

for i in range(24):
    vreal = 2.5 + 1.2*log2(freq/440.0)
    x = round(256/5*vreal)
    vclose = 5*x/256

    freqclose = 440*2**((vclose-2.5)/1.2)
    error = (freqclose-freq)/freq * 100

    print("%.3f\t%.3f\t%.3f\t%.3f\t%+2.3f" % (freq, vreal, vclose, freqclose, error)
)
    freq *= root
```

`vreal` is the actual input voltage that can produce the desired frequency. `vclose` is the closest voltage that can be produced through the analogWrite function. `freqclose` is the frequency we get using the microcontroller's voltage, and `error` is the percent error of the frequency.

freq	vreal	vclose	freqclose	error (%)
220.000	1.300	1.309	221.095	+0.498
233.082	1.400	1.406	233.925	+0.362
246.942	1.500	1.504	247.499	+0.226
261.626	1.600	1.602	261.862	+0.090
277.183	1.700	1.699	277.058	-0.045
293.665	1.800	1.797	293.135	-0.180
311.127	1.900	1.895	310.146	-0.315
329.628	2.000	1.992	328.143	-0.450
349.228	2.100	2.109	351.125	+0.543
369.994	2.200	2.207	371.500	+0.407
391.995	2.300	2.305	393.058	+0.271
415.305	2.400	2.402	415.867	+0.135
440.000	2.500	2.500	440.000	-0.000
466.164	2.600	2.598	465.533	-0.135
493.883	2.700	2.695	492.548	-0.270
523.251	2.800	2.793	521.130	-0.405
554.365	2.900	2.891	551.371	-0.540
587.330	3.000	3.008	589.986	+0.452
622.254	3.100	3.105	624.223	+0.316
659.255	3.200	3.203	660.446	+0.181
698.456	3.300	3.301	698.772	+0.045
739.989	3.400	3.398	739.321	-0.090
783.991	3.500	3.496	782.224	-0.225
830.609	3.600	3.594	827.616	-0.360

The error is less than 0.5%, our goal, for most pitches but it exceeds it for a few. This isn't too bad, but it also isn't ideal. We will improve this by tuning the VCO to have a slightly lower volts/octave, closer to the 97.7mV/octave, as this matches better what the microcontroller can produce. It should not be too hard to make this adjustment because the scale is determined by resistor values. We will add trimpots to the circuit that allow for manual adjustment, and we will tune it after it is built.

3. Cost and Schedule

3.1 Cost Analysis

In order to analyze the fixed labor costs for having a three person team, it was estimated based off of an average UIUC ECE graduate makes per year in 2021. According to The Grainger College for Engineering ECE graduates in 2018-2019 starting salaries average $91,781/year [2] based on 40 hour work weeks on average. Breaking this down into hourly wage of $44.16/ hour on average, and the expectation for the team will be that each person commits at least 10 hours a week to this course.

$$3\ team\ members\ \times (\$44.16/hour) \times\ 10\ hour/week\ x\ 16\ weeks$$ x 2.5 = $52,992

Electrical Components

Part Number	Description	Manufacturer	Quantity	Unit Cost	Site
652-PDB181 K420K104A2	Potentiometers 16mm Rotary Panel Potentiometer	Bourns	17	$0.90	Mouser
595-TL084HI PWR	Operational Amplifiers - Op Amps Quad-channel, cost-optimized	Texas Instruments	25	$0.474	Mouser
485-1314	Adafruit Accessories Speaker 3"	Adafruit	1	$1.95	Mouser
556-ATMEG A328P-PU	ATMEGA328P-PU 8-bit Microcontrollers	Microchip Technology	1	$2.60	Mouser
638-6N138	6N138 Optoisolator	Everlight	1	$1.02	Mouser

485-1134	Adafruit Accessories Breadboard Friendly MIDI Jack 5-pin DIN	Adafruit	1	$1.75	Mouser
CFR-25JB-52-68R	68 Ohm resistor	Digikey	10	$0.67	
	Assorted resistors, capacitors, connectors, diodes	Digikey, mouser		$20.00	
511-L7812ABV	12V Voltage Regulator	STMicroelectronics	1	$0.64	Mouser
771-MJD2873J	Bipolar Transistors	Nexperia	15	$0.399	Mouser
AS78L05RTR-E1	5V Voltage Regulator	Diodes Incorporated	1	$0.41	Digikey

The price of all the parts comes out to about $68.21, which is within our budget. This does not include the cost of a MIDI keyboard.

3.2 Schedule

	Breanne	Michael	Yash
9/27	Finish Design Document and get it uploaded onto the webboard. Finish the PCB initial design	Finish Design Document and get it uploaded onto the webboard. Finish the PCB initial design	Finish Design Document and get it uploaded onto the webboard. Finish the PCB initial design
10/4	Finish cost analysis and midi subsystem schematic and design with necessary parts needed.	Finish design of synthesizer subsystem schematic design and design with necessary parts needed.	Create a gitlab for the team to use for the microcontroller, as well as work toward designing schematic for the midi subsystem.

10/11	Awaiting PCB arrival	Awaiting PCB arrival	Awaiting PCB arrival
10/18 [parts arrive]	Co-create midi controller connection with chip and begin audio output design	Begin building out first section of synthesizer	Finish keycode assignments for each midi keyboard and co-create midi controller
10/25[PCB second order]	Go through second phase of design to see if another PCB order is needed	Go through second phase of design to see if another PCB order is needed	Go through second phase of design to see if another PCB order is needed
11/1	Begin building the audio output and continue to create test cases for midi controller code, and test outputs	Begin Soldering PCB, and building out schematic for synthesizer module	Begin Soldering PCB as well as code for the Atmega microcontroller
11/8	Test audio output functionality	Solder Box that the front panel will be on and continue building out synthesizer module	Test keys on midi keyboard that correct voltage is being outputted
11/15	Mock Demo	Mock Demo	Mock Demo
11/22	Fall Break	Fall Break	Fall Break
11/29	Prepare for the final demonstration and finish up writing the final paper.	Prepare for the final demonstration and finish up writing the final paper.	Prepare for the final demonstration and finish up writing the final paper.

4. Ethics and Safety

After reviewing IEEE Code of Ethics document, section 7, we do not perceive there to be a privacy risk of data or information that is stated in the first statement of the IEEE Code of Ethics. There will not be any user specific data that will be stored in the current design, so that the

safety of user data will be protected. In section 1.5 the ethics document specifies that it is a professional's opportunity to acknowledge and correct errors [1]. As a team we will uphold this by routinely evaluating roadblocks that occur within the project and remaining honest if there are flaws in implementation. Also, the project will uphold the code detailed in the ACM Code of Ethics and Professional Conduct [7]. Specifically, ACM states in Section 1 that computing professionals must honor confidentiality. In order to uphold this, the project will not be containing any confidential or any patent application initially. In Section 2, Professional Responsibilities, professionals working should ensure that they are creating high quality work and communicating with either stakeholders or team for transparency. This is important because it will ensure that awareness of potential consequences is understood, and through this project the team's responsibilities will be to communicate effectively with TA and professors and the rest of the team to mitigate any discrepancies. Furthermore, in section 2.9 of ACM, it is the professionals responsibility to design and implement solutions that are robustly secure. The responsibility aligns with continuously patching and reporting when there could be a security breach. The music synthesizer will not be connected to any public interface, for example public web facing applications which will keep the device secure from third party threats.

Also, in regard to ethics at a design level, many books and other resources exist for our project to explain the circuits in synthesizers, and our project has been implemented before. Schematics of many old synthesizers can be found easily online. Some particular circuits have also been patented, though most of these patents are old and have expired, such as the patent for the Moog ladder filter [3]. If we use any designs from some reference material, patent or schematic, we will need to first make sure that we can legally use it and then reference where it came from. In regard to outside factors that may be affected by the project, the synthesizer produces sounds that could be used to play loud noises of any frequency. For example by connecting the audio output to large speakers and the user can potentially cause a lot of noise pollution. We plan to prevent this by ensuring that we are testing the audio output as we build the circuit to ensure that the volume on the speaker will be at an acceptable level.

5. Citations

[1] "IEEE Code of Ethics", ieee.org, 2016. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 15- Sep- 2021].

[2] Grainger College of Engineering, “2018-2019 Illini Success Survey Outcomes* UNDERGRADUATE STUDENTS,” https://ecs.engineering.illinois.edu/files/2020/04/UG-ECE-2018-2019.pdf. [Online]. Available: https://ecs.engineering.illinois.edu/files/2020/04/UG-ECE-2018-2019.pdf. [Accessed: 22-Sep-2021].

[3] R. A. Moog, “ Electronic high-pass and low-pass filters employing the base to emitter diode resistance of bipolar transistors,” United States Patent 3475623A, Oct. 28, 1969.

[4] A. Lanterman, “Exponential Voltage-to-Current Conversion & Tempco Resistors,” in *ECE4450 (Analog Circuits for Music)*, 5-Apr-2021.

[5] H. Chamberlin and H. Chamberlin, “Basic Analog Modules,” in *Musical applications of microprocessors*, Indianapolis, Indiana: Hayden, 1987, pp. 177–220.

[6] B. J, “ MIDI Tutorial ,” MIDI tutorial. [Online]. Available: https://learn.sparkfun.com/tutorials/midi-tutorial/hardware--electronic-implementation. [Accessed: 27-Sep-2021].

[7] “The code affirms an obligation of computing professionals to use their skills for the benefit of society.,” Code of Ethics. [Online]. Available: https://www.acm.org/code-of-ethics. [Accessed: 15-Sep-2021].

[8] “AnalogWrite,” analogWrite() - Arduino Reference. [Online]. Available: https://www.arduino.cc/reference/en/language/functions/analog-io/analogwrite/. [Accessed: 28-Sep-2021].

[9] Note names, MIDI numbers and frequencies. [Online]. Available: https://newt.phys.unsw.edu.au/jw/notes.html. [Accessed: 30-Sep-2021].

[10] Technavio, “Global music SYNTHESIZERS MARKET: 48% growth to emerge from North America DURING 2021-2025: Technavio,” Global Music Synthesizers Market | 48% Growth to emerge from North America During 2021-2025 | Technavio, 21-Apr-2021. [Online]. Available: https://www.prnewswire.com/news-releases/global-music-synthesizers-market--48-growth-to-em erge-from-north-america-during-2021-2025--technavio-301273258.html. [Accessed: 15-Sep-2021].