Monophonic Hybrid Analog/Digital Synthesizer

“NSynth”

Final Document

Department of Electrical Engineering and Computer Science

University of Central Florida

Dr. Lei Wei

Group 27

Anna Wegener (EE)

Magdalena Dobinda (CpE)

Sebastian Carrion (EE)

Devin Vanzant (CpE)

Table of Contents

Total Page Count: 128

1. Executive Summary

A synthesizer, typically referred to as a “synth” by those in the music world, is an electronic hybrid musical instrument—think of it as a piano that is endlessly configurable to create almost any sound imaginable thanks to its internal analog circuitry. Synthesizers generate audio signals by generating waveforms through various popular methods known as subtractive synthesis, additive synthesis, and frequency modulation synthesis. These methods of synthesis will be discussed in more detail in Chapter 2, Section 5. Synthesizers allow the user to manipulate the three core elements that define sound: pitch, timbre, and loudness. These parameters can be controlled by devices known respectively as oscillators, filters, and amplifiers—the three essential elements of any synthesizer.

Since their introduction, synthesizers have been used in nearly every genre of music. They first rose to mainstream use in 80s pop music and are still utilized heavily to this day in genres such as pop, hip-hop, and electronic music. Their use has even birthed countless subgenres of electronic music such as house, techno, dubstep, synthwave, drum & bass, and electro; all of which are usually referred to under the umbrella term “EDM”, which stands for Electronic Dance Music.

At a high level, a synthesizer contains audio signal-processing modules and some form of input, the most popular being a piano-like keyboard. At the heart of our synthesizer, is the oscillator – the heart of the synthesizer – producing raw tone generation. At the minimum, a synthesizer contains one oscillator for tone generation, however, additional oscillators allow for more nuanced and complex “layered” sounds to be generated. Oscillators can produce a handful of basic wave shapes such as sine, square, triangle, and saw that when combined, produce waveforms unachievable through a single oscillator. Further manipulation of these waveforms can be achieved through various audio signal processing components such as filters, low-frequency oscillators, and amplifiers. The user is able to control each of these components through a variety of input sources such as buttons, switches, knobs, or physically connecting cables.

Synthesizers can take on several types: analog, digital, and hybrid analog/digital. On top of this, their form factor can be categorized into normalized, modular, or semi-modular. Normalized synthesizers are completely self-contained in one enclosure and the user interacts via switches, knobs, potentiometers, etc.

Whereas modular synthesizers are composed of separate modules which each have a different function. A semi-modular synthesizer takes on elements from both form factors—being connectable through cables just like a modular synthesizer, but these individual modules are hardwired and unable to be swapped out.

As a team of two electrical and two computer engineers, we all share the same passion for electronics and music. We found this idea of an analog synthesizer to be the perfect opportunity for our skill sets and interests. For hardware, there was analog circuitry and signal analysis techniques involved. For software, there were a number of standard protocols to be implemented through embedded C programming.

This project serves as a medium for both expertises to shine and meet in the middle for an end product we can interact with and see both our studies and interests unite. Our senior design project was a learning opportunity for all of us in the field of audio engineering that we hope to continue building upon after graduation as a lifelong hobby.

2. Project Description

In this section we discuss the most prevalent motivation behind this project, and how each of us were able to play a role in the process of researching, designing, and completing this project. The next section discusses important synthesizer fundamentals. This is a quick review of the fundamentals to help the reader understand the parts of the project better, as it’s a somewhat niche topic in the music/engineering industry and these fundamentals are instrumental to understand before attempting to design a synthesizer of any sort.

2.1. Motivation

This project served as a bridge between engineering, computer science, music production, and performance. By utilizing the MIDI protocol, our analog synthesizer is playable by any device with MIDI-out capabilities–whether this be a MIDI keyboard or a digital audio workstation (DAW) with external instrument functionality such as Ableton Live, FL Studio, or Pro Tools. This project implemented our knowledge of embedded systems, linear circuits, electronics, circuit design, and computer science courses as a team.

We created this project as a bridge between engineering and music. This project allowed us to have full control over the type of controller we built and how it interacted with the channels, notes, and software. This synthesizer connected a bridge between software and hardware by using a MIDI channel web API, this way we can evolve the software to create niche functionality some artists have a real need for, such as a visualizer controlled by MIDI commands. Most artists do not really have control over software like that, but we can make the software easy to customize and create awesome visuals using the sequences/bpm. Overall, this was a learning experience which honed our skills that we have acquired over the years. This project directly implemented our knowledge learned from embedded systems, linear circuits, electronics, circuit design, and computer science courses as a team.

The motivation behind this project was to create an analog synthesizer that uses voltage-controlled circuitry where the voltage is directly proportional to the oscillator waveform's pitch (i.e. frequency). The analog synthesizer by nature generates continuous waveforms crossing infinite points which makes the sound much more pure compared to a digital synthesizer which discretizes the waveform and is limited to a number of samples. Hybrid synthesizers also exist to get the best of both worlds. These feature a digitally controlled oscillator and analog audio processing to avoid common environmental impacts on the delicate VCO circuitry like temperature and humidity of the room. These imperfections however add texture and an endless plethora of sounds to be created, hence why we are choosing to build an analog synth for this project

2.2. Synthesizer Fundamentals

It is important to understand the elements of a synthesizer and their applications in music production. Below we will be highlighting the basic terminology used throughout synthesis that is essential to understanding our project methodology and design.

2.2.1. Oscillator

[1] Oscillators are the heart of any synthesizer–they are used to generate an audio waveform that can then be further processed by other components of the synthesizer.

The standard waveforms that can be found in most analog synthesizers include: sawtooth, square, triangle, and sine. The sine wave is the most basic waveform as it contains a single fundamental frequency and no additional harmonics or overtones. Other waveforms contain harmonics or overtones that give them each their own distinctive sound or timbre.

2.2.2. Filter

Next in the audio signal processing chain is the filter. The waveform output from the oscillator is used as the input for the filter – whether it be lowpass, bandpass, or highpass. Filtering allows the user to shape said waveform's timbre and tone by removing certain frequency harmonics. Depending on the filter technique, the musical tone heard can sound warmer, brighter, or harsher.

Musicians interact with the filter with two main parameters: cutoff and resonance. Adjusting the cutoff point dictates at which frequency the harmonics will be attenuated. Resonance, also known as Q factor, can be controlled to amplify harmonics at cutoff point on the frequency scale to intensify certain timbres.

2.2.3. Envelope Generator (ADSR)

Envelope generators, also known as the Attack Sustain Decay Release (ADSR) module, are used to manipulate the entire sound envelope. In synthesis, an envelope refers to the beginning, middle and end portions of sound heard while a key is depressed. A knob is usually used to control each element of ADSR, each providing a different voltage. In this context – applied to the amplifier, each knob represents how high or low the voltage will be at different points in the envelope such that the user can define how loud or quiet the sound gets throughout the envelope.

Attack controls how long the sound takes to reach its maximum volume. Decay sets the rate at which the volume will decrease from max to zero or at the level dictated by sustain. The sustain knob controls where the envelope decays from max volume to hold at as long as the key is depressed. Once the key is released, the release knob controls how long until the volume drops to zero from the sustain or decay level.

2.2.4. Amplifier

Amplifiers are what the user controls the loudness of the sound through a volume knob. Technically speaking, the amplifier's role is to modulate the amplitude of the signal by either increasing or decreasing it. The higher the input voltage fed through to the amplifier, the more signal can pass through, resulting in a higher volume heard by the listener.

2.2.5. Voices

Having decided to build an analog hardware synth, our team had to consider how many “voices” the synth would be capable of. Defining the term “voice” can be ambiguous but most commonly refers to the ability to play chords or multiple keyboard notes simultaneously. A monophonic synthesizer can play only one note at a time while a polysynth can play multiple. While this may seem like a disadvantage it actually means the synthesizer has much more control on that one sound as .

2.2.6. MIDI Protocol

[2] MIDI is an acronym that stands for Musical Instrument Digital Interface. It’s a way to connect devices that make and control sound—such as synthesizers, samplers, and computers—so that they can communicate with each other, using MIDI messages. MIDI is the primary way to control a synth from an external device. The connection between the two serves as the internal CV/gate control signals. “” This allows one keyboard trigger sounds on another synthesizer, and it makes it possible to record music in a form that allows for easy note editing, flexible orchestration, and song arrangement. “” MIDI supports up to 16 channels of simultaneous sound. Each channel can have its own melody, rhythm, and instrument.

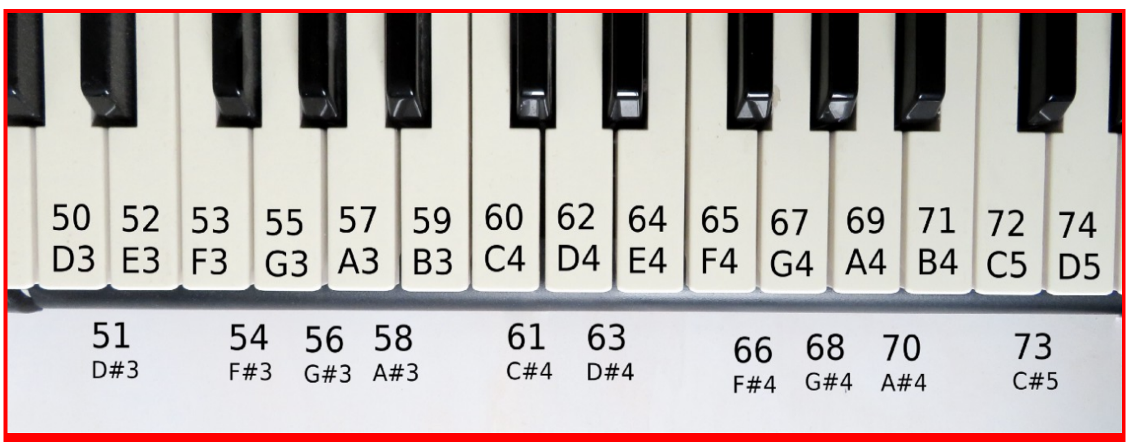


Figure 2.0: Keys and Numerical Representations

Figure 2.1: Formatting of MIDI Messaging

MIDI Messages

When you take a closer look as to how computers and software interprets MIDI messages, you can see that it's much of what we as computer and electrical engineers study. You'll find that MIDI messages are sometimes represented as binary numbers, sometimes as decimal numbers, and sometimes as hexadecimal numbers (hex, for short). Binary and hexadecimal are useful ways to represent MIDI messages because they allow us to divide the messages into meaningful groups.

MIDI messages can be divided into two main categories: Channel messages and System messages. Channel messages contain the channel number. They can be further subdivided into voice and mode messages. Voice messages include Note On, Note Off, Polyphonic Key Pressure, Control Change, Program Change, Channel Pressure/Aftertouch, and Pitch Bend. System messages are sent to the whole system rather than a particular channel. They can be subdivided into Real Time, Common, and System Exclusive messages (SysEx). For this project, we will be limiting our focus to mainly channel messages.

The concept of channels is central to how most MIDI messages work. A channel is an independent path over which messages travel to their destination. There are 16 channels per MIDI device. A track in your sequencer program plays one instrument over a single channel. The MIDI messages in the track find their way to the instrument over that channel.

2.3. Block Diagrams and Designated Roles

In this subsection we will show a block diagram and the roles designated to each member of the team. The first block diagram shown is a high-level overview of the synthesizer which shows the chain of the audio signal path. The second block diagram is a more detailed system-level view of the sound processing modules integrated with software and human interface devices. Both diagrams are shown to show the core of the synthesizer first before introducing software features and external devices.

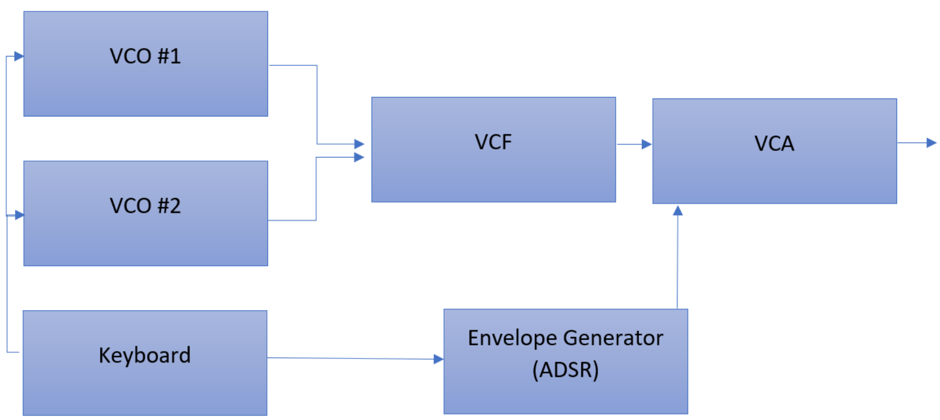


Figure 2.2: High-Level Block Diagram

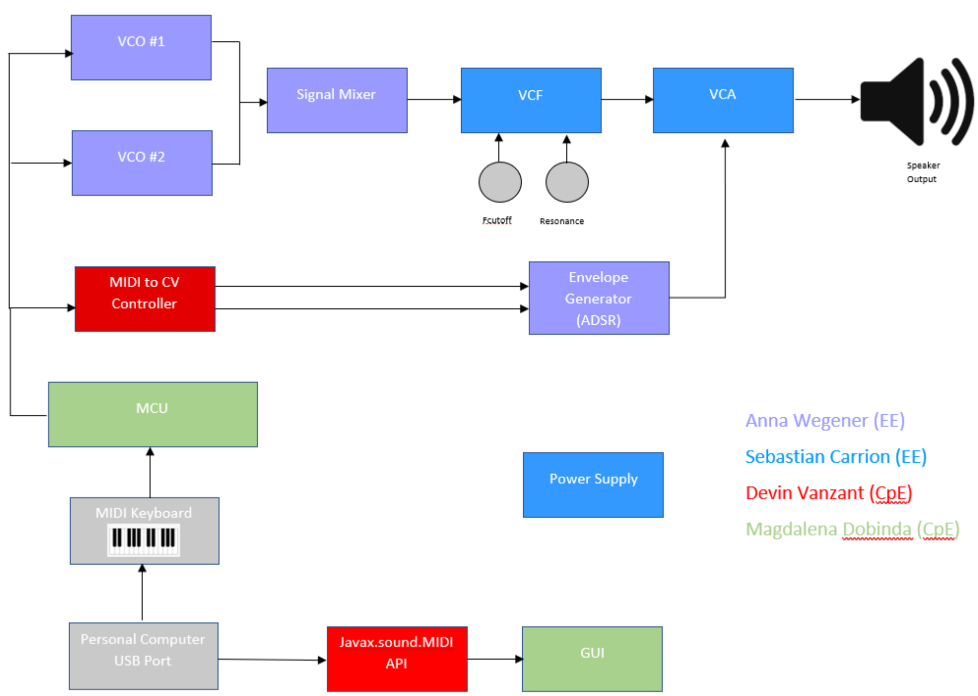


Figure 2.3: Detailed System-Level Block Diagram with each members focus

2.4. History of Synthesizers

In this section, we will discuss the history of synthesizers, as well as the math behind music theory and how electrical and computer engineering plays a large part in our project.

The First Synth Ever – The RCA Mark II

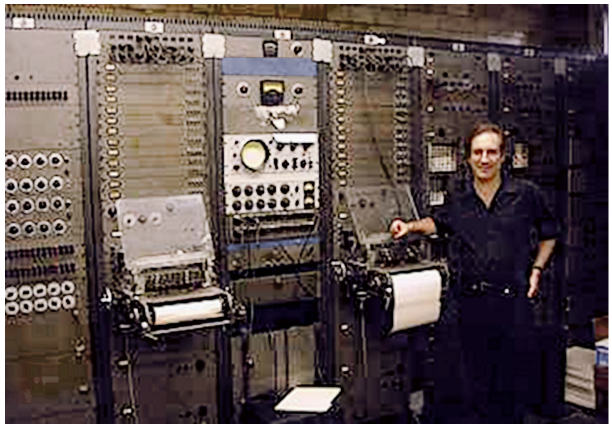


Figure 2.4: RCA Mark II

[3] The first synthesizer – the RCA Mark I – was created by Radio Corporation of American engineers Harry Olson and Herbert Belar in 1952. While you likely imagine a synthesizer having a keyboard and a bunch of knobs, the RCA Mark I was a giant machine consisting of many bulky vacuum tubes that took up an entire room at Columbia University… and instead of being played using a keyboard, it was tediously programmed using paper punch cards. Sound was generated by twelve sinusoidal oscillators via electromagnetically stimulated tuning forks. Changes in timbre and volume of the sound processing modules were done by the paper punch cards making it impossible to play and hear the output simultaneously.

The First Voltage Controlled Synthesizer – The Moog Modular Synthesizer

Figure 2.5: Moog Modular Synthesizer

Key to the development of synthesizers was the birth of the transistor–born at Bell Laboratories in 1947 by Walter H. Brittain, John Bardeen, and William Schockley. The new electrical component was the bread and butter for voltage-controlled synthesizers. Each module operated off an increase or decrease in voltage to control changes in sound. Enter Robert A. Moog, known as the father of synthesizers. Though not the inventor of the voltage control concept, he established the famous exponential 1 volt-per-octave standard and trigger signals that are still used today. He is responsible for the rapid popularization of the modular system in 1967.

After realizing several successful units, he quickly realized it was too big and complicated to be an appealing product for the music retail market. In order to expand his reach to performing musicians, he created a portable and affordable normalized synthesizer known as the Minimoog Model D. Though not as flexible, the convenience of an integrated keyboard, pitch and modulation wheels, and hard-wired signal flow sold over twelve thousand units.

The Math Behind Synthesis

[4] Music and mathematical theory go hand in hand, and as engineers was one of the reasons we were so intrigued with a project that had both. Here we will provide a quick mathematical overview behind music and synthesizers.

Sound as a concept: Sound travels through air as a series of concentric rings of high and low pressure. If you measure the pressure along a radius starting from the outside edge and going in, or if you pick a point and measure the pressure with *respect to time*, we get a function which, in the case of music, has a lot of periodic components.

We consider the ***sine wave*** to be the basic "unit" of musical sound since any sound can be decomposed into sound waves (fourier/laplace series/transforms) much in the same way as a taylor series.

Sinusoids, amplitude and frequency: Electronic music is usually made using a computer, by synthesizing or processing digital audio signals. These are sequences of numbers,

$..., x[n-1], x[n], x[n+1],....$

Where the index n, may range over some or all the integers. A very common example of a digital audio signal is the *Sinusoid:*

$$x[n] = acos(\omega n + \theta)$$

Where a is the amplitude, ω is the angular frequency, θ and is the initial phase.

Sinusoids play a very important role in audio processing. If you shift one of them left or right by any number of samples, you get another one. This makes it easy to calculate the effect of all sorts of operations on sinusoids. Our ears use this same special property to help us parse incoming sounds, which is why sinusoids, and combinations of sinusoids, can be used to achieve many musical effects.

The Overtone Series: You have most likely heard this term thrown around with "harmonic series", but unless seriously studied we do not realize what an important role this plays into tonality and sound production. The overtone series occurs naturally in all non-synthetic tone production.

In mathematical terms, the harmonic series is the infinite series formed by summing all positive unit fractions:

$$\sum_{n=1}^{\infty} \frac{1}{n} = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \cdots .$$

Figure 2.6: Harmonic Series

The first n terms of the series sum to approximately $ln\, n\ +\ \gamma$, where ln is the natural logarithm and γ is the Euler-Mascheroni constant.

The division of the octave into twelve parts is our brains' interpretations of a simple mathematical phenomenon. When the frequency of a soundwave doubles, our brains hear those two frequencies as sharing some fundamental commonality, so it interprets those two pitches as the "same" but separated by an octave. Therefore, octaves always have a 2:1 ratio. The next two simplest ratios are a 3:2 ratio and a 4:3 ratio, which create a perfect 5th and a perfect 4th respectively.

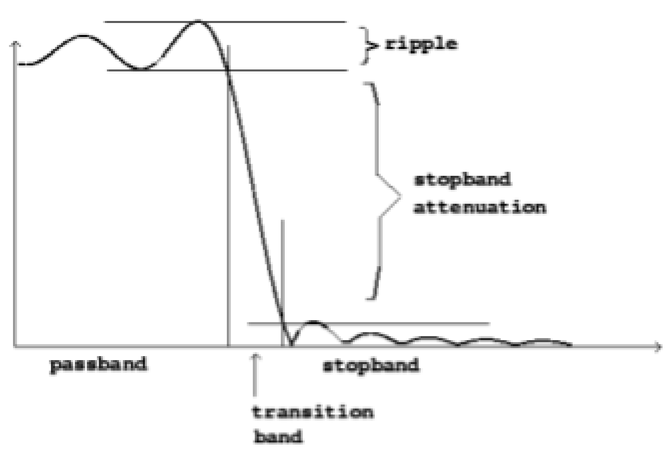


Figure 2.7: Transition Band Illustration

Filters: Filters can generally be identified as low-pass and high-pass filters in music. The most frequent purpose for using a filter is extracting the low-frequency or high-frequency portion of an audio signal, attenuating the rest. This is accomplished using these filters.

We see in the figure below that this is the frequency response of a low-pass filter. On the horizontal axis, frequency is divided into three bands. The passband is the region where the filter should pass its input through to its output with some unit gain. For this specific filter, the passband reaches up to a certain frequency limit. For a high-pass filter however, the frequency would go up to the highest frequency possible. The passband's deviation from flatness is called the ripple, often specified by giving the ratio of the highest and lowest gain in the passband expressed in decibels. The ideal low-pass or high-pass filter would have a ripple of 0 dB.

The Rise of Performance Synthesizers

The Minimoog Model D was the beginning of a domino effect as other famous manufacturers including ARP, Roland, Korg, and Yamaha entered the market. Japan had a strong holding in the market. Korg released their first synthesizer known as the MiniKorg. This was a simple monophonic synth that stood out for its three ring modulators and its quirky front facing control interface design. ARP released their iconic semi-modular ARP 2600 known for its internally-wired modules with the ability to be rerouted using patch cables – making it ideal for performance and teaching.

Roland released their first synth known as the SH-1000. This was a monophonic, single oscillator synth that more so resembled an organ with ten programmable voices. Yamaha delivered the SY-1, a preset loaded synth famous for its Attack/Bend features and velocity sensitive keyboard. Though the list goes on, these are just several of the first synth manufacturers that marked the beginning of an electronic music era.

Evolution of Digital Synthesizers

In 1978, the iconic American Engineers Dave Smith and John Bowen of Sequential Circuits created the polyphonic Prophet-5. This was the breakthrough into the digital world as it combined analog oscillator ICs and digital microprocessor memory for presets. Soon thereafter, the 80s was the beginning of a new chapter of synthesizers. In 1982, Yamaha launched the first commercially available digital synth known as the DX7. This was a monumental moment for the field and it soon became one of the bestselling in history and was widely used in pop music. The DX7 sounded nothing like its predecessor analog synthesizers and instead sounded glassy and bright with little input required. It was a feature packed synth, offering dozens of patches and pre sampled tones via frequency modulation. Alongside the DX7 were other digital synthesizers famous for their own unique features like the Fairlight CMI's digital sampling technology and the Synclavier's pure digital oscillators. Renowned synthesizers like the Prophet and Jupiter by Roland were rereleased as the first MIDI instruments. This was a time of pioneering as the digital territory was completely untapped until now.

Figure 2.8: Yamaha DX7

Modern Day

The digital technology takeover was here to stay as analog sales went down due to their cost and practicality. The 90s was yet another turning point for the industry as software instruments emerged. Software instruments sought to emulate hardware synthesizers as what is known as "plug-ins" to be used on Digital Audio Workstations (DAWs). First to do so was Cubase, developed by Steinberg. As analog synthesizers fell out of production and into collector's studios, the most popular such as Moog's MiniMoog, Sequential Circuit's Prophet-5, Oberheim's OB-X, and Roland's Jupiter-8 were emulated as software instruments. These Virtual Studio Technologies (VSTs) provide us access to the most iconic synthesizers in history as they replicate their exact knobs and controls. Nowadays, other DAWs such as Ableton Live, Logic Pro, and FL Studio dominate the market and provide us with a means of endless analog, digital, and hybrid synthesizers.

In this section, we will go over the most common forms of synthesis and how they are relevant technology to this product.

2.5. Types of Synthesis

There are many different forms of synthesis, though the most common are subtractive, additive, wavetable, and frequency modulation (FM) synthesis.

Examples of other forms of synthesis found in digital synthesizers include: wavetable, frequency modulation (FM), granular, and additive synthesis.

2.5.1. Subtractive Synthesis

[5][6] Subtractive synthesis is the most common form of synthesis used in the first analog synthesizers. It is still widely used today due to its simplicity and signature sound. A subtractive synthesizer takes a harmonically rich waveform generated from the oscillator such as sawtooth wave and subtracts parts of the audio signal using its filter to change the timbre of the sound. In this technique, the most common methods of manipulating sound is through filtering and envelope modifications.

Subtractive synthesis assumes that an acoustic instrument can be approximated with a simple oscillator that can produce waveforms with different frequency spectrums. The filtered or unfiltered signal is shaped over time by the amplifier section of the synthesizer.

Overall, subtractive synthesizers aren't perfect at emulating real-world instruments, but most or really all synthesizer models follow a fundamentally similar architecture and signal flow that is based on subtractive synthesis. The true strength of subtractive synthesis is that they offer a truly unique, and adaptable sound palette.

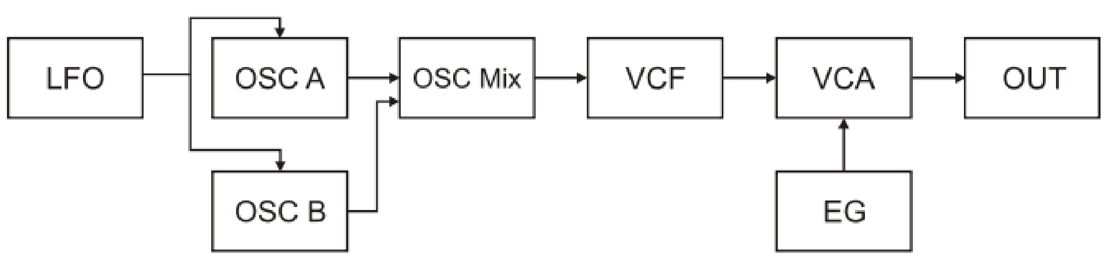


Figure 2.9: Subtractive Synthesis

2.5.2. Additive Synthesis

[7] In contrast, additive synthesis seeks to replicate the way sound is created in nature through sine waves. This involves summing the output of two or more sine wave-generating oscillators. It is based on the principle that any sound can be recreated by combining enough varying sine waves. Unlike subtractive synthesis, where the timbre of the sound is created by subtracting parts of the audio spectrum.

Depending on the sophistication of the additive synthesizer you are using, you will either have individual envelope control over the level and pitch of each sine wave, or you will be limited to envelope control over groups of sine waves. It's much more practical to work with groups of related harmonics due to the mathematical relationships between them and the impact it has on the overall tone.

2.5.3. FM Synthesis

Frequency Modulation (FM) synthesis is when the timbre of a simple waveform from the oscillator is changed by modulating its frequency against another waveform. This method can create both harmonic and inharmonic sounds by using the concept of carrier and modulator waves. The process begins with a pure sine wave (carrier) and is modulated against another inaudible sine wave (modulator) to create an endless array of sounds.

These are just the most common of the many types of audio synthesis techniques. In this project, we have chosen to utilize subtractive synthesis due to its commonality and compatibility with the analog world.

2.6. Goals

In this section we discuss the basic goals of the project, which we prioritized for our senior design 2 demonstration. The advanced goals are what we determined to be achievable within the timeframe. Stretch goals are, as stated, overachieving goals that did not fit our timeframe.

Basic Goals

- **Hardware:** The hardware of this device should be able to generate analog waveforms, which are produced by the VCO. Such waveforms should be able to be modulated with the VCF, VCA, and envelope generator. The bare minimum goal of this project is to be able to output an audio signal that can be modulated by these components and heard by the user.
- **Software:** The analog and digital components will interface with a MIDI to CV controller allowing for 1 V/Octave tracking. The synthesizer will be MIDI compatible through a USB and will be able to be interfaced to a computer.
- **Overall:** The baseline goal of the final product is that the device will be able to generate audio waveforms that can be transcribed back into audible sounds for the user to perceive as well as letting the user be able to display said waveforms within the computer application through a GUI.

Advanced Goals

- **Hardware:** The hardware will allow the user to see and hear the effects of their chosen setting parameters reflected on the signal's waveform.
- **Software:** GUI interface / App Interface will actively read MIDI input from the protocol and show the current waveforms, frequency, pitch, etc. being played. Interface should be able to show the type of waveforms being produced by the synthesizer.
- **Overall:** As we develop the device more, our goal is to enable the user to have more parameters to customize and in turn produce a wider variety of audio waveforms. These parameters will be tracked within the application on the GUI so that the user may see both the parameters affecting the generation of the waveform and the waveform itself.

Stretch Goals

- **Hardware:** The hardware would include a module that would enable the user to connect to the synthesizer via Bluetooth. The synthesizer will be able to recreate previously generated sounds through recalling previous user parameters.
- **Software:** Real time visualization of notes/pitches/etc being played with ability to digitally manipulate. Visualization would include waveforms as well as giving the user artistic expression through customizable visuals that are following the pitch, frequency and beat.
- **Overall:** The long term stretch goals would be real time visualization and playback of previously generated sounds.

2.7. Chosen Methodology

After learning the basics of synthesizers, we decided on which features we wanted to include in our senior design project. Ultimately, this came down to several factors: ease of implementation, time, cost, and division of labor. We took both antique and modern day synthesizers into comparison when deciding how to make an electronic instrument that can withstand the test of time in today's digital music production world. Preliminary design decisions for the synthesizer itself and its features will be explored below.

Analog vs Digital

Before deciding on which sound processing modules we wanted to include we had to decide on what type of synthesizer we wanted to build: analog, digital, or hybrid. Considering we are a group of two electrical engineers and two computer engineers, we figured a MIDI-compatible fully analog synthesizer would allow us both to exercise our areas of expertise in analog circuit design and programming. All circuitry can be prototyped on breadboard for testing and done on PCB making it ideal for a senior design project.

Monophonic vs Polyphonic

Our team had decided to go with a monophonic synthesizer over a polysynth for a number of reasons. [8] The main advantage is the hardware/budget constraints and limitations of a polysynth. Application is critical in deciding which is better over the other. Monophonic synthesizers are a natural fit for bass duties—solos and portamento (the gliding or slurring between notes) are more common on monos than polys. The monophonic synthesizer is considered “vintage” in the music industry and a very rare piece of hardware to come by. Mono synthesizers sound “bigger” because they can be designed to be more harmonically rich, since the synth doesn’t have to accommodate multiple voices at once.

Building a monophonic analog synth was our all-encompassing goal for Senior Design, however what is a synthesizer instrument without being able to play it? A keyboard is a necessary part of the project. Piano-like keyboards are commonly seen in commercial synthesizers today. In this project, the user can either play the synth with an external keyboard or through their PC keyboard with a Digital Audio Workstation (DAW). How a fully analog synthesizer can be controlled with a keyboard and interfaced with a DAW is what is known as MIDI protocol.

Oscillator

Seen in almost every analog synthesizer is an oscillator, however, a single tone on its own does not sound particularly interesting to most listeners. It was decided to include at least two VCOs to add versatility and depth to the sound. Incorporating two VCOs more than doubles the potential tones and essentially allows you to play chords even with a monophonic synth. This allows two oscillators playing the same note but slightly out of tune for what musicians call a “fatter” sound. Having multiple oscillators also allows for combining different waveshapes to produce unique sound spectra that would not be possible otherwise with a single oscillator. Though sawtooth, sine, square, triangle, and noise waves are all possible, it was decided to go with sawtooth and triangle waveforms. Sawtooth waveforms are harmonically rich for a fuller sound and triangle waveforms offer a similar sound to sine without the waveshaping complexity. A mixer was built to combine the two oscillator outputs into a single signal to be fed into the sound processing modules.

Filter

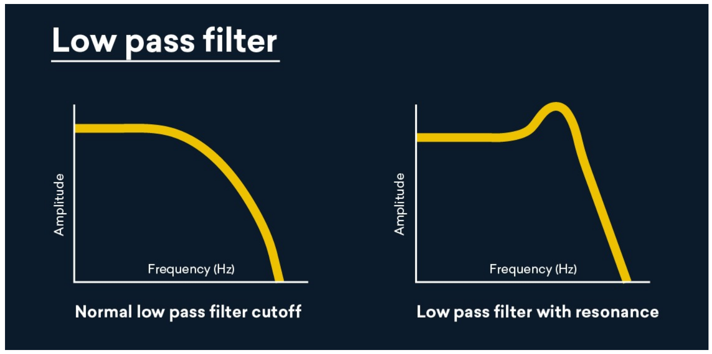


Figure 2.10: Low Pass Filter

After the two oscillator waveforms are summed together, the analog signal path continues to the filter next. Most of the interesting timbre changes in subtractive synthesis come from the Voltage Controlled Filter (VCF). Recall that we are starting with a harmonically rich waveform and seeking to cut off certain frequencies. Because we want to attenuate high frequency harmonics and make darker, warmer sounds, we decided to use a low pass filter. The resonant low pass filter is the most common type of filter seen in analog synthesizers since it allows for the fundamental frequency to stay intact without affecting the pitch of the signal. An active filter topology will be used to add resonance functionality as well.

ADSR

Considering a VCO continuously generates waveforms to no end, envelope generators (also known as ADSR modules) are a vital component to synthesizers. Without them, synthesizers would only produce sounds that start and end immediately as you press and release keys. ADSR provides a means for waveform contour to add musical character. When making design considerations, attack and release time were among the most important. A short attack time will result in a near instantaneous sound heard while a long attack time will resemble a violin's gradual crescendo. The same can be said for the release segment as these are the two most obvious parts of a sound envelope heard. We contemplated leaving this module out but decided it was necessary to simulate a starting and stopping point to the oscillators.

Most synthesizers have one envelope generator for the VCA and one for the VCF. The difference lies in which module's envelope you want to control. Applying ADSR to the filter would allow you to exercise control over the cutoff frequency over time – making the sound more or less bright. Applying ADSR to the amplifier allows you to control the volume output generated by the oscillators. This allows the synthesist to create slow/fast attacks and long/short sustains and releases. We decided the user having the ability to shape the sound contour by applying ADSR to the amplifier was far more advantageous for making an interactive/intuitive analog synthesizer.

Amplifier

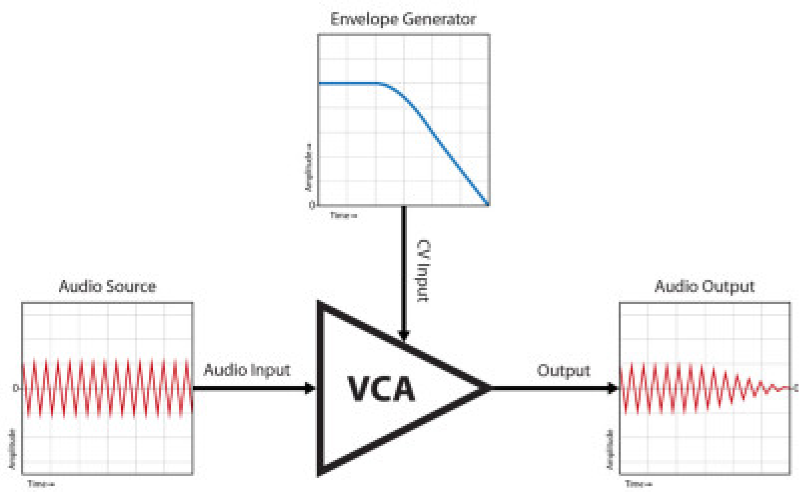


Figure 2.11: Amplifier Functionality with Envelope Generator

When evaluating the amplifier module of our synth, we had to first differentiate between those used in the audio signal amplitude change versus those used in modifying control voltages. In this case, we concluded that the VCA would be responsible for control voltage modifications. [9] The amplifier here can be thought of as a “multiplier” in that we are multiplying the signal’s amplitude by some value known as gain. Synthesizers generally deal with zero to unity gain values. In our project, the sequence of events would look like the following: the user presses a key on the keyboard generating a gate signal from the MIDI to CV converter, triggering the envelope generator outputting a CV signal to modulate the amplitude of the VCO’s signal then passing through the VCA.

In our common subtractive synthesis setup, the VCA creates dynamics in collaboration with our envelope generator. This way, just like hitting a key on a piano – the volume heard starts out very loud and fades away over time after letting go of the key. The sound envelope effects heard by the ear are controlled by the envelope generator, but applied by the VCA which is responsible for turning the signal on/off – turning them into separate notes with silences in between. With this effect in mind, we answered the debate of linear vs exponential VCAs and decided to use exponential for volume control due to the similarities in human hearing processing sound exponentially.

MIDI Abilities

Though building a fully analog synthesizer may seem like a blast from the past, we wanted our project to withstand the test of time and integrate into today's digital world, hence the hybrid title. Building a MIDI capable synth allows for musicians to seamlessly connect to common DAW production setups. With the audience in mind for this project being those who share an interest in electronics and making music, we wanted to satisfy those with both interests. Adding MIDI functionality allows for both us electrical and computer engineers to work together to build a vintage synthesizer with modern capabilities.

An example of what MIDI can add to our synthesizer is an arpeggiator. An arpeggiator is a device that takes a single note or multiple MIDI notes from a held chord and plays them in a certain rhythmic pattern – an arpeggio – as defined by the settings of the arpeggiator. It's a common feature found on most synthesizers built after the 90s, but first rose in popularity in the late 60s and early 70s on iconic vintage synthesizers such as the Roland Jupiter 8 and Korg Polysix. Since then, they have become an essential part of electronic music and are responsible for many of the melodies and basslines found in music today.

An example of how MIDI allows us to do this is through the built-in "Arpeggiator" MIDI effect found in the popular DAW Ableton Live. Not only is this a more straightforward approach compared to building one from scratch, it also allows for much more flexibility and rhythmic capabilities than we'd be able to build given our monetary & time constraints. Through Ableton Live's "Arpeggiator" effect, one is able to control the style of arpeggio (down, up/ down/up, up/down, converge, diverge, etc.), speed of note playback (either synced to the DAW's BPM or a chosen time in milliseconds), gate (duration of the notes played), groove or swing of the arpeggio (Ableton Live offers dozens of customizable "groove patterns"), as well as the ability to "hold" the last notes played (a feature common on most arpeggiators)

2.8. Requirement Specifications

In this section for our requirement specifications are introduced as basic building blocks of our project. Highlighted areas in yellow are base level goals for demo. Being able to test and demonstrate that each part works in SDII allows us to prove that we were able to build the project at its core level and function as intended.

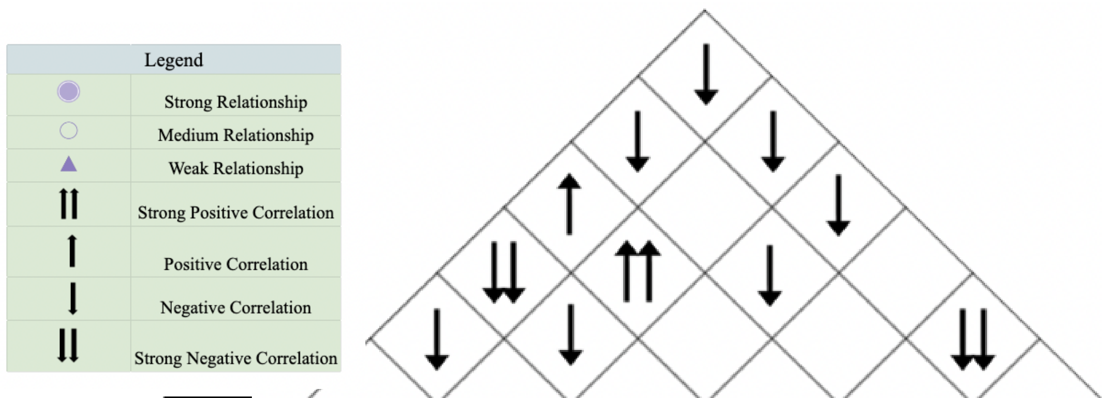
VCO (Voltage Controlled Oscillator)	Our required specification is a **dual** Voltage Controlled Oscillator **20Hz to 20kHz**. By using two, the versatility of our product doubles and We got richer and thicker synthesizer tones by slightly detuning the 2nd oscillator—formally the sub-oscillator. It can also be detuned an octave or two lower for sub bass. Produces saw and triangle waves.
VCA (Voltage Controlled Amplifier)	Used to adjust cut-off frequency through an input controlled voltage with distortion **less than 5%**
VCF (Voltage Controlled Filter)	Used to adjust cut-off frequency through an input controlled voltage for a frequency range of **20Hz to 21kHz** Able to control cutoff frequency and resonance
Signal Mixer	Used to combine two oscillator waveforms to **one single output**
Envelope Generator	The Envelope Generator will be used to control four different stages of a sound. **Four knobs for each stage** of: ADSR (Attack, Decay, Sustain, and Release)

RTDV	Real Time Digital Visualizer will be used to show the signal being applied and the output of such a modulated signal. Can be used only locally as deploying the feature would be a very far stretch goal. Should be able to run on **Windows, OS X, Linux**. Designed for website applications and not mobile. Capabilities of waveform visualizations/regular audio tempo based visualization will vary upon known API connections.
Software	**Two-tier architecture** from GUI to DAW or directly MIDI API connection will be implemented. The second tier, usually used for database/server will be where the MIDI notes will be stored and accessed (stretch could be three tier). Can (possibly) record samples & save them. Perhaps any audio quality option can be used such as lossless audio file format. Eg. FLAC, WAV, or AIFF. A low quality option can be incorporated as default.
MIDI Inputs	MIDI Inputs USB, Power Supply, VCF, VCO, VCA. MIDI defines **128 notes**, ranging from a C two octaves below the lowest C on an 88-note piano, to a G about 1-1/2 octaves above the High C on that same piano. The system will be able to send all 128 notes through to the DAW. MIDI-compliant
Power Supply	Will be used as a source of controlled power to activate different components on

Table 2.0 - Requirements Specifications

2.9. House of Quality (HOQ) Trade-Offs

This section will introduce a house of quality for our synthesizer as well as an in-depth comparison of each cross section. The HOQ is defined as a product planning matrix that is built to show how customer requirements relate directly to the ways and methods companies can use to achieve those requirements. The cross sections of the rectangular area show the relationship that each parameter of the product (in gray) has with qualities of the product (in white). The triangular top cross section provides the correlation, whether positive or negative, each diagonal intersection has with each other. For example, the very top section relates power consumption to the dimensions of the product.



	Importance	Power Consumption	Computational Power	Cost per Unit	Audio Processing	Weight	Dimensions
Sound Quality	5	○	●	○	●	▲	▲
Size	4	○	▲	▲	▲	○	●
Latency	4	○	●	○	●	▲	▲
Cost Per Unit	1	▲	○	●	●	▲	▲
Features	3	●	●	●	○	○	○
Portability	5	▲	○	○	○	○	●
Targets for Engineers		<30 W	>200 MIPS	<$200	4-stage processing	<12 lbs	2' x 5'

Table 2.1 - House of Quality

Power Consumption vs Computational Power - Negative

Computational power is based on the microcontroller that is selected for the project. This computational performance is measured in how many instructions the microcontroller can execute per second, generally in the millions or billions. If more instructions are required, the microcontroller works harder and will draw more power. This is known as the performance per watt measure of energy efficiency. The efficiency can be increased if the computations are simplified by writing more efficient code which will in turn reduce power draw. Since power drawn will increase as there are more instructions to execute on the hardware, this is a negative relationship.

Power Consumption vs Cost Per Unit - Strong Negative

Similar to power consumption vs computational power, a stronger power supply unit (PSU) will more often than not cost more than that of a lower wattage. As power consumption goes up, the cost per unit will increase as the PSU is upgraded. In addition, as power is consumed whether it be from a battery or an outlet, there is cost involved in producing this power. Considering the multiple factors that lead to a price increase, this is a strong negative relationship.

Power Consumption vs Audio Processing - Positive

[10] According to Rod Elliot, there is a correlation between energy efficiency and output power when looking at certain types of amplifiers (AB or D as shown in

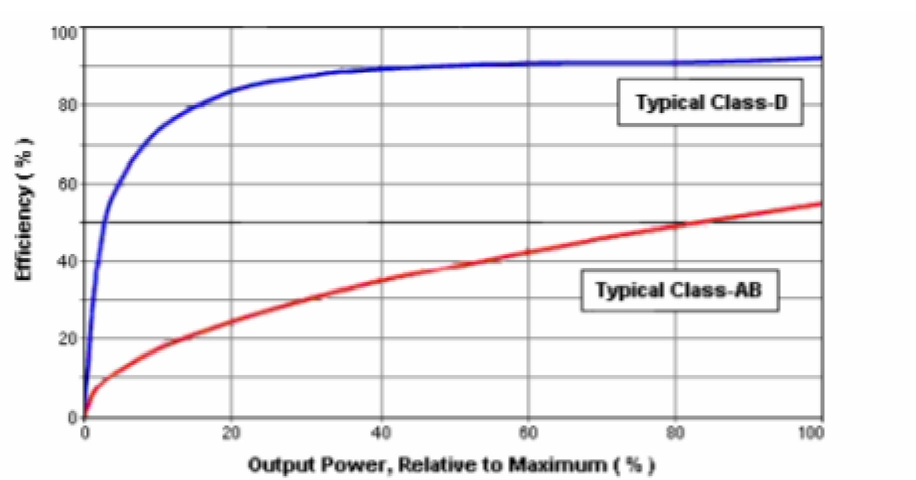


Figure 2.12: Efficiency vs. Power Consumption of Class D & Class AB Amplifier

the image above). There is a limit on how much energy amplifiers can output as denoted by the x-axis of the chart, however, the amplifiers become more efficient as their power increases as denoted by the positive parabolic curve.

This limit is more often than not below the absolute maximum possible power consumption to avoid unnecessary stress on the hardware. With more power and more efficiency, the sound quality also improves. The sound is not necessarily louder but the amplifier is able to provide more efficient, cleaner output which can enable louder audible output.

Power Consumption vs Weight - Negative

Devices that require more power require either a stronger battery or a bigger power supply. Considering that larger sized power supplies generally are heavier than weaker ones, the weight would increase as power demands increase. A heavier component therefore increases weight per unit and is a negative correlation.

Power Consumption vs Dimensions - Negative

With more power requirements for the device comes more powerful power supplies. There is a general positive relationship between power supplied by a power supply unit (PSU) and the size of the PSU–that is a higher wattage PSU will be larger than those that provide lesser wattages. This increase in size will utilize more space within the device and thus increase the overall size leading to a negative correlation between the two specs.

Computational Power vs Dimensions - Negative

As demands for computational power increase, so does the necessary space required to fit the computers that run the device. Although many computers' core processors come in the form of microprocessors, the scale of this project does not require such optimizing on the central processing unit (CPU) to require miniscule sized microprocessors. If multiple processors are needed, multiple spaces for these processors are required in turn. Albeit a slight increase in dimensions (or even negligible at times), this would be considered a negative relationship.

Computational Power vs Cost Per Unit - Negative

Referring to the comparison between power consumption and cost per unit, more powerful microcontrollers are required as the amount of computational power necessary increases. In turn, these more powerful microcontrollers generally cost more than their weaker counterparts. For example, a Raspberry Pi 2 microcomputer running at 900MHz costs roughly $25 compared to its more powerful successor, the Raspberry Pi 4 at 1.5GHz costing roughly $45. This would be considered a negative relationship. This negative relationship between computation power vs cost per unit is why we made this decision on the chart.

Computational Power vs Audio Processing - Strong Positive

Increasing the computational power allows the device to process audio more clearly as well as more accurately. Considering that the audio is processed as analog input and is transformed into digital waveforms. The ability to manipulate these waveforms with higher accuracy comes with the ability to process the sound which is granted by, and bottlenecked by, the computational power of the microcontroller. This correlation is considered strongly positive as the ability to process clear audio is one of the core functions of the device.

Cost Per Unit vs Dimensions - Negative

This comparison compiles multiple previous comparisons regarding power consumed, computational power, and audio processing quality. As the demands for these requirements increase as seen fit, their respective parts generally increase in size as well (as seen with the PSUs and CPUs) and require a larger housing space. These more powerful components also cost more than their weaker, generally smaller counterparts. Although not a direct correlation, there is a negative correlation when observing the dimensions of the device and its cost.

Cost Per Unit vs Weight - Negative

There are multiple factors that can increase the cost per unit physically. If more parts or components are required for the device, there will be a higher cost per unit. Not only would more money be required to build each unit, but each part that needs to be ordered would add to the cost. If these parts are heavier, the shipping more often than not would increase in price too leading to a negative correlation between weight and cost.

Weight vs Dimensions - Strong Negative

Looking at the size of the overall project and–generally speaking–bigger usually means heavier, an increase in the size of the enclosure of the device will lead to an increase in weight. If more components are added not only does the weight increase, but there needs to be generous enough space to fit said components. Therefore there is a strong negative correlation for each of these physical demands.

3. Research Related to Project Definition

In the following sections we discuss existing similar products on the market currently that were done for research prior to our build. This research aided heavily in overall design choices and knowledge. We will then discuss the relevant technologies that were researched and used within the scope of our project.

3.1. Existing Similar Products

Though digital synthesizers and virtual instruments inundated the market in the 80s, analog synthesizers made a comeback in the 21st century. Digital synthesizers solved many of the problems inherent to analog synthesizers such as size, temperature variation, tuning, portability, complexity, and cost. They solved these problems and expanded the limits of the electronically produced music world with new forms of synthesis, precision, polyphony, and complexity. These new selling points did not come without sacrifice though as digital synthesizers had a tendency to sound sadly sterile and cold. Not only this but the convenience of digital synthesizers comes at the cost of productivity in the production.

In the creative process, hardware synthesizers are more ergonomic and tactile in nature – allowing the musician to "connect" with their music as one during their workflow. Complaints such as these dominated the market until the late 90s when musicians began purchasing second-hand synthesizers that had gone obsolete. Analog synthesizer "warm" and "fat" sounds became highly sought at the turn of the century as the supply of vintage synthesizers dwindled and became increasingly expensive. Adding to the demand was the rise in eurorack modular projects and electronic dance/pop music. This called for synth manufacturing giants like Sequential Circuits, Korg, and Roland to get back in the game and many more to join.

Alongside these giants rose new competitors like Arturia, Behringer, Elektron, Novation Peak, and more. The market's renewed demand for organic and interactive sounds was made possible and more attainable than ever thanks to modern day surface-mount technology. Though still not cheap to the average consumer, analog synthesizers can be purchased for as low as $300.

To the common user, features such as usability, price, features, should be considered when comparing commercially available synthesizers today. On a more technical note, we compared their number of voices, keyboards, modular/normalized form factor, connectivity, and functions. The following analog synthesizers currently on the market will be compared: Arturia MatrixBrute, Moog Matriarch, Korg Minilogue, Behringer DeepMind 12, Roland System8, and Analog Four MKII.

Arturia MatrixBrute

Offered at a steep price of $2,500 is the Arturia MatrixBrute. Arturia is a popular synth manufacturer now that began as a software instrument company and then ventured into hardware in 2009 after emulating one of Dr. Robert Moog's infamous modular synthesizers. The MatrixBrute is the opposite of small and compact, it is known for its massive form factor and grandiose sound and rich functionality as a monophonic synth. Seen above are its four-octave capable 49 keys, making it ideal for composition on the spot and ideal for a keyboardist's workflow. The keybed is semi-weighted so that the musician feels a strong sense of control over their sound response. Its busy front panel is packed with three VCOs, three Low-Frequency Oscillators (LFOs), a noise oscillator, two filters, three envelope generators, and a massive matrix interface.

Figure 3.1: Arturia MatrixBrute

Of these functions, the matrix most likely sounds the most unfamiliar. In simple terms, it is the 16x16 purple grid array offering an interface for toggling the included 256 presets. It visually enables the 64-step sequencer and allows you to see and align which notes you desire to edit and toggle. VCO1 and VCO2 encapsulate all the essential waveforms possible for synthesis: sine, sawtooth, square, and triangle waves. The filters are all multi-mode. This means they are capable of low-pass, high-pass, band-pass, and notch – with cutoff, resonance, and drive options. The three envelope generators all feature Attack, Decay, Sustain, Release, and Delay. It has MIDI and USB I/Os and 12 CV/gate I/Os making it ideal for any workstation or MIDI controller. This beast of a synth is capable of creating any analog sound possible to the human ear. It is targeted towards experienced synthesists and ideal for highly sought-after warm analog sounds with edgy possibilities.

Moog Matriarch

Made by the most famous name in the synth world at $2,199 is the unique Moog Matriarch. Seen above is its iconic semi-modular form factor. Eleven hard-wired modules are included and cannot be removed or swapped out – however, patch cables are seen coming out. This is because the Matriarch offers the ability to reroute and create 90 patch points for a diverse range of sounds. It is the closest thing one can get to owning an original Moog modular synth at a fraction of the cost. Moog was able to stay true to their traditional analog signal path while making a modern MIDI compatible synth with a digital sequencer.It features a 49-key velocity-sensitive keyboard and CV out The signal processing consists of four oscillators, each offering triangle, sawtooth, square, and pulse waveforms.

Figure 3.2: Moog Matriarch

Though monophonic in nature, the Matriarch has paraphonic capability such that you can play chords up to four notes for each oscillator being passed through a single amp/filter by bypassing the mixer. If the six-channel mixer is used, we reach the two filters, configurable for lowpass and bandpass with two resonance and one cutoff knob. The envelopes are hardwired to the amp and filter but can be patched to other locations on the panel. Dual VCAs have three modes each such that the envelope is either applied to either VCAs, filters, or both can derive gain from their CV input. Everything on the Matriarch is usable without any patch cables at all but has impressive patchability with five CV inputs and two outputs. The synth itself is lacking in LCD displays for menu settings which some may say should be expected for the price point. Overall, it is a specially engineered vintage yet modern design – making it ideal for experienced synthesists that know how to yield the most out of its near endless patch possibilities for experimentation.

Korg Minilogue

One of the most affordable fully analog synthesizers on the market is the Korg Minilogue at $540. This is a small, portable synthesizer not to be judged by its size as it is feature packed and known for its high-quality Korg sound offering the best of both worlds, warm yet progessive.

The Minilogue's compact form factor features a 37-key velocity-sensitive keyboard. Though lacking in aftertouch, it is responsive and ideal for traveling musicians. It was their first keyboard equipped polyphonic synth since the 80s and it is still deemed one of the best analog synthesizers for beginners. Compared to the last two products, it might be thought of as basic but for the price point it has no competition. Upon startup, the LCD display greets the user with a message indicating the synth is self tuning and then goes on to serve as a preset setting display and real-time oscilloscope. This allows the user to get a visual indication of any changes they are making to their sound and allows you to "see" the sounds.

Figure 3.3: Korg Minilogue

The MiniLogue is a four-voice polyphonic synth with two oscillators, each with sawtooth, triangle, and square options. It allows for a maximum four voice polyphony such that four notes can be played simultaneously. It contains a low pass filter with the unique ability beyond cutoff and resonance – velocity tracking. This allows you to manipulate how the filter acts depending on how hard you strike the keybed. The VCA is a standard audio amplifying VCA. Couple all these features to the 16-step sequencer and arpeggiator and you have massive control over your analog synth all in a small, yet affordable package. Being MIDI-compatible makes it a great synth to fit on every musician's desk and its basic analog topology makes it an excellent choice for beginners.

Roland System-8

After four decades of Roland's success in the music industry, they decided to launch one of the most iconic modernly available versatile synthesizers, known as the System-8. The sound engine is purely analog – warm and fat yet still allowing for experimental shark sounds too. It was meant to replicate certain sounds of Roland's classics like the Jupiter-8, Juno, and more. Designers did so by replicating identical oscillator, filter, and effects architectures from the

Roland’s circuit designs. It even comes with a collection of a handful of Roland’s infamous synthesizers that can be used in a DAW setup through MIDI over USB. Alongside its handy MIDI/USB connections is built-in CV/GATE outputs so you can use it with your software VSTs or any other hardware gear on hand.

Figure 3.4: Roland System-8

It consists of two primary oscillators for sawtooth, square, and triangle waves and a third oscillator that serves as a sine or triangle wave generator. With a 49-key velocity sensitive keybed, the System-8 is polyphonic with up to 8 voices. The possibilities for vintage Roland sounds that any synthesist could recognize from a mile away make this synth worth the price of $1,750.

Elektron Analog Four MKII

Elektron – another emerging name in the synthesizer industry is the Analog Four MKII. Unlike giants such as Moog or Roland, this company did not come to fruition until 1998 and is most famous for their drum machines.

Figure 3.5: Elektron Analog Four MKII

Seen on the left is the unique Analog Four MKII, a four-part analog synth that can easily fit on your computer desk. Marketed as a four-voice, polyphonic synthesizer with a built-in sequencer, CV I/O, and built-in digital effects – the MKII at its core is an analog synth from the oscillator to output. Its most notable features looks wise is the clear OLED display for editing and generous amount of ports on the backside. It shines connectivity wise, with USB, MIDI, CV/Gate out, and audio I/O for each of its four voices.

Also noticeable is its lack of a traditional keyboard. Elektron decided to use a simple one octave keyboard with no piano keys in sight and no velocity sensitivity. This was a bold design choice in terms of playability but definitely makes for a cohesive and attractive look.

Moving internally, the sound engine of the Analog Four MKII consists of two dual-oscillators (sawtooth, square, and triangle) for the ability of creating four voices. With such, you can either play one note per voice or four notes per voice. Included are dedicated LFOs and sub-oscillators that can be assigned to each oscillator for added complexity and layered sounds. The filter section consists of two filters, one lowpass and one multi-mode. There are two envelope generators – assignable to either the filter or amplitude modulation. With the audio input ports, musicians can work the filters and effects with external sounds of their choosing. The CV input ports allow you to control other hardware gear like modular synths too. Another modern feature of the MKII is its use of digital encoders over potentiometers. This means all knobs on the synthesizer spin continuously, making them fairly uncommon in this domain due to their software complexity but have great feedback on the MKII. Though this synthesizer comes in at a steep cost of $1,649, its connectivity, built-in digital effects, and sequencer make it stand out in anyone's setup.

Summary

These are just several of the many different analog synthesizers on the market right now. In our comparison between manufacturers, models, and their corresponding specifications we were able to draw inspiration and insight on what we wanted to incorporate in our design. We covered a wide variety of synthesizers – some more suitable for beginners looking to learn (like the Korg MiniLogue) and some meant for professional musicians (like the Moog Matriarch). Each synthesizer shines in their own way, whether it be their sound design capabilities, connectivity, ease of use, added digital features, effects, or modulation options. Though all options compared were rich in functionality beyond our constraints, we were able to draw inspiration and features we sought to replicate under the cost of $400.

3.2. Relevant Technologies

This was one of the most important aspects to why and how the analog and digital synthesizer has developed such a high growth and appeal to artists, hobbyists, musicians and so on. The use of many technologies to complement and work with a synthesizer such as a MIDI controller, DAW system, keyboard plugins, etc. These technologies are those that are well aligned with the goal we are trying to achieve with our creation.

Analog Signal Filtration:

Due to the goal of making a hybrid synthesizer, signal filtration is a very high priority for our team to incorporate. In Moog's and other famous contributors to the world of synthesizers and modulation, the primary method of modifying raw sounds involved using Voltage-Controlled FIlers (VCFs). This also provides a way of utilizing and expanding the range of sounds the user is capable of creating within the product. Filters are devices that provide the potential to limit the loudness of certain parts of the audio spectrum. This offers the user the ability to reduce the volume of certain harmonics present in a sound. VCFs receive signal input from signal sources, more specifically in this project our oscillators and by varying the cutoff frequency, the filter passes or attenuates partials of the input signal. Some filters are even designed to provide enough feedback to go into self-oscillation, and it can serve as a sine-wave source! The musical characteristic of a particular VCF depends on both its linear frequency response and its nonlinear response to large amplitude inputs.

Envelope Generator:

Another important piece of technology we will implement in the synthesizer is envelope generators. They are a very old concept regarding synthesizers and are a common means of modulation. An envelope generator produces a rising and falling control signal (the envelope itself), it defines a trajectory along which any parameter changes over time can all be dictated by envelopes.

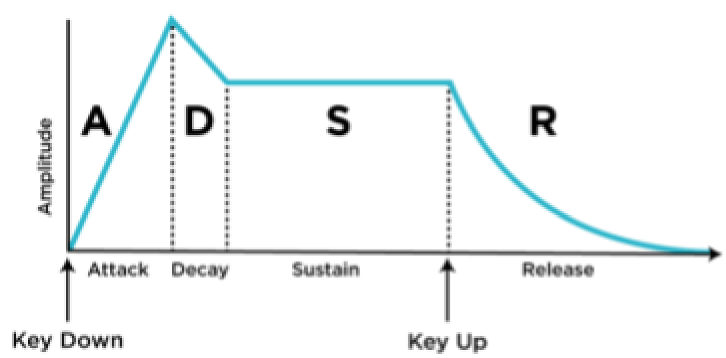


Figure 3.6: Envelope Generators

In traditional synthesizers, envelopes are usually one-shot modulation sources triggered by pressing a key or advancing a sequencer. The envelopes usually affect the loudness of the sound or the timbre. One can think of the envelope as a container: it is a particular shape that evolves over time, and the sound's loudness and timbre conforms to the container's shape, as if it were a liquid or malleable substance. The most common type is ADSR, a four-stage envelope generator. It stands for attack decay sustain release, referring to the four states through which the envelope advances from the press of the keyboard. Shown in the figure above and also discussed more in depth in section 2.2 regarding chosen methodology.

DAW System:

A digital audio workstation (DAW) is software used for recording, editing and producing audio files. There are four main functions of a DAW (although there are many more now with modern ones):

1. It serves as a digital audio processor, meaning it can record, edit and mix audio digitally.
2. It servers as a MIDI sequencer, meaning it can record, edit and mix MIDI notes
3. It has virtual instruments. It receives MIDI information and translates it to different instrument sounds.
4. Music notation, it can turn MIDI notes into printable sheet music.

Overall, it is the translator of all MIDI information. This is a technology we will use towards the final steps of our project using Ableton Live.

MIDI Controller/Keyboard:

A MIDI controller keyboard is a device used to send MIDI data to a computer or other piece of hardware. It's essentially a keyboard that is used to trigger sounds coming from an external device. The external sounds are coming from another piece of hardware such as a virtual instrument from a DAW system. This type of controller gives the user real control over a number of various parameters and come in many shapes and forms.This device for our group will be bought on it's own and used to show the syntehsizer's capabilities in the digital sense as well as how well we implemented MIDI to CV capabilities and interacting with software. In this sense we are not dealing with audio signals at this point, but the language of MIDI itself that only your DAW and virtual instruments understand. Our team deemed this a relevant technology in our project as it gives it a more physical feel and understanding of the types of

Figure 3.7: Example of a MIDI Complaint Keyboard

sounds they're making, rather than just clicking them with a mouse on the DAW system.

The MIDI keyboard we are working with specifically is the AKAI Professional MPK MINI, it gives the user maximum portability and supports easy plug-and-play as well as USB-MIDI which is relevant to our synthesizer design and standard selections.

3.3. Strategic Components and Part Selections

In this section, we compare potential options for the several core components of our design: microcontroller, ports, VCO, VCF, VCA, envelope generator, power supply, and human interface devices. We will be differentiating the pros and cons between part options and highlighting the selected parts in the color yellow.

3.3.1. Microcontroller

A microcontroller (MCU for microcontroller unit) is a small computer on a single VLSI integrated circuit (IC) chip that is designed to govern a specific operation in an embedded system. A typical microcontroller includes a central processing unit (CPU), flash memory or random access memory, and input/output (I/O) peripherals on a single chip.

Sometimes referred to as an embedded controller or microcontroller unit (MCU), microcontrollers are found in devices we use everyday, and in this project with our synthesizer including digital aspects, we need one! They are essentially simple miniature personal computers (PCs) designed to control small features of a larger component, without a complex front-end operating system (OS). By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes.

This will serve as essentially the brain of our synthesizer and will be responsible for allowing all the separate parts to communicate with each other, as well as most importantly be processing and aiding in the conversion of MIDI notes to control voltage signals—an essential for our synthesizer to work.

Some considerations our team went through in choosing the right MCU:

What is the scope of the instrument? Meaning, is this meant to be monophonic for a simple modular system or buildable for a huge polyphonic system in hopes of creating orchestral-like sounds?

What waveform models do we want to use? The synthesis model we choose will heavily inform the choice of the processor we should use. Vintage FM will have much different demands and constraints than something like wavetable, for example.

In our stretch goals, do we want to integrate a sampler into the system at any point? What type of **fidelity requirements** are we aiming for?

What additional functional support for I/O do we want? All formats and outputs (such as, MIDI conversion) should be taken into consideration. Lastly, is the community supportive enough or has enough resources such that if we run into an issue, it won't be far too niche to solve and move forward.

Following are the microcontrollers we have researched and considered in the project, with comparisons and overall specifics to why we chose the MCU for this project.

ESP32-S2-MINI-2

ESP32 is a low-cost, low-power system-on-a-chip (SoC) series with Wi-Fi & dual-mode Bluetooth capabilities. There's a dual-core or single-core Tensilica Xtensa LX6 microprocessor with a clock rate of up to 240 MHz. ESP32 is highly integrated with built-in antenna switches, radio frequency balun, power amplifier, low-noise receive amplifier, filters, and power management modules. Engineered for mobile devices, wearable electronics, and Internet of things (IoT) applications, the

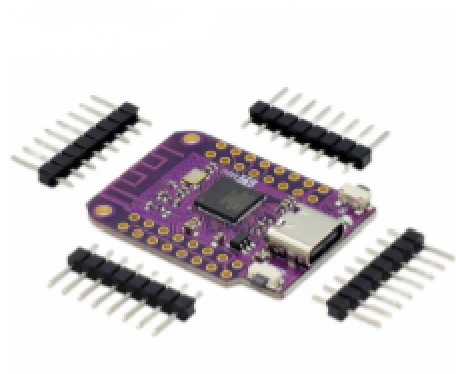


Figure 3.8: ESP32-S2-MINI-2

ESP32 achieves ultra-low power consumption through power saving features including fine resolution clock gating, multiple power modes, and dynamic power scaling.

Processors:

Main processor: Tensilica Xtensa 32-bit LX7 microprocessor

- Cores: single-core
- Clock frequency: up to 240 MHz

- Performance: up to 600 DMIPS
- Ultra low power co-processor: allows you to do ADC conversions, computation, and level thresholds while in deep sleep.

Peripheral input/output:

Rich peripheral interface with DMA that includes capacitive touch, ADCs (analog-to-digital converter), DACs (digital-to-analog converter), I^2C (Inter-Integrated Circuit), UART (universal asynchronous receiver/transmitter), CAN 2.0 (Controller Area Network), SPI (Serial Peripheral Interface), I^2S (Integrated Inter-IC Sound), RMII (Reduced Media-Independent Interface), PWM (pulse width modulation), and more.

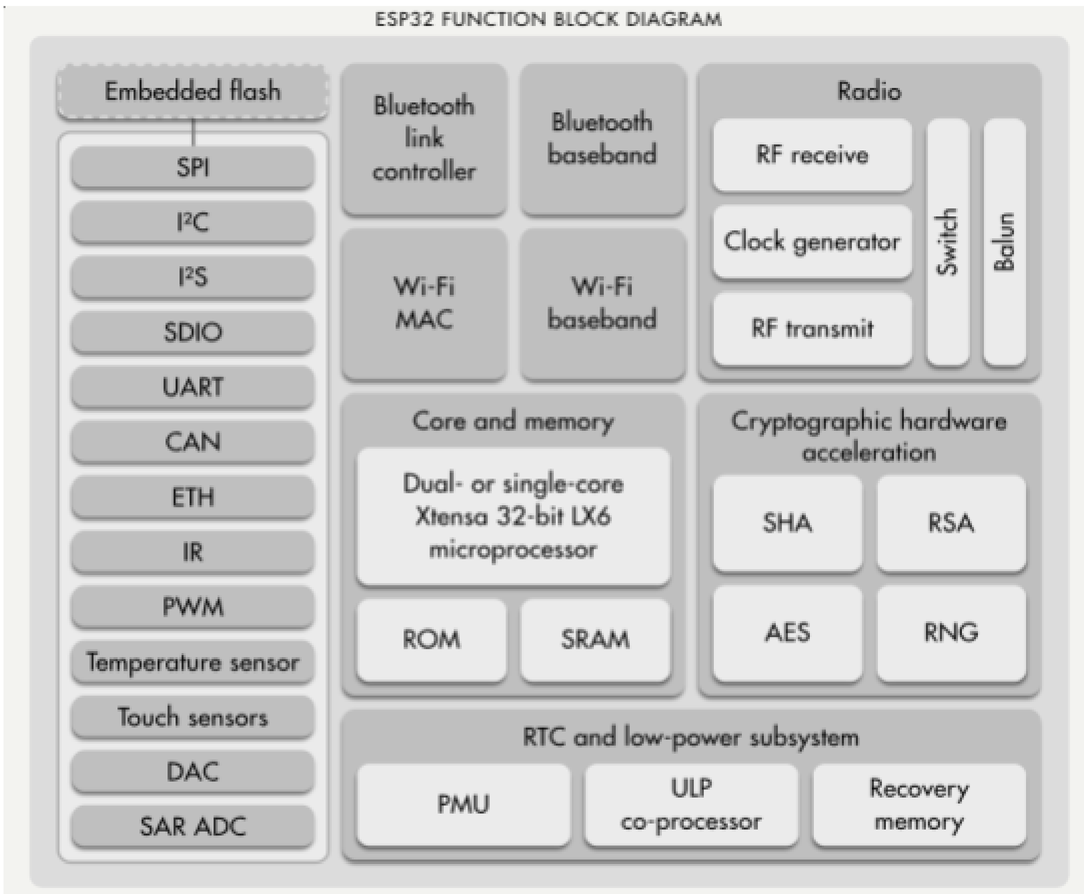


Figure 3.9: ESP 32 Block Diagram

Downfalls: From the synthesizer community we have learned that while the price is amazing, we’re not really looking for wifi/bluetooth capabilities regarding our project right now. We have seen through research that analog inputs have proven to be only okay, meaning you’re not going to get the great fantastic crisp quality that we may be looking for. For extra quality it has been recommended that we use an I2C ACD module. Analog output is really bad and this has been the case with most of our research when used in any synthesizer projects.

STM32F4

The STM32 family of microcontrollers offer a large number of serial and parallel communication peripherals which can be interfaced with all kinds of electronic components including sensors, displays, cameras, motors, etc. All STM32 variants come with internal Flash memory and RAM. The STM32F4, specifically, is widely used in the DIY synthesizer community for its niche audio capabilities and built in libraries or already product debugging tools.

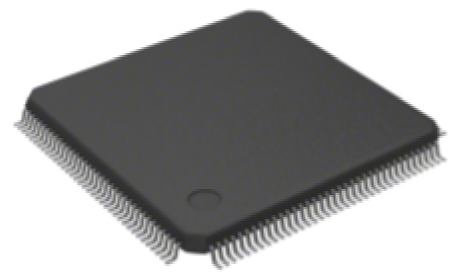


Figure 3.10: STM32F4

The STM32F4DISCOVERY Discovery kit is the MCU our team was looking at, and it leverages the capabilities of the STM32F407 high-performance microcontrollers, to allow users to develop audio applications easily. It includes an ST-LINK/V2-A embedded debug tool, one ST-MEMS digital accelerometer, one digital microphone, one audio DAC with integrated class D speaker driver, LEDs, push-buttons, and a USB OTG Micro-AB connector.

Axoloti Core/STM32F427

Axoloti Core is a circuit board with stereo audio in- and output, audio analog-to-digital and digital-to-analog converters and a microcontroller suitable for digital audio processing. All connectors are on one side to make it easy to build your own tabletop device, rackmount, stompbox or something else. The Axoloti Core comes with a built in STM32F427 microcontroller and has been specially crafted for hobbyists and creators of audio products specifically aimed at the audio synthesizer community. This comes with a lot of advantages, but also a lot of limitations which was the main reason we did not choose this MCU.

The specs and connections include an assembled board, 168MHz STM32F427 microcontrollers, 24 bit audio ADC/DAC, MIDI input/output, USB device port, compliant USB-MIDI host port, and more. For what it costs it is very powerful, and can support 3-4 voice polyphonic devices. The main disadvantage we saw with this device is that it was too powerful for our monophonic project, and took away building some of our key hardware ourselves since it was all included. It also had a great community but most of the products were discontinued and the documentation on the platform and code was limited, and this could cause a chain reaction of issues in the future. The device will be stored for future use as it has endless possibilities in musical instrument creation with such fantastic user feedback.

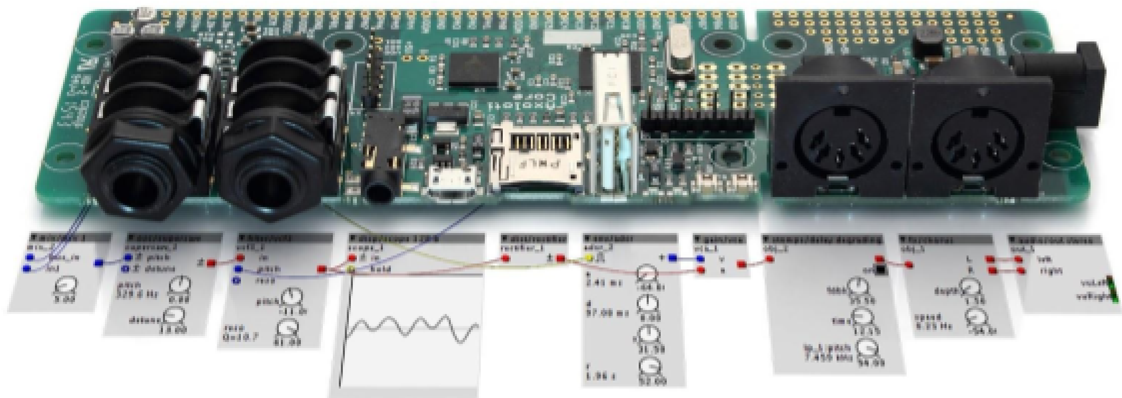


Figure 3.11: Axoloti Core/STM32F427

Raspberry Pi 4 Model B/BCM2711

Figure 3.12: Raspberry Pi 4 Model B/BCM2711

The Raspberry Pi 4 Model B is the latest version of the famous mini- computer family of Raspberry Pi. Our team had originally opted for the Raspberry Pi 4 due to its small size, relatively cheap cost, and extensive utility. The Raspberry Pi 4 has multiple random access memory (RAM) options, at this point in time we do not have a memory requirement thus it is uncertain how much RAM we will need, however, the Raspberry Pi 4 has options ranging from 1GB to 8GB of memory. The processor is a quad-core ARM chip on a 64-bit system with a clocking speed of 1.5GHz, this means the processor is able to execute up to 1.5 billion instructions per second.

The main reasons we decided against using this model and all models of a raspberry pi is that it was too advanced for what we needed. The OS and coding environment wasn't C based and might have run into much more bugs trying to set it up and connect against the Arduino Uno/ATtiny85. It was also much more expensive than the rest of the products we compared and not worth the price. If we were to make it a completely digital synthesizer, we would have gone down this route and maximized the product for all it's uses, but really it's powerful processor and all it's features would have gone unused in the time frame we had for this project, so this was our reasoning for not using it.

Teensy 4.0/ARM Cortex-M7

Teensy is a line of microcontroller boards developed by PJRC, an independently co-owned component creation company. The 4.0 is the smaller brother of the latest 4.1 edition of their Teensy boards, however despite its size, still boasts a reputable ARM Cortex-M7 CPU capable of 64-bit floating point precision calculations at 600 MHz as well as 24 solderless general purpose input-output pins (GPIOs). The board is also capable of **pulse width moderation (PMW)** timing functions through the use of a 24 MHz crystal and a 32 KHz crystal. All this can be done with as little as 3.3V powering the board.

The Teensy microcontroller line runs on the Arduino integrated development environment (IDE) through the use of PJRC's add-on, Teensyduino. Once this add-on has been installed, we can program the Teensy board on the IDE using a vast library created specifically for this board. There are other programming options available as well if the Arduino IDE is not the path one wishes to take.

Arduino Uno Rev3/ATmega328P

The Arduino Uno Rev3 is a microcontroller board created by Arudino. The board contains six analog inputs, 14 digital input/outputs, a ceramic resonator, and other various microcontroller features. The "Uno" marks the big change in Arduino's software with the release of Arduino's 1.0 Integrated Development Environment (IDE) first released in 2010. These boards are relatively cheap due to their age, but with age also comes extensive documentation. The Arduino Uno is relatively weaker compared to other microcontroller boards such as the Raspberry Pi, however, it is

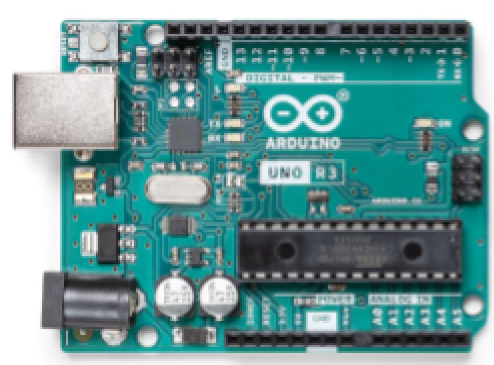


Figure 3.13: Arduino Uno Rev3/ATmega328P

successful at accomplishing low-requirement tasks efficiently. This is the main reason we have opted to utilize the Arduino Uno for this project.

The brains of the Arduino Uno come from the installed ATmega328 microcontroller which boasts 23 GPIOs and a 20 MHz clocking speed whilst only requiring 1.8 V on the board. As both the ATmega328 and the following microcontroller, the ATtiny85, have been developed by Microchip Technology (previously Atmel Corporation), previous projects have utilized the Arduino board in parallel with an ATtiny85 microchip processor for MIDI-to-CV conversions.

ATtiny85

One of the microcontrollers our team opted to use is the Atmel ATtiny85. This microcontroller is the second of two microcontrollers that will be utilized for the MIDI-to-CV conversion part of the project; the Arduino Uno being the aforementioned first microcontroller board utilizing the ATmega328 which also has a USB input and various other components that you can plug into a laptop and use it seamlessly.

Figure 3.14: ATtiny85

This is just a small 8-pin microcontroller, and it also is an 8-pin dual in-line package (DIL) which means you can't program it without other hardware due to its parallel rows of connecting pins. This may seem like a disadvantage, however it can be easily manipulated for our own purposes at a fraction of the cost of alternative microcontrollers. Sometimes, especially in engineering, more expensive or newer doesn't always mean better. In terms of programmability, this ATtiny85 is able to interface with other microcontroller boards to be programmed by such as the Arduino Uno or programmed by an in-circuit programmer called the USBasp developed by the creators of the microcontroller, Atmel.

The reason we don't mention a Raspberry Pi here as an option is because it has a completely different operating software, programming language and learning curve. This learning curve isn't too vast but at UCF we have been learning and pushed to understand, live and breathe the C language. The entire curriculum at our University is around the C language. In embedded systems, the program looked exactly as the code we see on the Arduino IDE. Even as first year engineers, our first and second semester is programming an Arduino MBOT to race in a maze. It would be a great way to both start and end our engineering educational careers and see our progress as programmers.

Over anything else though, this is the cheapest option and uses much less power. Setting up an environment is much easier with the IDE and debugging is much easier having such a large community. The Arduino Uno model we are getting has been out for 10 years, and when mentioned that “newer isn’t necessarily better” this comes into play as many issues we run into more than likely have been talked about, posted and solved. This is another primary reason why we chose to go this route.

This device has two types of memory on it. It has an electrically erasable programmable read-only memory (EEPROM) of 512 bytes that is non-volatile along with 2096 bytes of flash memory. The ATtiny85 has 5 inputs or outputs furthermore it can stretch to 6 if you program the reset pin to function as an extra in/out. The ATtiny85 has a bidirectional port known as port B. This means it can work as inputs or outputs. Along with this it also has a 10-bit analogue to digital converter with 4 multiplexed inputs.

3.3.2. Parts Used

There are many factors to consider when selecting a microcontroller unit from how powerful the main chip is on the board to how much power is drawn from the board. These comparisons are outlined in the tables below.

Out of the six microcontrollers researched, our team has decided on selecting the combination of the Atmel ATmega328P and Atmel ATtiny85 units (which are highlighted in yellow below). Previous synthesizer projects have utilized the two processors in parallel with each other as the ATtiny85 is known to be very programmable, lightweight, and efficient. The ATmega328P is included on the Arduino Uno and is the main brains of the microcontroller. It also allows full programmability over the ATtiny85 as we can connect the two microcontrollers together.

One of the main reasons for selecting a single-board microcontroller such as the Arduino is that we are able to utilize the general purpose input/output pins (GPIOs) to provide power and functionality. The ease of connection with a single-board microcontroller with an external computer allows for easy programming of the boards to fit our needs. In the following diagram we will compare MCUs and their relevant technical specifications for our synthesizer and more specifically the MIDI-to-CV conversion.

MCU	ESP32-S2-MINI-2	STM32F4	BCM2711 [11]
Manufacturer	Espressif	STMicroelectronics	Broadcom Inc.
Relevant Board	N/A	Axoloti Core	Raspberry Pi 4
CPU	Xtensa LX7	ARM Cortex-M4F	ARM Cortex-A72
OS	32-bit single core	32-bit single core	64-bit quad core
Clocking Speed	240 MHz	Up to 180 MHz	1.5 GHz
Flash Memory	4 MB	0.5 - 2 KB	1 MB
RAM	320 KB	340 KB	Up to 8 GB
Power	3.3 V	1.8 - 3.6 V	5.1 V
# of GPIOs	37	51	28
Communication	I^2C, UART, SPI	I^2C, SPI, UART	I^2C, SPI, UART

MCU	TEENSY40 [12]	ATtiny85 [13]	ATmega328P [14]
Manufacturer	PJRC	Atmel	Atmel
Relevant Board	Teensy 4.0	N/A	Arduino Uno
CPU	ARM Cortex-M7	AVR	AVR
OS	32-bit RISC	8-bit single core	8-bit single core
Clocking Speed	600MHz	20 MHz	20 MHz
Flash Memory	2 KB	2 KB	32 KB
RAM	1 MB	512 B	2 KB
Power	3.3 V	2.7 - 5.5 V	2.7 - 5.5 V
# of GPIOs	24	6	23
Communication	I^2C, SPI	USI	I^2C, SPI, UART

Table 3.0: Microcontroller Comparison

3.3.3. MIDI Port vs 5-pin DIN Connector

In this part selection we examine MIDI transmissions and what kind of connections we want to make between our devices. There are four main ways to transmit MIDI data and the two that we will compare are DIN-MIDI and USB-MIDI. It is important to reemphasize that MIDI consists of a set of digital signals that are used for controlling and playing electronic instruments, and to realize that MIDI does not contain any actual sound itself. It simply acts as the middleman we need to transmit these sounds digitally and back. It's instructions for how other instruments should create sound, in a sense like the conductor of an orchestra. He doesn't make the sound himself, but tells them what to play, when to play, and what expression to use while playing. Once we understand this concept, we understand that choosing how we want to transmit this data is very important for our part selection in our overall product quality and design.

In the following table, we compare the pros and cons between using a DIN-MIDI port vs USB-MIDI port. These are two popular routes in synthesizer projects thus we wanted to compare socket type, bidirectional capabilities, pin spacing, MIDI compatibility, and most importantly – reliability and cost. Here is our overall comparison table:

Device	***DIN-MIDI port***	***USB-MIDI port***
Price	$1.75	$13.00
Reliability	Mature technology	More susceptible to noise, more compatible for future solutions
Socket Type	Female 5-pin DIN	Female USB Type A
Bidirectional connection?	No	Yes
Major Drawback	Limited MIDI channels	Single connection, works only on Host / Peripheral Basis
Pin Spacing	Yes	Yes
Voltage	1V	
MIDI compatible?	Yes	Yes

Table 3.1: Comparison of DIN vs USB for MIDI Reading

USB-MIDI connection:
The reason this type of connection exists is because USB has become the most widely used and accepted standard for computers. Your very own computer doesn't have a "MIDI" port, right? [15] Many modern MIDI instruments or controllers include both a DIN connection and a USB connection port now. This makes it very easy to plug your MIDI device seamlessly into your computer or hardware needed, and we considered this a high positive in making our design simple and easy to demonstrate/implement in our design. On top of that, USB-MIDI connection is bi-directional, which is not the case with standard DIN ports.

Downfalls: USB-MIDI works on a host/peripheral basis. This means that the computer MUST be the host and any MIDI device or controller is considered a peripheral. If you wanted to connect a MIDI instrument and a keyboard directly via USB, you would not be able to. You have to connect each of them to a computer system first. Another drawback is that USB-MIDI cables are much shorter than DIN-MIDI cables, which reach up to 50 feet. This could be important for our final presentation! DAWs, workstations, and sequencers use MIDI to compose music. They use a grid or series of steps to play notes in a pre-defined sequence. Composition and sequencing are core to why MIDI was created and is still used to this day. Other than storing and transferring .MIDI files, USB has no comprehension of what a musical sequence is because there isn't a significant need for it. This makes it extremely difficult for developers and engineers to work with this type of connection.

USB was created as a means to make file storage and transfer more universal. USB wasn't created with music and musician's needs in mind, as it was made for a much larger audience and for more applications. While it can make our lives easier, as engineers with design and quality in mind, this is why we chose not to go with this standard and design choice. DIN-MIDI cables are made for the specific type of device we are making and it keeps music in mind throughout the entire way.

DIN-MIDI connection:
5 pin connector cables, officially named DIN 41524 connectors) were originally used as hi-fi cables back in the 1970s. They were not used for a long while, but when the concept of MIDI was introduced in 1983 Japanese manufacturers decided to make this the original MIDI cable standard. It is used for much more than hi-fi now, and you will see a "MIDI OUT/MIDI IN" port on almost every piece of MIDI gear even today.

The main advantage is that MIDI is a universally used and recognized format, making the same MIDI cable functional for uses across all versions and forms of musical equipment. Whether it's from a guitar amplifier to a channel switcher, or an electric keyboard to an audio interface, the same cable serves both equally, including our hybrid synthesizer. MIDI simplifies commands and is an easy organizational tool for professionals of all types.

Standard MIDI cables are wired with pins 4 and 5 carrying the current, and pin 2 being used for the shielding to ground the cable. This means pins 1 and 3 aren't needed and thus not wired. With a 5-pin discrete cable, which is what almost all "MIDI cables" follow now, all the pins are wired, allowing for any application that requires pins 1 and 3 to transmit something like power.

Other MIDI cables without all pins wired could not perform the additional functions that require the connection of all five pins.
the build quality, reliability, warranty, and secure connection. With classic tried and true cables like these ones, they have stood the test of time and we can't necessarily say the same about USB-MIDI transmission.

Downfalls: Although this has long been the standard, DIN-MIDI connections are flawed in that they are not bi-directional. MIDI signals always go out from a MIDI Out connection, and then to a MIDI In connection. That means one must always connect DIN-MIDI cables from a controller's MIDI Out jack to an instrument's MIDI In jack. However, this downfall is only relevant to the use you have of the ports, but it's a fantastic and organized system if we only want to connect our computer/speaker to one keyboard or controller. This is what our project entails and we don't need multiple connections or MIDI instruments. We mainly focus on pure analog manipulation and transferring between the two.

There is also a way to handle multiple sounds (note, these are not voices such as those for a poly synthesizer), however, even with the 5-pin DIN. This is by MIDI channels. Each hardware DIN-MIDI connection has 16 MIDI Channels available, each of which can send instructions to a different MIDI instrument. This means, in theory, we can talk to up to 16 instruments all at once. This is how we defend the downfall a 5-PIN port has, and if we ever needed more than that, we would revisit this conversion and lean more towards a USB port, or simply adding more DIN-MIDI hardware ports on our board, each of which bring us each 16 MIDI channels that we can use.

This diagram shows how the MIDI cable is connected: (Note that two of them remain unconnected, but the standard cables made today are "discrete" meaning you can use the two unconnected pins if an application requires it).

First connector → Second Connector	Cable
PIN 1 → PIN 1	No connection
PIN 2 → PIN 2	Shield
PIN 3 → PIN 3	No connection
PIN 4 → PIN 4	Voltage Reference Line
PIN 5 → PIN 5	Data Line

Table 3.2: MIDI Pin Definitions

When we implement our MCU to manipulate MIDI data, the logical signaling that this transmission type uses is fairly simple to understand as we used the same type in our embedded systems class. This is a schematic found on the data sheet for when the UART signals are at logic levels. When it is idle, it sits at a logical high state. Each byte is prefaced with a start bit, followed by 8 data bits, then one stop bit. MIDI does not use parity bits.

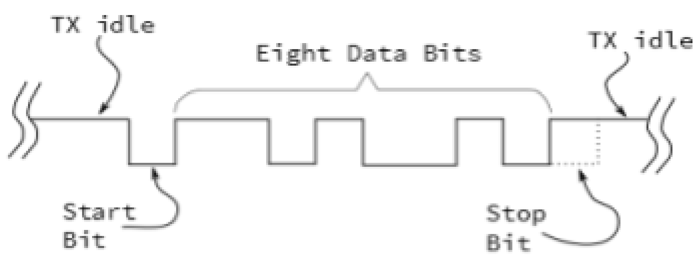


Figure 3.15: Parity bit illustration

MIDI uses a clock rate of 31,250 bits per second. To send an 8-bit byte, it needs to be bookended with start and stop bits, making ten bits total. That means a byte takes about 320 microseconds to send, and the maximum throughput on a MIDI connection is 3,125 bytes per second. The average MIDI message is three bytes long, taking roughly one millisecond to transmit.

To end our part selection discussion, our overall decision to go with this transmission port is due to it being the original standard, it’s easier to implement, cheaper, and since we only need a unidirectional connection it made the most sense to choose it.

Extra parts needed for MIDI-CV Converter:

This diagram illustrates the additional parts we will need for our converter

Component	Type	Quantity	Price
Resistor	220R	2	~$0.04
Resistor	4K7	2	~$0.05
Capacitor	100nF	2	~$1.04
Opto Coupler	6N137	1	$1.06
DIN Socket	5PIN	2	$4.24
Speaker	N/A	1	N/A

Table 3.3: MIDI-CV Converter Parts

3.3.4. VCO

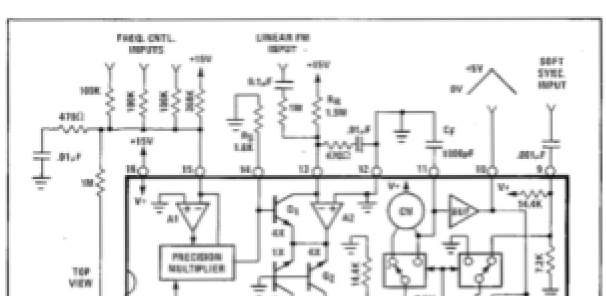
The following table goes into comparing two viable options for our VCO integrated circuit. The synthesizer IC market is largely dominated by the AS series but we discovered the SSI line after extensive research. We discussed their temperature compensation abilities, octave tracking abilities, waveform output possibilities, and of course – cost and packaging.

Device	*AS3340*	*SSI2130*
Price	$5.43	$8.95
Packaging	16-pin DIP	SMD QFN
Waveform Outputs	Saw, square, tri, PWM	Saw, square, PWM
1v/oct Tracking?	Yes	Yes
Trackable Octaves	10	10
Temperature Compensated?	Yes	Yes
Temperature Drift	+/- 50 ppm/degC	+/-200 ppm/degC
Oscillator Core	Triangle	Triangle
Frequency Sweep Range	500,000:1	1000:1

Table 3.4: VCO Device Comparison

AS3340

Based off of the iconic CEM3340, the AS3340 is an identical reissue.



This Integrated Circuit (IC) replaces all the complex circuitry involved in VCOs making it ideal for our large PCB. It is the same oscillator used in celebrated synthesizers such as the Moog MemoryMoog, Oberheim OB-X, and Roland Jupiter-6. Seen below is its functional block diagram.

It is capable of all four output waveforms: sawtooth, square, triangle, and pulse. It is operable off of +/-15V supplies as well as +15, +/-12 and -5V supplies. It is fully temperature compensated – a crucial part of keeping the VCO in tune to 1 volt/octave. Additionally, the AS3340 at its core is a triangle core VCO, meaning there is no reset time in the circuit making it excel at linear tracking. Thanks to the chip's trimmer adjustment procedure and high exponential scaling accuracy. It comes in through-hole packaging and comes in at around a cost of $5 making it ideal for our design.

SSI2130

Another viable option is the new SSI2130 by the Sound Semiconductor manufacturer. This chip is relatively new to the synthesizer DIY community. It is a more robust modern chip released within the last two years and comes in 32-pin Quad Flat Lead (QFN) packaging. The SSI2130 is capable of triangle, sawtooth, and pulse width modulation.

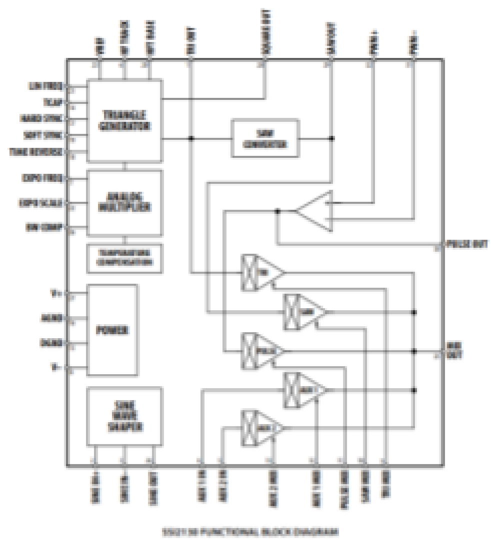


Figure 3.17: SSI2130 Chip

Similar to the AS3340, this triangle core IC provides excellent temperature stability compensation with minimal additional components. With trimmers at the exponential frequency converter, 1 volt/octave tracking can be achieved. Two key advantages stick out about this chip: its low power operation at +/-5V, sine wave generator, and integrated on-chip five-channel mixer. This means each oscillator waveshape has its own designated linear VCA to sum waveforms and control its level on the mix output. Though this all sounds highly appealing, this chip comes in at a high price tag of $8.95 per chip and its 16-lead QFN packaging makes it difficult for breadboarding testing.

3.3.5. VCF

The following table is meant to compare options for VCF integrated circuits. Similar to the VCO, you will notice the same IC product lines. These are two very strong candidates for VCFs. Factors such as their passband distortion, total harmonic distortion (THD), temperature compensation, cost and packaging will be considered in the table below:

Device	***AS3320***	***SSI2140***
Price	3.89	$4.14
Packaging	PDIP-18	SSOP-20
Configurable	Yes	Yes
Passband Distortion	~ 0.1 - 0.3%	~ 0.1%
THD - frequency	0.05 % 0.1 % for <10kHz	~ 0.1%
Temperature Compensated	Yes	Yes

Table 3.5: VCF Device Comparison

There are different synthesizer filter types such as: Transistor ladder filters, Diode ladder filters, Sallen-Key filters, or OTA filters. A high order low pass filter can be implemented using Op Amps, Resistors and Capacitors however VCF Integrated Circuits are available on the market and produce a more efficient output, less noise, are configurable, low pass band distortion,etc. We believe that purchasing and configuring a VCF IC is much more reasonable by means of efficiency and performance.

The two types of VCF IC's of interest are SSI2140 & AS3320. AS3320 seems more suitable since there are more examples shown of how it's configured. With regards to the total performance of the device they seem to perform about the same. The AS3320 might have a small upper hand on the THD at frequencies lower than 10KHz however the passband distortion for the SSI2140 seems more stable. However the same reasoning for the VCO is also used on the VCF IC which is that the SSI2140 is come as a SSOP-20 packaging which adds another step to the testing of this component it is much more convenient to pick the AS3320 due to its packaging and testing it on a breadboard.

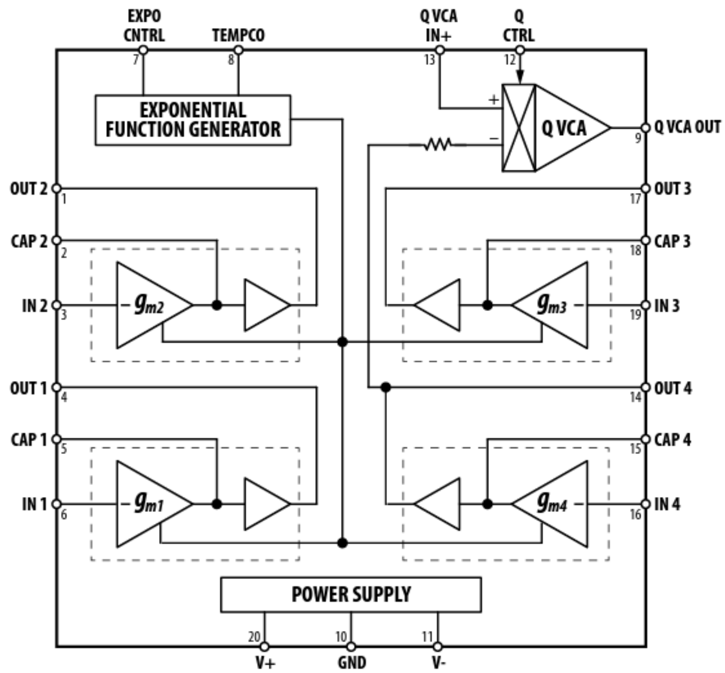


Figure 3.18: AS3320 Functional Block Diagram

3.3.6. VCA

The following table goes into detail in discussing options for the VCA integrated circuit between the AS3330 and SSI2162.

Device	**LM13700**	***AS3330***	***SSI2162***
Amplifier Class	AB Config.	B & A	A & AB
Distortion	< 1%	~0.1%	~0.025%
Packaging	PDIP-16	PDIP-18	SSOP
Control Range	>30 dB	120dB	120dB
Operating Temperature	0 C - 70 C	-25 C - +75 C	-25 C - +85 C
Price	$1.36/Free	$6.25	$6.55

Table 3.6: VCA Device Comparison

The parameters chosen are relevant to the synthesizer we are building. We weighed out their distortion percentage values, control range, operating temperature, and most importantly – amplifier class type, price, and packaging.

AS3330

The AS3330 is a dual voltage controlled amplifier and its predecessor would be the CEM3330 dual voltage controlled amplifier. Great features of this VCA IC is that it provides exponential and linear control over a range of 120 dB. It provides a current control scale for gain tracking which is less than -90 dB. This IC presents a distortion of less than 1%, operates from Class B to Class A amplifier, and can be used as a VCO and VCF.

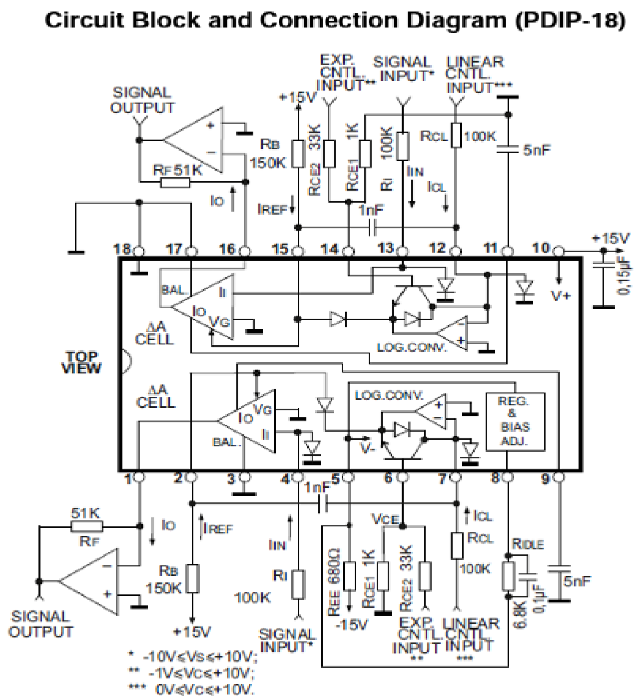


Figure 3.19 AS3330 Chip Configuration

SSI2162

Whereas the SSI2162 boasts of a pin selectable Class A or Class AB amplifier and has a lower distortion than the AS3330. Everything aside is the same.

Now in terms of characteristics of each VCA there are tradeoffs. The SSI2162 performs better but by nature consumes more power due to its configuration assimilating more of a class A amplifier role however the distortion of the signal is far less. However the AS3330 IC holds a reputation for being integrated in many famous synthesizers such as the Sequential Prophet 5 and T8 ,Oberheim OB-Xa, OB-8, Roland Jupiter 6 and the MKS-80. The AS3330 IC has the same configuration as the CEM3330 thus more sources for this IC are available online which makes it a great alternative in case the assembly process becomes a higher priority than the characteristics of the IC.

In the figures the circuit block diagrams for the AS3330 (right side) and SSI2162 (below) are shown:

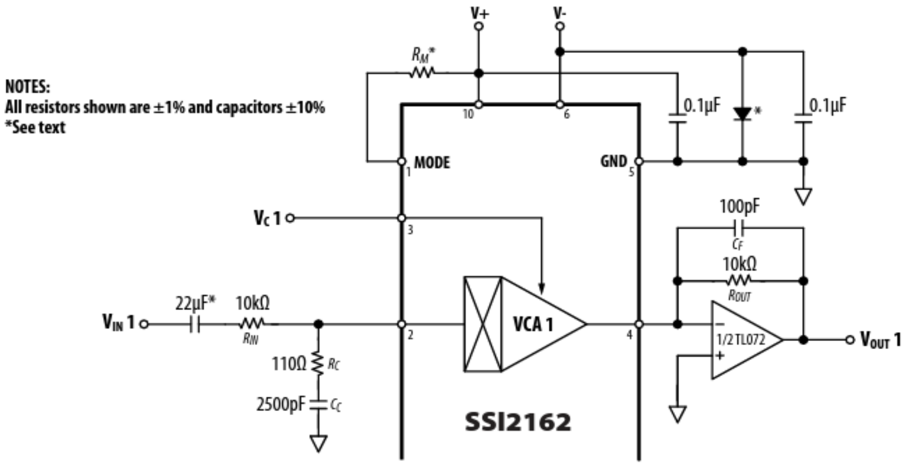


Figure 3.20: SSI2162 Configuration

** Talk why we choose LM13700… Reflection? **

3.3.7. Envelope Generator

The following table serves to compare the last of our synthesizer modules.

Device	*AS3310*	*ENVGEN8C*
Price	$4.40	$4.60
Packaging	16-pin DIP	16-pin DIP
Full ADSR Envelope	Yes	Yes
Time Control Range	50,000:1	10,000:1
Time Control Response	Exponential	Logarithmic
Gate and Trigger Avail?	Yes	Yes
Sustain Levels	0 to 100%	0
Isolated Inputs	Yes	Yes

Table 3.7: Envelope Generator Device Comparison

AS3310

Manufactured in Riga, Latvia is the AS3310 from the AS Integrated Circuit series – a pin-for-pin swappable replacement for the famous CEM line. Its original CEM3310 was used in well-known synthesizers of the 80s like the Moog MemoryMoog, Oberheim OB-X, and the Sequential Circuits Prophet-10. We really liked this historical aspect of the chip as we are striving to create a nostalgic, retro synth that fits in with modern-day setups. The AS3310 is a voltage-controlled ADSR envelope generator that generates full ADSR envelopes with a true RC response. It has a 50,000:1 control range at minimum, meaning that we will have exponential voltage control down to 1 millisecond over 50 seconds for an envelope. This IC will allow us to produce an interesting ADSR envelope over a more than adequate time window.

All four inputs for Attack, Decay, Sustain, and Release can be controlled with simple linear variable resistors. These four inputs are isolated from the rest of the internal circuitry so that control pins (gate and trigger) can be tied together. This allows us to control our ADSR envelope by either gate or trigger since they are conveniently independent. With this IC, we are able to manipulate the sustain level from 0 to 100% of our peak voltage V_{peak} such that we can sustain any voltage level of our attack level. This IC comes in a 16-pin through-hole packaging at a moderate price of $4.40. We are likely using this IC due to its popularity in famous synthesizers and sufficient time control range.

ENVGEN8C

Developed by Electric Druid, the ENVGEN8C is a revision to its predecessor ICs the VCADSR-7B and LOOPENV-1B. It is a less well-known chip and comes in a 16-pin dual in-line through-hole packaging. What makes it distinguishable is its microprocessor topology and fully voltage-controlled looping abilities. This means the chip is capable of full ADSR abilities, looping, and LFO looping. Though these features sound nice on paper, they are a bit excessive in terms of what we are looking for in our envelope generator. The chip has a time control range of 10,000:1 – meaning that it also has a fast 1 millisecond attack time (like the AS3310) but over a span of ten seconds.

The ENVGEN8C also has full voltage control over all ADSR parameters and also features a TIME CV pin for envelope and LFO modulation. This allows you to shorten the entire ADSR envelope and perform frequency and shape modulation on LFO waveforms. Again, though these features are nice to have, they are beyond the scope of what we want to do with our envelope generator. Keep in mind we are trying to keep our synthesizer sound processing module design traditional but still integrable into electronic production setups, thus we will be applying ADSR to our VCA and maybe VCF too if we have time. This IC has logarithmic control over its time control range making it controllable via linear potentiometers. It has similar isolated inputs making it controllable by either gate, trigger, or both. At a price of $4.60 a piece, though the ENVGEN8C has increased functionalities, it is beyond what we are looking for and not a popular choice in the synthesizer DIY community.

3.3.8. Resistor and Capacitor Types

Resistors

When selecting resistors for audio applications, parameters such as thermal noise and current noise are important to reduce noise and distortion. Bulk metal resistive elements like foil and wire wound offer the lowest level of noise; however, wirewound is avoided due to its inductance which can lead to chopping of signal peaks, making foil a more preferred choice.

Carbon composition resistors are generally used for audio applications because they have low noise levels and operate at low frequencies. These resistors also have good linearity and can be used to create precise voltage dividers or current regulators. However, carbon composition resistors tend to drift over time due to temperature changes causing them to become less accurate as time passes. They also suffer from poor power handling capabilities making them unsuitable for high-power applications such as amplifiers or power supplies.

Metal film resistors offer better performance than carbon composition types but are more expensive due to their higher precision levels and better temperature stability characteristics compared with other resistor types. Metal film resistors feature very low thermal noise levels which makes them ideal for use in sensitive audio circuits where even small amounts of distortion need to be avoided at all costs. Additionally, metal film resistors have excellent long-term reliability with minimal drift over time meaning they’re well suited for use in critical circuits like oscillators or filters that require accuracy over extended periods of operation without adjustment or recalibration being necessary every now and then.

Wirewound resistors are designed with higher wattages than other resistor types so they’re capable of dissipating much more heat before failing when compared with other resistor varieties; this means war wounds can handle large currents without suffering from excessive thermal rise temperatures which could damage some components on the board while operating under normal conditions if left unchecked (for instance). The downside however is that this type of resistor tends to be noisy due its inductive nature since it's constructed using a coil wrapped around a core material like ceramic, mica etc leading to increased distortion levels if not taken care properly during design phase. The ideal type of resistor for building an analog synthesizer would depend on what kind of sound you're trying to achieve as each will affect the tone differently depending on how it's implemented within your synth design but overall metal film ones will generally offer superior results especially when dealing with sensitive signals like those found in an analog synthesizer setup requiring extra attention regarding noise/distortion caused by electrical components inside circuit itself.

In order for there to be less noise inside the circuit we generally want to pick resistors that have a tolerance of 1% or less. Picking resistors of this category will help account for small parameter changes of the circuit such as temperature, aging, humidity, etc.

Capacitors

When picking the capacitor for a synthesizer there are a couple of considerations taken which can be Total Harmonic Distortion(THD), temperature stability, timing stability, tolerance. Using high quality capacitors helps reduce the degradation of the signal. In the case of a synthesizer we want a capacitor with a low THD which depends upon the dielectric material of such a capacitor. Typically polyesters and electrolytic capacitors offer the lowest THD in class. For maximum performance of the synthesizer’s VCO and VCF.Polypropylene capacitors should be used since they are appropriate for high precision applications. However ceramic capacitors are fit for the application as well however they offer a higher THD.

Polyester capacitors are the most common type of capacitor used in analog synthesizers. They have good temperature stability, low THD and a wide range of tolerances. However, they can suffer from aging over time which can lead to degradation in performance. Polypropylene capacitors offer superior sound quality due to their higher precision, low distortion and excellent frequency response but tend to be more expensive than polyester capacitors.

Electrolytic capacitors are also popular for use in analog synth designs because they provide high capacity at a relatively low cost compared to other types of capacitor. They have good tolerance and temperature stability but tend to suffer from leakage current which can cause signal distortion if not managed correctly.

Ceramic capacitors are known for their small size and fast response times which make them ideal for applications such as envelope generators or oscillators where timing is critical. While they do offer lower THD than electrolytic or polyester types, ceramic caps tend to be less stable with respect to temperature changes leading to increased noise levels over time when used in audio circuits.

The ideal capacitor for an analog synth design depends on the specific application it's being used for; however, generally speaking polypropylene would be considered the best choice due its high precision and excellent frequency response characteristics while still offering reasonable cost-effectiveness when compared with other options available on the market today.

In regards to other components such as the VCO, VCF, VCA, envelope generators, power supply and signal mixer. These can be made from scratch and designed their own individual schematic however it is more intuitive to buy an integrated circuit since these have been researched and tuned over the years and output greater or equal desired output.

The following table compares multiple capacitors and their uses:

Type	MLCC (Multilayer Ceramic Capacitor)	Polypropylene Film Cap	Styroflex (Styrene) Capacitor
Benefit	Timing Stability, Temperature Stable, Cheap	Timing Stability, Temperature Stable at High Temperature, Long Term Capacitor, Small Dielectric Loss, Suitable for Large Capacity,	Temperature stable, timing stable, low distortion factor long term capacitor
Negative	$$ for values < 100pF	Size	Size
Use-Case	VCO, VCF, Clock Timer	VCO, VCF, Clock Timer	VCO, VCF, Clock Timer
Lowest Tolerance	5%	1%	2%
Best	PCB Design	High precision	PCB Design

Practice	Prototyping	applications	

Table 3.8: Capacitor Comparison

3.4. Possible Architectures and Related Diagrams

Power Supply

In terms of the power supply we are presented with different options such as a Linear, Switched-mode (SMPS), and a Hybrid (SMPS with a Linear Post Regulation)

[16] While Linear Power Supplies tend to be less efficient than SM Power Supplies, they do offer better performance in terms of energy dissipation. In terms of efficiency a linear power supply tends to operate around 60% efficiency whereas a SMPS operates about around 80%. SM Power Supplies are cheap, efficient, and lighter than linear power supplies however the noise produced by the switching makes it less suitable for the synthesizer and makes the Linear Power Supply more suitable for an audio based application such as the synthesizer.

The synthesizer would benefit from incorporating a linear power supply since linear regulators have excellent rejection of AC line ripple and tend to be less noisy than SMPS.

Half-Wave Rectifier

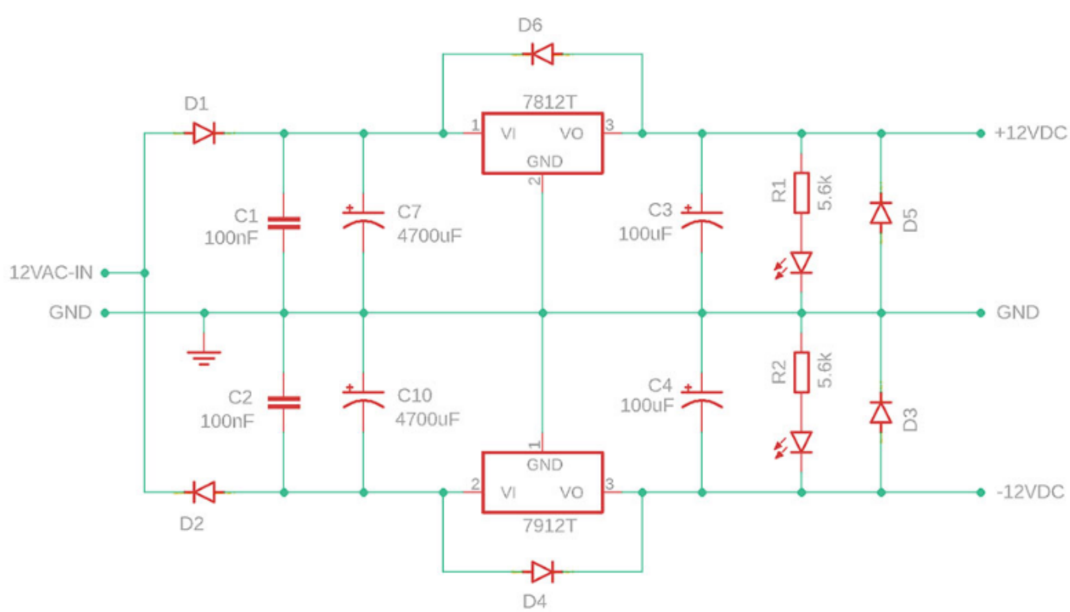
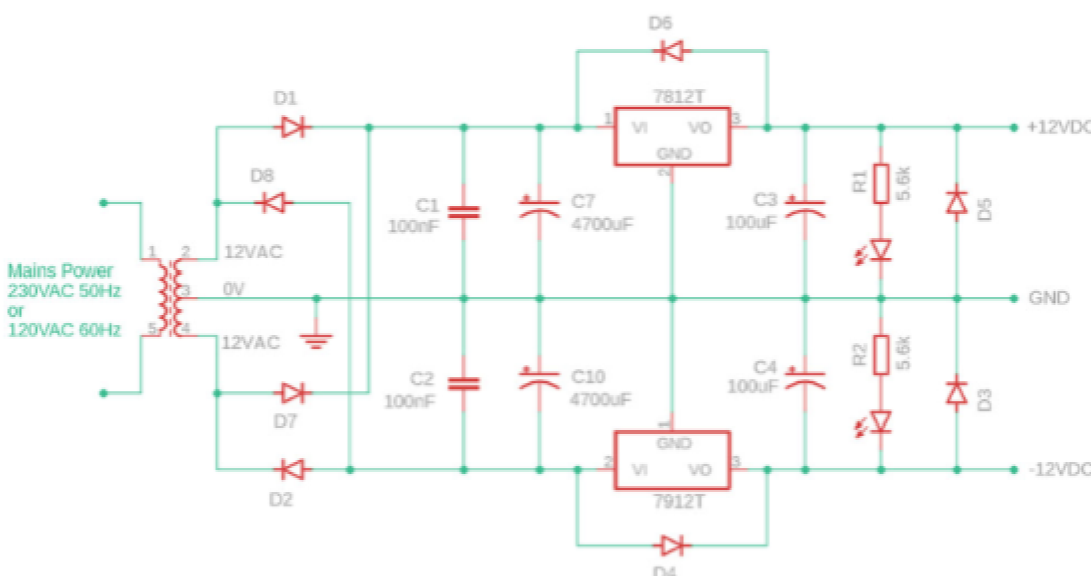


Figure 3.21: Half Wave Rectifier Configuration

The other option for this would be building a half wave rectifier circuit and buying a commercial wall plug transformer that would do the AC voltage step down conversion which is then used by the power supply circuitry. By the means of using a half-wave rectifier circuit it would make the assembly a bit safer to work with.

A halfwave rectifier only lets half of a cycle of the AC voltage waveform pass and blocks the other. They are usually used in converting AC to DC voltages and at a bare minimum only require one diode to achieve a pulsating half wave DC output. Half-wave rectifiers are simple because of their lower number of components involved and are cheaper because of the same reason. There is a drawback to this though. They are cheaper initially because of their lower number of components but can cost more in the long run due to their increase in power losses over a full wave rectifier circuit.

Full-Wave Rectifier



Schematic 3: *Full-wave Rectification Circuit*

Figure 3.22: Half Wave Rectifier Configuration

[17] A full wave rectifier would ideally be the ideal rectification circuit used in the synthesizer since this is the type of circuit used for professional synthesizer equipment. Full wave rectifier power supply designs are often used in analog synthesizers to provide a DC voltage from an AC source. This type of circuit uses two diodes, one for each half cycle of the AC input signal, with the output being taken between them. The advantage of this design is that it ensures a

smoother and more consistent supply voltage than other types of power supplies due to its ability to fully utilize both halves of the input signal. Additionally, because full wave rectification requires fewer components compared to other types of circuits, it can be more cost effective for some applications. Finally, this type of circuit can also reduce electrical noise which may interfere with audio signals by providing a cleaner source voltage.

The concern with building this type of circuit is the risk of electrocution therefore safety equipment should be used in the making of this circuit or at least prototyping of it unless designed and made on a PCB-A.

Dual Power Supply

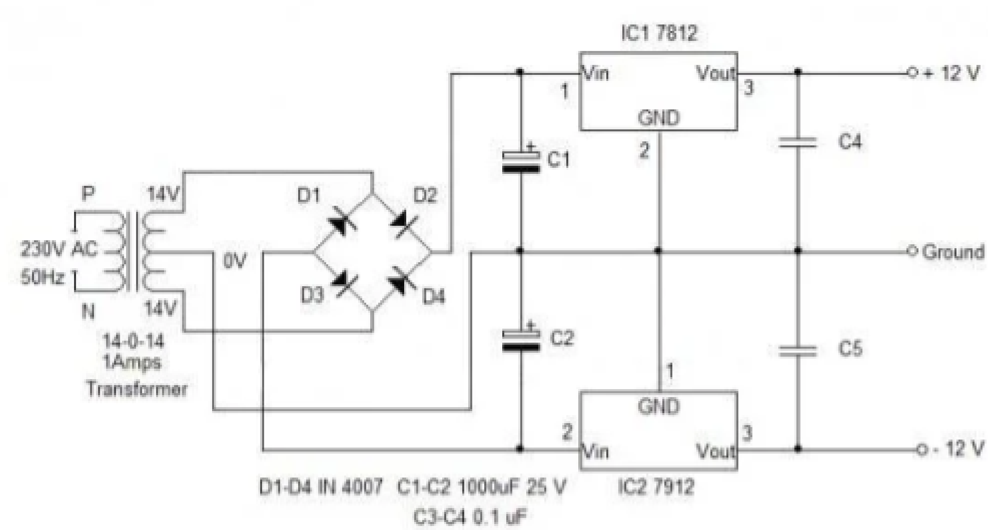


Figure 3.23: Dual Power Supply

[18] Above is a typical dual output power supply. Dual output power supplies are very common in audio circuits as most ICs and operational amplifiers seen throughout function off positive negative voltage rails. We will be implementing similar circuitry as seen above with the use of an AC output wall wart at +/-15V such that in conjunction with rectifier diodes, capacitors, and voltage regulators we are provided +/-12V.

Overall, analog synthesizers require a reliable and consistent power supply in order to function optimally. Dual rail power supplies are well suited for this purpose as they provide two separate outputs, one positive and one negative, that can be used to drive different parts of the circuit. This allows for more flexibility when designing circuits and can help reduce unwanted noise or interference from other components in the system. Additionally, these types of supplies tend to have better ripple rejection than single output designs which is beneficial when dealing with sensitive audio signals.

VCO

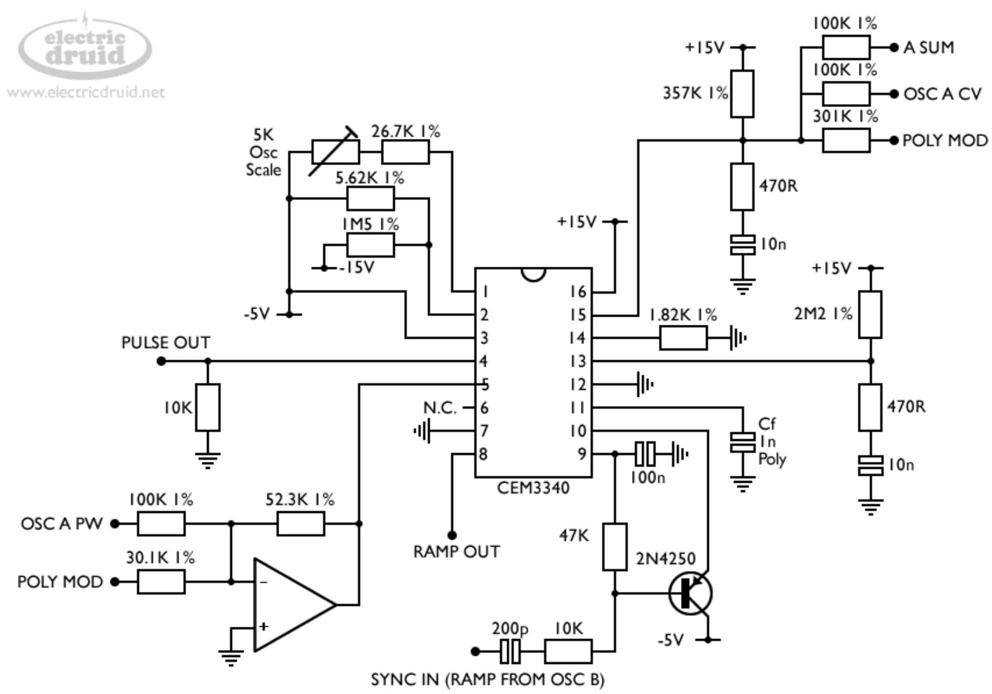


Figure 3.24: AS3340 VCO Circuit (e.g Sequential Prophet VCO model)

This is a potential candidate for our VCO design. Notice the IC name being CEM3340. Recall that this is the precursor IC to the AS3340 that we will be using in our project but is one-to-one identical. Originally made by Dave Smith's Sequential Circuits, the Sequential Prophet 5 is one of the most classic synthesizers of the 1980s. It has five-voice polyphony with two oscillators per voice. Though we are not striving for polyphony to this level and are striving to build a monophonic synthesizer, we were able to take inspiration from the design.

The Sequential Circuits Pro-One VCO provides plenty of flexibility in terms of tone shaping capabilities while still maintaining its classic sound characteristics beloved by many synthesists worldwide thanks to its simple yet effective circuit design combined with modern updates for improved performance.

What we really were interested in was the individual voices of the Prophet-5. This led us to Sequential Circuits Pro-One. This synthesizer was released after the success of the Prophet 5 and is essentially a monophonic version of its counterpart – known as the best vintage monophonic synth. Essentially, the Pro One is a single Prophet 5 voice and the design shown above is one single voice. Due to this difference, we disregarded certain things in the design above like Poly Mod in and out. It uses the same dual oscillator topology that we are going for and offers all waveshapes. The second oscillator can be used as an LFO which is something we are considering.

3.5. Parts Selection Summary and BOM

Below is the tentative part selection and the bill of materials for the parts we plan to procure for our design. If a part number is not specified, it is either a generic, commonly found part or has not been finalized on.

Name	Part	Qty.	Price	Packaging	Manufacturer
			MCU		
Arduino Uno Rev3	ATmega328P	1	$20.70	N/A	Arduino
ATtiny85					Digi Key
VCO	AS3340	3	5.36 x 3 = $16.08	Through-hole	Alfa
LFO	XR2206	1	$4.60	Through-hole	T&F
VCF	AS3320	3	3.75 x 3 = $11.25	Through-hole	Alfa
VCA	AS3330	3	5.35 x 3 = $16.08	Through-hole	Alfa

Table 3.9: MCU BOM

Name	Part	Qty.	Price	Packaging	Manufacturer
			Misc Parts		
Potentiometer	Pot	Kit	$18.99	Through-hole	OCR
Resistors	R	Kit	$12.99	Axial	
Ceramic capacitors	Cap	Kit	$12.99	Radial	
Electrolytic capacitors	Cap	Kit	$16.99	Radial	
Jumper Wire	Wire	Kit	$9.99	Kit	

Cables	Connection	N/A	N/A		

Table 3.10: Miscellaneous Parts

Name	**Part**	**Qty.**	**Price**	**Packaging**	**Manufacturer**
Power Supply					
Volt regulator	LM3171T	1	N/A	Through-hole	STMicroelectronics
Volt regulator	LM337	1	N/A	Through-hole	STMicroelectronics
Electrolytic Capacitor	2200uF	2	4.38 x 2 = $8.76	Radial	Vishay
Electrolytic Capacitor	10uF	2	4.97 x 2 =	Radial	Vishay
Resistors	Various	N/A	N/A	Through-hole	

Table 3.11: Power Supply BOM

Name	**Part**	**Qty.**	**Price**	**Packaging**	**Manufacturer**
MIDI to CV					
Resistors	Various	N/A	N/A		
Diode	1N4148	1	$0.20	Axial	TubeDepot
Capacitors	100nF	2	in kit	Radial	
Opto coupler	6N137	1	1.82	Through-hole	Vishay
Microcontroller	ATtiny85	1	$5.85	Through-hole	Microchip Technology
DIN socket	5-pin DIN Female	1	$1.49	PCB mount	Jameco

Table 3.12: MIDI-to-CV BOM

Name	**Part**	**Qty.**	**Price**	**Packaging**	**Manufacturer**
VCO					
Resistors	Various	N/A	N/A		
Trimmer	10kΩ	1	$2.22	Through-hole	Vishay
Potentiometer	100kΩ	1	in kit	Through-hole	
Capacitor	Various	N/A	N/A	Radial	

VCO IC	AS3340	1	5.36 x 3 = $16.08	Through-hole	Alfa
Resistors	Various	N/A	N/A		
Potentiometer	100kΩ	2	in kit	Through-hole	
Op amp	TL072	1	N/A	Through-hole	TI
Switch		3	$5.99	Through-hole	
Potentiometer	100kΩ	1	in kit	Through-hole	

Table 3.13: VCO BOM

Name	**Part**	**Qty.**	**Price**	**Packaging**	**Manufacturer**
VCF					
Resistors	Various	N/A	N/A		
Capacitor	Various	N/A	N/A		
TL082		2	N/A		

Table 3.14: VCF BOM

Name	**Part**	**Qty.**	**Price**	**Packaging**	**Manufacturer**
Envelope Generator					
Resistors	Various	N/A	N/A		
Capacitor	Various	N/A	N/A		
Variable resistors	100kΩ	8	$5.40	Through-hole	
ADSR IC	AS3310	3	4.40 x 3= $13.50	Through-hole	Alfa

Table 3.15: Envelope Generator BOM

Name	**Part**	**Qty.**	**Price**	**Packaging**	**Manufacturer**
VCA					
Resistors	Various	N/A	N/A		
Capacitor	Various	N/A	N/A		
Trimmer resistor	50k	2	In kit	Through-hole	
Trimmer resistor	1k	1	In kit	Through-hole	Alfa

Table 3.16: VCA BOM

4. Related Standards and Realistic Design Constraints

The following sections will introduce all the standards that have relevance to the project and must be followed. These standards are important to keeping the project professional and safe. The following sections will discuss the economic, time, environmental, social, political, ethical, health and manufacturability constraints faced regarding the project and how we plan to deal with each of them as a team.

4.1. Standards

The following standards we have chosen and have found to have relevance to the project we are contributing to. The core components have yet to be decided, so we cannot go into a fully detailed discussion regarding the chosen components.

[19] Universal Serial Bus (USB)

Title	USB Implementers Forum
Authors	Apple - Dave Conroy, HP Inc. - Alan Berkema, Intel Corporation - Brad Saunders, Microsoft Corporation - Toby Nixon, Renesas Electronics - Philip Leung, STMicroelectronics - Gerard Mas, Texas Instruments - Anwar Sadat
Year Created	1996
Description	This is an industry standard that establishes standards and specifications for cables, connectors and protocols and communication between, for example, computers and hardware. The USB standards are maintained by the USB Implementers Forum (USB-IF). It was designed to standardize the connections to personal computers (and it sure did). The type of USB we plan on using is the USB-A. The USB A connector is the most well-known and widely used connector type. It's a rectangular shape and can only be inserted one way round. USB A cables and ports are compatible with all

	other versions, from USB 1.0 to the latest USB cable standards.

[20] UART

Title	KeyStone Architecture: Universal Asynchronous Receiver/Transmitter (UART)
Authors	Texas Instruments
Year Created	November 2020
Description	The Universal Asynchronous Receiver/Transmitter (UART) performs serial-to-parallel conversions on data received from a peripheral device and parallel-to-serial conversion on data received from the CPU. We learned in our embedded systems labs how to implement this type of conversion and in our parts comparison will consider using this as transmission may be based on a UART signal in our final project design.

[21] MIDI Specification

Title	The Complete MIDI 1.0 Detailed Specification and MIDI 2.0 Specification is an extension of 1.0.
Authors	MIDI Manufacturers Association
Year Created	1996
Description	The MIDI 1.0 Detailed Specification captured the state of MIDI as of 1996 and is for developers, hobbyists, technicians, and others who want to know about the inner workings of MIDI. The document is an intense overview of the standards, practices, and terminology associated with the generation and transportation of sound using MIDI protocol. The specifications we will be looking at are specifically related to the hardware protocols and standards, software protocols,

	and algorithmic conversion to raw MIDI formats for use in digital practice and API usage.

MIDI Specification

Title	The Complete MIDI 1.0 Detailed Specification and MIDI 2.0 Specification is an extension of 1.0.
Authors	MIDI Manufacturers Association
Year Created	1996
Description	The MIDI 1.0 Detailed Specification captured the state of MIDI as of 1996 and is for developers, hobbyists, technicians, and others who want to know about the inner workings of MIDI. The document is an intense overview of the standards, practices, and terminology associated with the generation and transportation of sound using MIDI protocol. The specifications we will be looking at are specifically related to the hardware protocols and standards, software protocols, and algorithmic conversion to raw MIDI formats for use in digital practice and API usage.

[22] USB-MIDI

Title	Universal Serial Bus Device Class Definition for MIDI Devices, Version 2.0
Authors	MIDI Manufacturers Association
Year Created	1996
Description	As computers have become central components in many MIDI systems, USB has become the most widely used protocol for transporting MIDI data. With the introduction of MIDI 2.0, the USB Implementers Forum's USB MIDI 2.0 working group, headed by members of the MIDI Manufacturers Association (MMA) and the Association of Musical Electronics Industry (AMEI), have updated the USB Class Definition for MIDI Devices. This spec defines how USB transports MIDI data.

5-Pin DIN Electrical Specs

Title	MIDI 1.0 Electrical Specification Update [2014]
Authors	MIDI Manufacturers
Year Created	2014
Description	The MIDI 1.0 Specification includes an Electrical Specification which uses a 5-Pin DIN connector and 5 Volt electronics as was common at that time. The Specification was updated in 2014 to reflect current design requirements such as 3.3 Volt circuitry and reduced RF interference. As a team we have to go through a detailed analysis comparing the 5 PIN port against the USB regarding MIDI transport, so this specification will also be cited in the case we go with this connector.

[23] MIDI Reference

Title	Summary of MIDI 1.0 Messages
Authors	MIDI Manufacturers
Year Created	Not specified
Description	Description: The table in this reference guide lists the major MIDI 1.0 messages in numerical (binary) order, this table was adapted from "MIDI by the Numbers" by D. Valenti, Electronic Musician 2/88, and updated by the MIDI Association. It is a useful overview and not an in-depth specification, but will be used by our group as we intend to convert and interpret a lot of this type of data.

Java (JDK 8)

Title	Java SE 19

Authors	James Gosling, Bill Joy, Guy Steele, Giliad Bracha, Alex Buckley, Daniel Smith, Gavin Bierma.
Year Created	2022-08-31
Description	The Java programming language is a multi-purpose, class based, object oriented language. It is a relatively high level language. The language is normally compiled to the bytecode instruction set and binary format defined in the JVM specifications of this same version. This language is generally proposed to be focused on production rather than research as stated in the specification documents, and as will be used in this project in development of the product in relation to the hardware.

[24] JavaX (Java Sound API Package)

Title	“Trail: Sound” - section of Java Tutorials Documentation and Specifications
Authors	*Authors:* Oracle Primary Author: Cindy Castillo
Year Created	2022-08-31
Description	The Java Sound API is a low-level API for effecting and controlling the input and output of sound media, including both audio and Musical Instrument Digital Interface (MIDI) data. The Java Sound API provides explicit control over the capabilities normally required for sound input and output, in a framework that promotes extensibility and flexibility. As a group we will be using this if we develop the stretch goal of creating a GUI interface to manipulate the data signals into a visualizer.

4.2. Realistic Design Constraints

Every product of engineering is loaded with design constraints before a final product can be delivered. These constraints must be set in order to understand when a project is considered finished, however must be within realistic restraints as to not make a project impossible to complete.

4.2.1. Economic and Time Constraints

First, the main economic constraint of this project is that all funding is coming out of pocket for all four group members. It is realistic to consider that some components will be replaced with better ones as prototyping begins. An example of this would be using inexpensive potentiometers and jumper wires while breadboarding. Once PCB development begins, we would instead choose higher quality components for the end product. The most expensive component of this project was the PCBs/PCBAs since these are sold solely in pieces of three or more.

This alone was a big reason why the testing of each stage of the synthesizer should be thoroughly looked over. It is noted that while breadboarding reduces our economic constraints, shipping for components picked is another factor to consider in our economic constraints since while improving the design there will be shipments at different times. Considering that many electrical components are not available on sites like Amazon with free shipping, this was another added cost to our Bill of Materials.

In addition to the shipping, this ties directly as a time constraint since there is a delay period from which it ships and arrives to the user. In order to reduce this time constraint we opted to make the delivery faster however that would add to the economic constraints of the project. To mitigate this cost, we ordered parts early during Senior Design I to ensure we can begin testing and prototyping during this semester. This allowed us to opt for cheaper shipping methods, saving us money and allowing us adequate time to test and prototype during this semester. Doing such had set us up for success in Senior Design II – allowing us to focus on PCB layout and board fabrication. At some point, the group as a whole agreed on a faster shipping in the case of a last minute emergency since it may be a time constraint that will affect the prototyping of the design.

Overall, analog synthesizers are known for their high cost for a multitude of reasons. Their recent rise in the 21st century thanks to the resurgence of retro culture and fascination with revived films, TV shows set in the 80s, and boom in electronic music make for high demand and price increase. The increasing cost

of components – only worsened by the COVID-19 pandemic – makes their circuitry the most expensive part. Many circuit designs are shared and reused amongst each other. For example, VCOs might be based on the same integrated circuit or operational amplifier configuration, where parts can go obsolete and only found on third-party marketplaces such as eBay.
For our project, several of the integrated circuits we planned on using for the sound processing modules have gone obsolete. If the manufacturer had not decided to re-release these chips, we would have to pay around $15 per chip. Thanks to the release they are right around $5 each. Though this might sound expensive for an integrated circuit, the functionality provided is well worth the cost.

With regards to each module for the synthesizer such as the VCO, VCF, VCA, Power Supply, Signal Mixer, etc. There are certain components that should be bought for the sake of time and efficiency. For example, the prototyping of a CGIC Biquad Lowpass Filter would take more time and money than buying an IC VCF. By considering the option of buying an IC can help save time and potentially money since if the prototyping of the module goes wrong there are wide available schematics used with IC of the module.

With such involved analog circuitry and a large quantity of physical components, comes a meticulous manufacturing process which we consider during board fabrication. We had to take into consideration the number of board layers, component quantity, placements, through-hole vs surface mounted parts, thermal issues, signal integrity, testing, etc. All of these come at both monetary and time constraints.

The high cost of analog synthesizers comes at the determination of the market. During their emergence in the 1980s they were expensive due to their complexity and limited audience. As the digital revolution began, digital synthesizers and virtual instruments took over and analog synthesizers nearly completely fell out of the market and fell out of fashion – many collecting dust in studios due to their inconvenient form factors. As the early 2000s began and DJ culture, sampling, electronic music, and retro culture resurged, analog synthesizers had a demand again.

This leads us to the present day. There is a massive market for used analog synthesizers, such as third party marketplaces like Reverb.com. This is an online marketplace dedicated to the buying and selling of new/used/vintage musical instruments. Similar to eBay, Craigslist, or Facebook Marketplace, it is an online store full of music gear where “Keyboards and Synths” are one of the many windows in a mall to gaze into and browse. Upon browsing this website, it is obvious that analog synthesizers are nowhere near cheap today. Though this might come across as an intimidating hurdle for a group of undergraduate engineering students to take on as a senior design project, we took it as a challenge to keep our project low cost.

The time constraints placed upon this project are a very heavy factor in the entire project. It affects building, planning and prototyping more than anything else. It was very essential for each member in the group to meet with the group weekly, and meet their set milestones for documentation, part selection and prototyping. If any milestones were to be missed then the rest of the group is also set back because of the lack of progress. It may even cause severe circumstances such as failing one of the two senior design courses due to not being able to complete the project in time.

If any of these circumstances were to happen the entire project would not be completed, and it would hurt each of the authors in the group. It could result in a complete halt to the project, and with such high stakes such as the milestones and deadlines set by both the faculty at UCF, ABET, and the team members themselves must be taken very seriously at all times. This course is the highest priority for engineers and while it could take a mental toll on both work/life balance it is very fundamental in teaching us the standards and pipeline that we will be thrown into as engineers in the real world.

Each member of the group has signed contracts and jobs post graduation this spring and this project is our final exit interview and will be a core piece of our resume following UCF. Not maintaining a professional status in regards to time management will result in very difficult circumstances and possibly postponing graduation. However, as all of us work extremely well under pressure and together, these time constraints are used to our advantage and failure is not an option in our book.

4.2.2. Environmental, Social, and Political Constraints

It is extremely important for engineers to meet certain environmental, social, and political constraints when designing any work of engineering. Our synthesizer is no exception to these constraints.

Environmental

A lot of consideration went into our parts comparison regarding choosing sustainable and environmentally friendly companies, and parts in the climate we live in today. We can go out of our way when purchasing the parts we need to consider if the companies take a special role in recyclability, working conditions, materials, donations, etc. In doing further research, our team has found a dedicated company that's main mission statement is creating sustainable audio products while leaving no "footprint"(*), Neutral Labs.

Neutral Labs sustainability states that: *"the construction of electronic audio devices for private purposes still leaves a certain footprint regarding carbon dioxide through shipping, hazardous materials during manufacturing, waste through packaging and in other areas. Neutral Labs recognises that far too little is currently being done to combat the climate crisis and widespread environmental damage, and remains committed to help in small or big ways where possible."*

This statement is then followed by what measures the company is taking to follow, including: certified biodegradable paper packaging or reused materials (except for ESD protective packaging), use of renewable raw material (bamboo) for synth cases, 100% carbon neutral electricity in our own lab, carbon neutral shipping, lead-free PCBs and front panels, lead-free and fair solder used for assembled devices (Stannol HS10-Fair using European secondary raw materials and sustainably produced Brazil rosin), USB powered synth means no additional power adapter needed, and many components locally sourced in Europe. These efforts are what make Neutral Labs our top choice for our products in making our analog synth. The main issue is the shipping time and cost, which we will all compare in the parts comparison section.

Our top priority in this consideration is that the PCBs and other parts are lead free, as the U.S Environmental Protection Agency recently stated that lead and lead compounds present major environmental risks. Because lead does not biodegrade, it remains available in the environment as a hazard.

As engineers working with hardware, we realize this is a tougher consideration to balance, but it is growing easier with help from communities. Implementations/guidelines from our team regarding the build, design, and future ventures will be focusing on: recycling as much packaging materials as we can, using paper packaging over plastics, and focusing on repairability. Our schematics will be original but will be focusing on keeping it as simple as possible and the documentation very accurate so the team or a skilled technician can repair it without reordering and wasting the old part(s).

Social

Contributing to the wonderful world of music also comes with social constraints as there is a subset of producers, musicians, and listeners who do not think synthesizers are a great contribution to the industry. Musicians feared in the past being replaced as performers and composers by a digitized process of their work, or ruining the authenticity of sound itself by being able to manipulate it so easily.

The synthesizer can be used to make sounds and creations in the convenience of your home now, compared to the synthesizers created back in the 70s that were the size of a dresser. The social constraints that we face is that musicians even in 2022 still may consider it “offensive” to use a synthesizer to manipulate or imitate an instrument over the real thing.

The social impact of one of the first analog synthesizers, and the most famous, “Moog”, was revolutionary and still continues to be pushing social boundaries every generation. The synthesizer should consider social concerns raised by customers and do relevant market research and testing if our product should ever be released commercially.

Political

The creation and freedom that synthesizers provide as a tool in your music creation or a stand-alone instrument brings forward a freedom to also make political statements. Early synthesizers sought to imitate acoustic instruments, but over time people noticed the unique qualities each one could create and can inspire sounds that bring discomfort to listeners, or create otherworldly sounds. They can even imitate sounds of war, planes crashing, etc. Mainly focusing on how these can “bring discomfort” to listeners, a lot of composers and musicians used synthesizers for political movements including black rights, women's rights, LGBTQ+ rights, underrepresented communities, and much more.

Our analog synthesizer is no exception and may be used to aid in these movements. While this can be considered a positive for our company, being based in the United States today comes with very dangerous water politically and the users of this product can use it for movements that are not representative of the design team’s beliefs. It can be used for offensive music, sound design, films, and much more. Another consideration to keep in mind is that a lot of religious platforms consider certain synthesizer-created sounds “devilish”, and do not allow their communities to partake in the activity. This may prove harmful for our company so we have to consider a very detailed draft of documentation making it known that we do not support such causes, and preparing for such a scene as the U.S is a very politically active country at this time.

We have to consider that as far as we can go to limiting the user from creating harmful media; it is at a certain point out of our ethical responsibility to monitor it as it is also their right to create artistic representations of their beliefs if they deem it necessary, and we cannot control it but discourage any harmful or violent behaviors.

4.2.3. Ethical, Health, and Safety Constraints

Ethics must always be taken into account when developing any engineering system. When obtaining parts or components, there must be considerations taken into where the parts are being sourced from or how much, hopefully ethical, labor is put into the production. Among these ethical requirements, the effects that may pertain from the execution of this project may be seen on a health and safety level. It is important that we exercise caution and implement safety measures to avoid injury.

Ethical

Upon building an analog synthesizer, there are several ethical constraints. These include but are not limited to materials resourcing, employee labor wages, and economical effects. Considering the fact that analog synthesizers are built upon physical electrical components, most electrical parts from DigiKey or Mouser are sourced from China or Taiwan. The semiconductor industry – though a well-oiled machine – is not the most ethical in terms of ecological effects and questionable labor. The rare earth metals and heavily doped materials involved in electrical component manufacturing, mining, and their processing leave a detriment on our environment. Considering the fact that common electrical components such as resistors, capacitors, inductors, op amps, and ICs can be purchased in a large multipack under $10. With such low prices and apparent outsourcing, the electronics industry clearly demonstrates the concept of “no free lunch.”

Companies such as Texas Instruments, Analog Devices, and Micron Technologies almost always have a datasheet stating made in China or Taiwan. These countries are chosen due to their low manufacturing costs, unfair labor wages, and practices. Even if a synthesizer is made in the USA, the internal PCB is still fabricated by a third-party company made with components likely from overseas. Ideally, the way to mitigate these ethical concerns would be to buy second hand retailers and sellers on websites such as those mentioned above like Reverb.com. Even though this would allow us to work around issues mentioned above, it would not be realistic for us to buy all components secondhand. We had to conclude that we will have to accept a degree of uncertainty in the practices behind components chosen for this project since our time and economic constraints of a two semester timeline leave us little to no choice.

Health

There are a number of ethical, health and safety constraints. Health constraints would be taken into account when in the prototyping and assembling stage. Following a standard soldering procedure would be beneficial in order to reduce possible harmful situations since soldering irons reach temperatures of up to 800 F. In addition to this, keeping a proper soldering technique can reduce the risk of skin burns. In terms of materials it would be beneficial to purchase RoHS compliant materials when soldering to reduce the possible inhalation of toxic fumes.

Safety

Since this project deals with analog circuitry safety is a big concern. Since we will be dealing with live voltages and currents, there is a high risk of shock and burn. Even during our undergraduate studies, we have witnessed things like capacitors and operational amplifiers blowing out or resistors overheating in laboratories and want to avoid that in the future. There is the possibility of things going wrong if one node is connected to another and in some cases causing the component to melt or explode which occured. Even the simplest breadboarding mistakes can cause harm, so we want to ensure that we give the utmost attention from going from the schematic to the breadboard process.

With respect to the power supply, since this is the only module that deals with high-power analog circuitry it would be best to follow a circuit already implemented since there safety and health risk increases by already immersing into it. Since we used a wall wart which was used to transition from 120VAC to 12VAC output, this ensured that we stepped down the voltage safely without needing to step in and do it ourselves manually because there would've been great risk in doing that ourselves. In addition to this we know that this can be done so by adding voltage regulators and utilizing the Webench application to create a power supply.

Webench is an application we utilized during our Junior Design course for designing power supplies. It aims to make the design process easy by allowing you to enter a number of inputs and outputs, their corresponding Vin and Vout, and maximum Iout. This allows us to choose if we desire if we want our power supply to be balanced, low cost, efficient, or a small footprint while ensuring the safety of our group members.

With this project, the following modules were breadboarded before beginning PCB layout: VCO, signal mixer, VCF, VCA, and envelope generator. This means each module was successfully breadboarded individually and then integrated. Considering we have never worked with these designs before, we have explored new territory for our electrical engineering and breadboarding skills. The large majority of this process occured in the Senior Design lab in Engineering Building I (room 456). Thanks to having previous labs in similar rooms, we have a degree of familiarity with the lab equipment provided.

On top of exercising diligence and attention to detail when building these circuits, we must also take careful considerations when it comes to electrostatic discharge (ESD) when handling and building upon these components. It is imperative that semiconductive components such as integrated circuits and transistors are grounded and oriented correctly when supplied. We followed ESD procedures such as wearing an ESD wrist strap and grounding ourselves before handling such sensitive devices. The carpeted floors of this room can build static and took precautionary measures to ensure our success, otherwise time and money are going down the drain.

4.2.4. Manufacturability and Sustainability Constraints

Manufacturing

One of the things we learned regarding standards is that they do not measure the product quality, but rather they follow to see if they meet process-based requirements. This is a prerequisite for good quality products, even as a “quality stamp”. Manufacturability is considered to be the art of designing a device or product that can be easily constructed or mass-produced. The main manufacturing goals that we kept in mind are fabrication, assembly, testing, acquisition, shipping and repair. These are a lot of factors, and we might not be able to effectively reach the best possible outcome of all these goals but we will try to maximize it to our greatest efforts and goals. Our main goal was to design a product that does not require many cost intensive repairs, and with the strategic part selection we have done a very thorough job of keeping everything cost effective and the products/parts used kept in high stock. The project should take at most a few hours to piece together if bought as an assembly kit, as well.

In most of our breadboard designs, we noted that while the logic, design and main components took a long time to consider and research from our team, with our detailed documentation it should not be very difficult to replicate. In fact, we want as engineers to show as much openness to our customers without violating our intellectual property so that they can get the most benefit from the product. In regards to manufacturers, we want to make their job simple and easy to replicate in the case of mass production. If a component becomes obsolete for example, like our DIN-MIDI port is no longer the standard and doesn't work with any DAW system anymore, it might be a hassle to change this about our core design. This is because of the programming required on our MIDI-to-CV converter and our chosen microcontroller reading all the UART signals. This is not the case with other parts, however that can be changed and replaced without doing much damage to the entire internal system.

A major concern for us regarding manufacturability is the ease of use for the user and sustainability of the products. The lifespan of our chosen components is a major consideration for our parts selection as the user will be making use of each component in our project. It is not very customizable and/or pieces left idle or unhandled, like features on your phone that you never use. In this market today, the cost of products is continuing to increase slowly, but soon it will be a very considerable price. Will the consumers continue to pay for the product regardless of the market increase, or go with a much simpler product that does the same functions for cheaper? What makes our product stand the test of time?

One of the most important and favorable assets to our design and sustainability to all electrical/hardware components is creating an enclosure to protect the PCB and surrounding parts. This will make sure after soldering on all the parts, they will remain intact and safe from wear and tear. It will probably be 3D modeled and printed or cut using a laser cutter utilizing an even sturdier material given the tools we have on hand in the engineering lab. This enclosure will also be favorable because it will make our product a lot more aesthetically pleasing, with curated design choices from our team to meet modern standards and market needs. This will set us apart from other products as we will have a brand to represent, an obvious need from customers when they add this in their musical space/desk.

One thing we want to consider when building the outside box that will enclose all the parts of the synthesizer is manufacturability as we want to look as visually pleasing for the user as well as easy to use.

We are for this reason considering using a laser cutter for the majority of the box, even using a stretch goal to create beautiful labels. These labels would help the user understand what each part does and create an enjoyable experience, making it easier to market the product and create a satisfied finished product for the team as well. The main constraint here would be time, and availability of resources. As seniors we plan on using the senior design lab for most of our creation but finding a laser cutter might prove to be difficult or expensive. Creating a file for the laser cutter might also be using valuable time at the end of the second semester if we could put that time elsewhere.

Sustainability

All of our sustainability constraints were discussed in section 4.2.1 in the environmental section. To summarize our main priorities or implementations/guidelines from our team regarding the build, design and future ventures will be focusing on recycling as much packaging materials as we can, using paper packaging over plastics, and focusing on repairability. Our synthesizer will be designed off of common parts that are far from going obsolete.

We believe that this is the easiest way to make our synthesizer sustainable and replicable in the case that another group member wants to build a duplicate for themselves (since we cannot all take home the end product). Even components like our integrated circuits are recent reissues of a popular product line from the 80s that recently got reissued and will not be going out of production any time soon. We want to use a common microcontroller, operational amplifiers, resistors, capacitors, potentiometers, diodes, etc. to make our design one that can be easily DIY'd by any enthusiast with an electronics background.

On a lower level, we cannot guarantee the sustainability of our components due to their ambiguous origins when we tried researching where several components were made. We do have hopes though that by choosing manufacturers such as Texas Instruments for their ever-popular TL072 and TL082 operational amplifier series, we are supporting a company with a sustainability initiative to reduce their environmental impact. We saw this as an opportunity to make a conscious choice as there are few other areas in our project that we can pick and choose in.

Texas Instruments is minimizing their footprint by making lean manufacturing choices to reduce air emissions, energy consumption, and greenhouse gasses, and building for efficiency [#]. In doing so, they are making chips more effectively by committing to the environment in areas like energy, water, greenhouse gasses, and waste. They are able to do so thanks to a recent legislative decision in the United States known as the CHIPS Act of 2022.

In July of 2022, Congress passed the CHIPS Act of 2022 with a goal of strengthening the domestic semiconductor industry's research, design, and manufacturing. By reinforcing internal supply chains we reap the rewards of an electronics industry that can truly thrive and grow. Since its passing, Texas

Instruments has begun construction of a new fabrication facility (known as a FAB) in Richardson, Texas – a $30 billion dollar build. With demand outgrowing supply and the COVID-19 pandemic making matters worse, this will allow the company to oversee labor and environmental impacts internally rather than overseas in Asia.

We were hesitant when it came to researching semiconductor companies that rule the market but were relieved in educating ourselves on Texas Instruments new practices. Though we cannot make conscious supplier decisions in every aspect of our project for bulk components like resistors or capacitors, we are confident in this one. In the future, we hope to apply the same research to our suppliers and hope to see even more semiconductor giants following their sustainability footsteps.

5. Project Hardware and Software Design Details

In this section, we will be covering the detailed design going into each module of the synthesizer and its corresponding circuitry and software. Hardware modules including the VCO, signal mixer, VCF, VCA, envelope generator, and power supply will include their functional operation, parts selected, and any simulations involved. Software sections will cover MIDI to CV conversion and the user GUI application.

5.1. Voltage Controlled Oscillator (VCO)

The design we have decided on for the VCO is based off of the reference design we mentioned earlier, the Sequential Circuits Prophet One along with studying the AS3340 datasheet connection diagram. Depending on how the prototyping stage goes, we will also be implementing pulse width modulation (PWM) on pin 5. Right now, this design gives us sawtooth, triangle waves, and ramp waves.

5.1.1. Functional Operation

We powered the oscillator on +/-12V rails on pins 16 and 3 respectively. The absolute maximum rated voltage between Vcc and Vee is +24V. We were worried about 12V rails cutting it close to the limit but upon observing the circuit block connection diagram realized there is an internal zener diode on the negative supply pin to keep the total voltage under 24V. The rest of the schematic can be broken down based on features of the VCO.

Pin 1 and 2 are for temperature compensation. This is done by generating a temperature-dependent voltage and multiplying it times the incoming control voltage. In theory and in practice, this is proven the simplest way to compensate for the temperature of the VCO going out of tune. This phenomenon of detuning is usually accounted for by matching two transistors with thermal glue face-to-face in an effort to keep them “temperature matched.”

In our case, the VCO and temperature-compensation circuitry are all on the same chip die, they are all on the same temperature – without any need for matching transistors required – making it highly accurate and fool-proof. The 10k trimmer resistor on Pin 1 is for tuning the VCO to the 1 volt/octave standard which will be explained later on in the User Manual section. A trimmer resistor is used because we hope the user can “set and forget” at least for a few days or between synthesis sections.

Pin 3 as mentioned earlier is the negative voltage supply pin. Pin 5, 6, 7 and 9, are inputs Pulse Width Modulation Control Input, Hard Sync Input, High Frequency Tracking, and Soft Sync Input (respectively) that we chose not to use due to time constraints.

Our waveform outputs will now be discussed. Pin 4, Pin 5, Pin 8 and Pin 10 are outputs corresponding to Pulse Output, PWM Control Input, Sawtooth Output, and Triangle Output. Our Triangle, Sawtooth, and Pulse outputs have switches such that the user can switch them on and off depending on which waveshape they desire. The PWM circuitry uses a TL072 operational amplifier as a buffer as it is commonly used in synthesizer design and a go-to in the electronics industry.

Pin 12 is used as ground. Pin 13 is Linear FM Input which we decided to not use due to time constraints – however, we tied it to various resistors and capacitors as per the datasheet requirement. Likewise with Pin 14 also known as the Scale pin, we connected it with a 1.8 kilo ohm resistor as per the data sheet requirements.

Pin 15 is where the user will be interacting with the pitch of the oscillator – in our case, the frequency. This pin goes by Frequency Control Input (V_{FCI}). This is what is known as Coarse Adjust in most commercial VCOs. We definitely wanted a Fine Adjust as well for a more precise frequency sweep and added that to V_{FCI} as well. The majority of the resistors and decoupling capacitor values shown were deemed necessary from the datasheet.

5.1.2. Parts Used

Basic Ramp Wave VCO

- AS3340
- Breadboard and jumper wire
- Various resistors
- 10K multi-turn trim pot
- 100K potentiometer (linear)
- Various ceramic capacitors
- 1000pF capacitor polystyrene

Components for fine tune control

- 3.3M resistor
- 100K potentiometer

Components for pulse width control and modulation

- Various resistors
- Various ceramic capacitors
- 100K linear pot x 2
- TL072 Op amp

<u>Components for wave forms and output level</u>

- Various resistors
- Slide switch x 3
- 100K potentiometer

5.1.3. Schematics

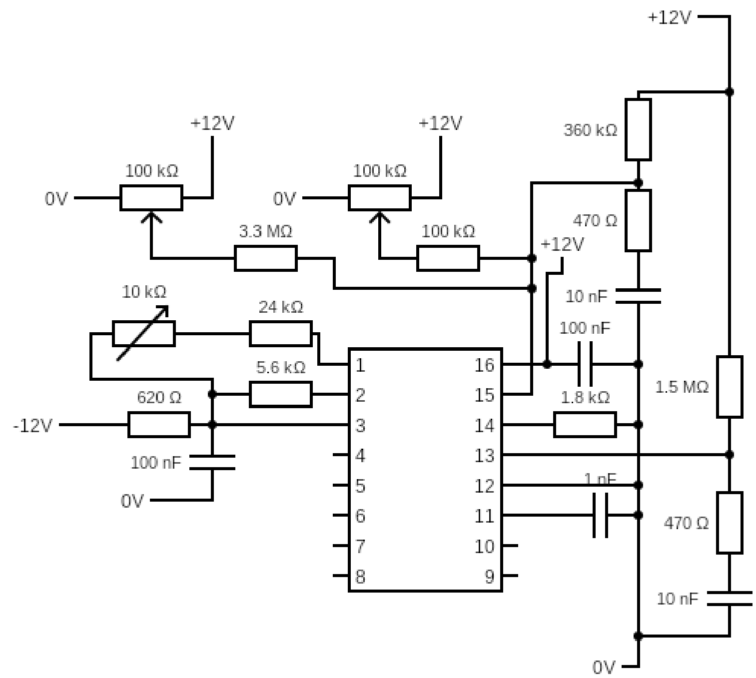


Figure 5.0: [25] VCO Ramp Wave Configuration

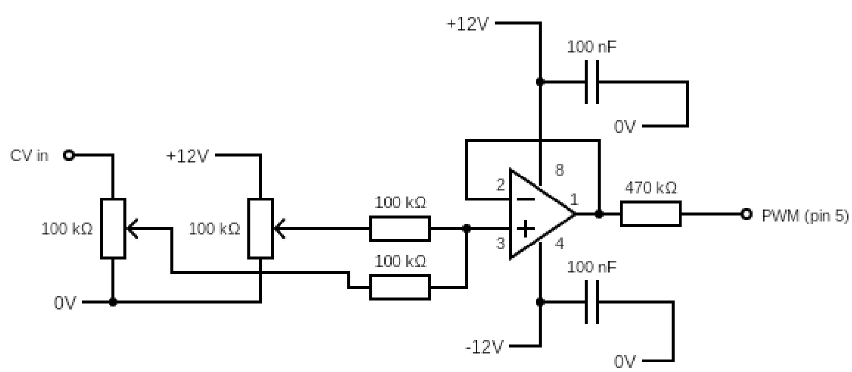


Figure 5.1: [26] VCO PWM Configuration

Figure 5.2: [27] Waveform Selection for Sawtooth, Pulse, and Triangle Waves

5.2. Voltage-Controlled Filter (VCF)

The voltage-controlled filter is an essential part of the development of the synthesizer. Considerations such as having the VCO to the right scale, for example 1 volt/octave, is important. Filters are used to alter the harmonic content of signals coming from the VCO. The hardware we decided to use was the AS3320 IC mentioned earlier.

5.2.1. Functional Operation

The functional block diagram of the AS3320 is shown in figure Figure 3.9. With that in mind we want our VCF to be connected to our VCO in order to see the change in the response. Pin 12 in the functional block diagram is what we regard as frequency control input to an internal generator which is connected to a four gain cell in the filter. There are different applications however for one application one can get away with a 100k/1.8k resistor network which produces a nominal 18mV/volt which gets connected to pin 12. The free-end of the 1.8k resistor goes to ground and provides a mean of control range to the 100k resistor between -1.4V to 8.6 V which is the range for an exponential generator. With this simple approach we can decrease the cutoff frequency by just increasing the positive voltage. An alternative is to add IC1 from Figure 5.1.

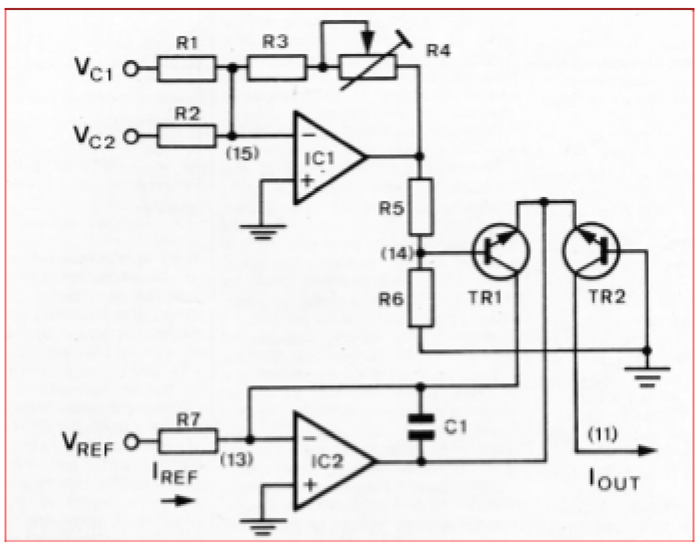


Figure 5.3: Typical Exponential Generator

In addition, it'd be ideal to add each resistor to its suitable value R1 = 100k, R3 = 91k, R4 = 20k, R5 = 56k and R6 = 1k0 with the junction of R5/R6 connected to pin 12. Another consideration to keep in mind is that even though this IC as a whole is temperature compensated the exponential generator is still temperature sensitive however this can be compensated by using a 3500 ppm/C temperature compensating resistor in the place of R6.

The inputs of the four filter stages are pinout 1,2, 17 and 18. Pin 1 is made the first stage that consists of a variable gain cell followed by a high impedance buffer. The first two stages of 24dB/octave low pass and high pass filters are shown in Figure # below.

One thing to keep in mind is that the input of each variable gain cell has a forward biased diode to ground and so provides a low impedance summing node 0.6 V above ground. So the input current can be obtained from the input signal, which may therefore be obtained with resistors terminating at this node.

For a normal operation each stage is set up with a feedback resistor from the buffer output to the input of the variable gain cell to establish a reference current. When no signal at the input the buffer output will adjust itself to keep maintaining its reference current. The lowest voltage control feedthrough for maximum output voltage of each buffer should be 0.46 Vcc which for a +15V Power Supply means 6.9 V. The internal reference current is nominal at 63 microamps at a feedback resistor of 100k. At stage 2 the input current is 0.63 microamps plus the 70 microamps across the coupling resistor between the two stages. To sink this excess a 220k bias resistor is connected to the -15V power supply.

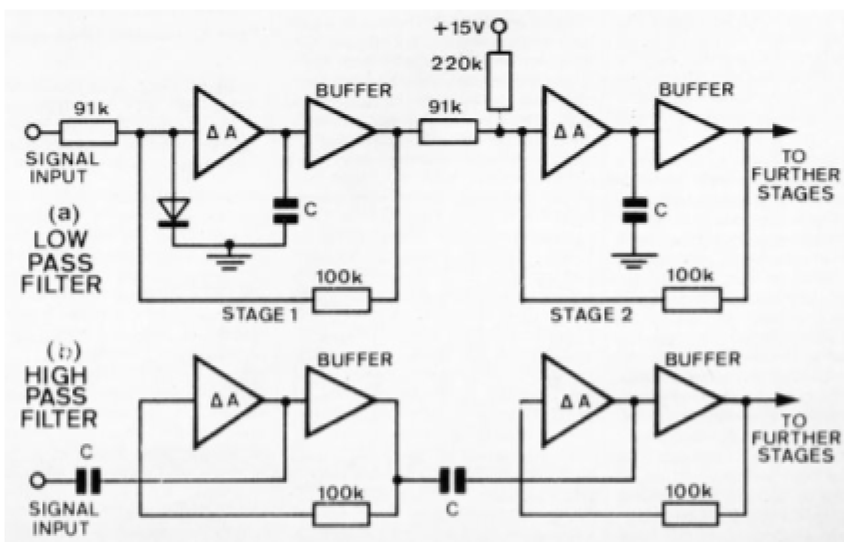


Figure 5.4: Four Stage Variable Gain Cells

Now for the highpass configuration the input stage is coupled with a capacitor which blocks the quiescent voltage of the buffer therefore the required reference current is found solely with the 100k feedback resistor for all stages.
For negative supply voltages greater than -4V a current limiting resistor is placed between the negative supply and pin 13. For a good control voltage feedthrough we use a 1k pot with a 1k resistor in combination with a -15 V power supply.

5.2.2. Schematics

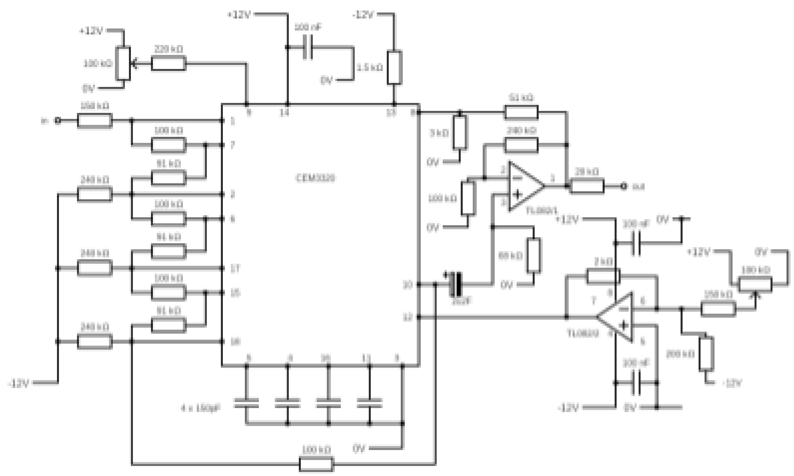


Figure 5.5: Voltage Controlled Filter Configuration

5.3. Voltage Controlled Amplifier (VCA)

The basic functionality of the VCA is to manipulate the amplitude of a signal proportional to a controlled voltage applied to its amplitude modulation control input.

5.3.1. Functional Operation

Now in the AS3330 there are two VCA's in the IC. Linear inputs are 7 and 12 while exponential control inputs are at pin 6 and 14. It should be noted that the linear control inputs are summing nodes that allow any number of signal and linear controlled voltages to be mixed within the IC.

Pin 8 is "Idle Adjust" which offers the feature to alter the quiescent standby current of the signal carrying resistors thus being able to alter the operation of the gain cells between Class A (100uA) and Class B(1uA). For a class AB operating point a 7uA standby current would be done with a 6.8k resistor in between pins 8 and 5. In addition compensation and trimming of inputs are needed. Below in figure 5.6 are some examples.

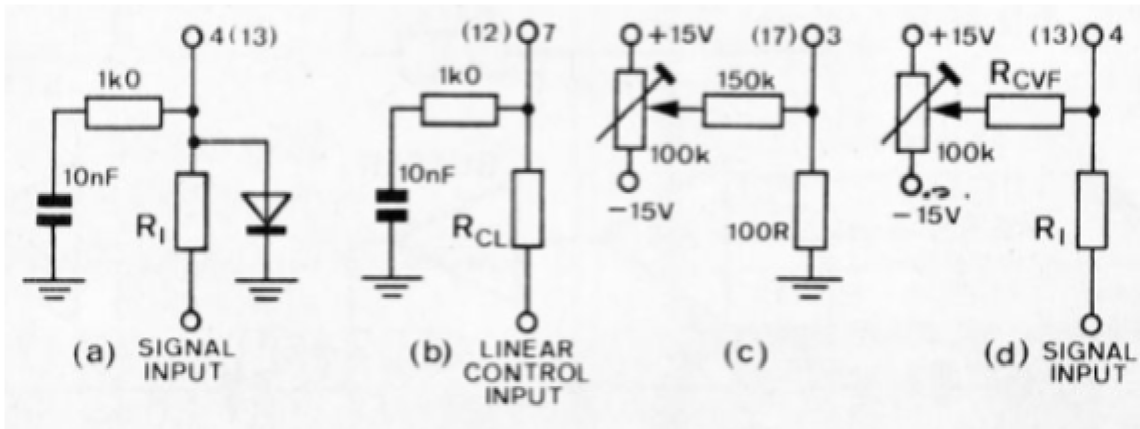


Figure 5.6: Compensation and Trimming of AS3330

In figure 5.6 (a) a diode compensates the signal input to prevent latch up. (b) shows the components used for compensation for linear control inputs and (c) allows for distortion to be trimmed while (d) allows the reduction of control voltage feedthrough.

Using the linear and exponential control it is possible to configure each VCA for exponential response and then incorporate linear amplitude modulation.
Another consideration is if the negative supply is lower than -7.5V then a current limiting resistor should be placed between pin 5 and the negative supply line.

5.3.2. Parts Used

- 5x 100k
- Various Resistors
- 15pF ceramic
- 100nF ceramic
- TL084
- dual-transconductance op-amp
- 4.7 zener
- AS3330

5.3.3. Schematics

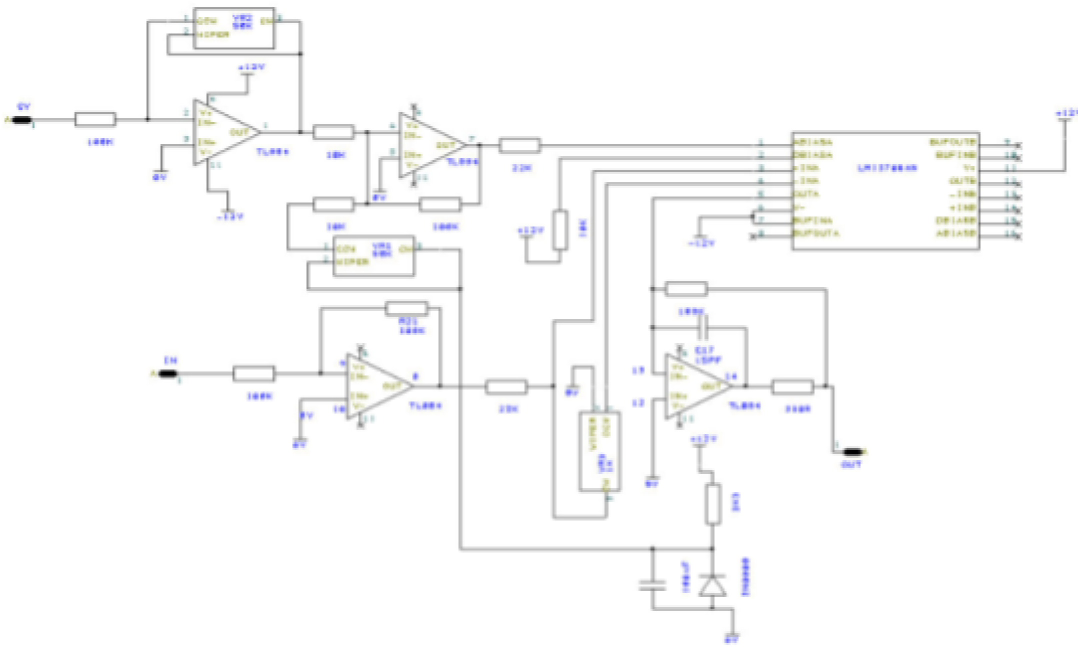


Figure 5.7: Voltage Controlled Amplifier

5.4. Envelope Generator (ADSR)

Our analog synthesizer design consists of two envelope generators. Some analog synthesizers only have one envelope generator but we were able to expand upon our original design to build two to be applied to both the filter and the amplifier to have envelope control over both modules. Similarly to the VCO and VCF, we based our design off of our intentions to emulate the Sequential Circuits Prophet One synthesizer.

5.4.1. Functional Operation

We power the AS3310 envelope generator IC off of +/-12V power rails. The absolute maximum rated voltage for this chip was again +24V, so we were able to use it in conjunction with our power supply design. For added protection, there is a 7.4V zener diode on the negative power output pin 6 to supply a maximum voltage of +15V and a current-limiting resistor. We add current limiting resistors throughout upon calculation using the following formula from the datasheet.

$$R_{EE} = \frac{V_{EE} - 7.5}{0.010}$$

First let us recall that when the user presses a key on our synthesizer, two voltages will be generated – a CV and a Gate voltage. The AS3310 requires two

CVs to provide a full ADSR envelope – gate and trigger voltages. The rest of the design can be explained by covering pin outs of the envelope generator IC.

Pin 1 is simply a 39nF capacitor known as Cx called for by the datasheet. It serves as part of the RC time constant control of the IC and is derived from the following formula that takes into account Rx. Pin 2 is our envelope generator output that we incorporate Rx in series with. In this case, we set Rx to be 24k.

$$R_x \times C_x(e^{\frac{-V_C}{V_T}})$$

The following pins are inputs and outputs. Pin 3 is the Attack Peak Input which will be adjusted with a 100k potentiometer. This lets the musician/user control the magnitude of their maximum peak on the envelope (i.e. their max volume or brightness). [39] Pin 4 is Gate Input and will be connected from the Gate Output of our MIDI to CV converter – also known as Pin 6 out from the ATtiny85 microcontroller. Pin 5 is the Trigger Input which is connected from the Trigger Output of our MIDI to CV converter as well – also known as Pin 5 from the ATtiny85. They are tied together with a capacitor in series as per the datasheet.

Pin 6 is a negative voltage supply which will be -5V. Pin 7 is Power Ground. Pin 8 is called Compensation and is only meant to tie to the capacitor specified by the manufacturer of 22nF for added stability.

Pin 9 is our Sustain Level Control Input and is controlled by the user by another 100k linear potentiometer. This is such that the musician can control how long they want the sound to maintain that volume or timbre. Pin 10 is for Input Current and features the Rx value of 24k mentioned earlier. This value came from interpolating the 30k value used in the datasheet from a 15V design. Pin 11 is our positive +12V supply.

Pin 12 is the Decay Control Input where the user can control how rapidly they want their waveform to decay from the maximum attack value (loudest point) to the sustain level (maintained point). A 470 ohm resistor is used in accordance with the datasheet. Pin 13 is the Release Control Input is configured with identical components but is tied to our negative -12V supply. Here, the user can control the release time with another 100k linear potentiometer in series with a 470 ohm resistor such that they can control how long their sound takes to diminish to complete silence. Pin 14 is Ground. Pin 15 is Attack Control Input to allow the user to control over how “sharp” or “abrupt” they want their attack time for a more aggressive zero to peak volume or brightness depending on if they are applying ADSR to the VCA or VCF. Pin 16 is the Attack Output where we will observe the sound envelope. We use this extensively during testing to view the sound contour and we considered adding an oscilloscope here in our final design so the musician can have a live view of their ADSR shaping the waveform.

5.4.2. Parts Used

- <u>Envelope Generator</u>
- AS3310 x 2
- Breadboard and jumper wire
- Various resistors
- 100k potentiometers x 8 (4 for ADSR of VCA and VCF)

5.4.3. Schematics

Below is the schematic for the Envelope Generator design. It gets its gate input from the MIDI to CV module we built and is based on the external circuitry provided in the AS3310 datasheet.

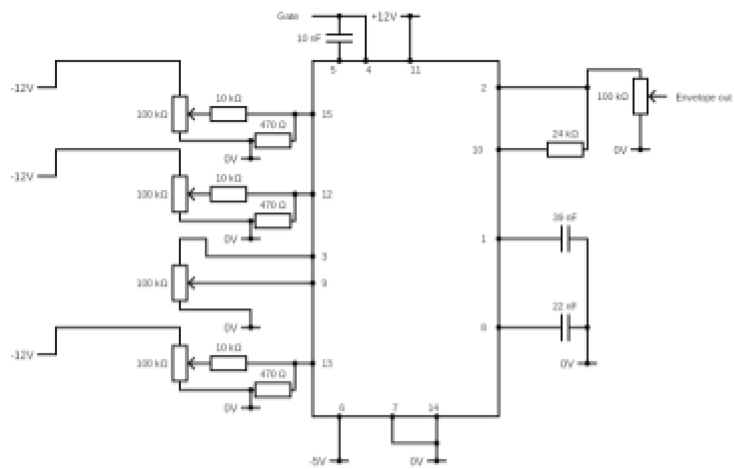


Figure 5.7: Envelope Generator Configuration

5.5. Power Supply

The power supply ensures the activation of IC's and Operational Amplifiers within the circuit. The idea behind having a power supply is to ensure that a steady dual supply +/-12V voltage is maintained around mentioned components as their load fluctuates. Analog synthesizers are most commonly run off of either +/-15V supplies or +/-12V supplies. We chose to +/-12V supplies in the possibility that we would want to integrate our synthesizer into a eurorack setup later on which are always powered on 12V rails. Our power supply design outputs dual DC supplies as many of our synthesizer integrated circuits and operational amplifiers run off of dual voltages.

5.5.1. Functional Operation

At its core, our power supply is a dual variable output +/-12V DC supply. We used a half wave rectifier design – the most popular configuration in synthesizer design. This is because we were able to use a wall wart transformer off the shelf to convert the mains 120VAC to 12V AC for us. Being able to avoid coming into contact with mains power was not only safer for us as electrical engineering students but also saved us on cost and time. We would otherwise have to design even more circuitry with an already involved analog hardware focused project.

Using a 15V AC to AC wall wart we found on an audio amplifier website, we were able to provide the half-wave rectifier its AC input to obtain our positive and negative 12V +Ve and -Ve. This way we only need a single AC input. We will be able to plug the AC output wall wart in and not have to worry about wiring a transformer to convert the mains power down. We plan on plugging in the AC wall wart into a power barrel jack. This way we can plug AC power in directly and make the synthesizer easy to power by “plug and play.”

The rest of the power supply circuitry is relatively straightforward. We used a LM317T voltage regulator as it was an ideal candidate due to its through hole, single output, positive polarity, and 1.2 V to 37 V output range, and 1.5 A current output. We also used an LM237 for its ideal through hole, single output, negative polarity, -1.2 V to -37 V output range, and 1.5 A current output. As these voltage regulators have a tendency to reach high temperatures and have a Maximum Operating Temperature of + 125 degrees celsius or + 257 degrees fahrenheit, we decided to use heat syncs on them. The heat syncs are meant for these voltage regulators and are made from anodized aluminum to draw heat away from the regulator under load.

The rectifying 1N4007 diodes are in flipped orientations such that the top diode lets the positive AC signal pass through and the bottom diode lets the negative AC signal pass through. As a result, we get a full AC waveform. The 2200 uF 35 V rated electrolytic capacitors are for smoothing purposes such that they provide an even and continuous waveform to the regulators so they can properly stabilize the +/-12 V outputs on the power supply. The 10 uF electrolytic capacitors are for refining the transient response of the power supply itself when under high loads and are rated for high 50V voltages. The two 5k trimmer resistors are so that we can set the proper voltage levels. In our case, we want a dual output of 12 V so we will trim them to 15V of AC input accordingly.

5.5.2. Parts Used

Components for AC Input

- US (110-120V) 15VAC Power Adapter
- DC Power Barrel Jack

Half Wave Rectification

- LM317T Voltage Regulator
- LM237 Voltage Regulator
- Heat syncs
- Ceramic capacitors
- Various electrolytic capacitors
- Various resistors
- 5k trimmer resistors x 2
- 1N4007 rectifier diodes

5.5.3. Schematics

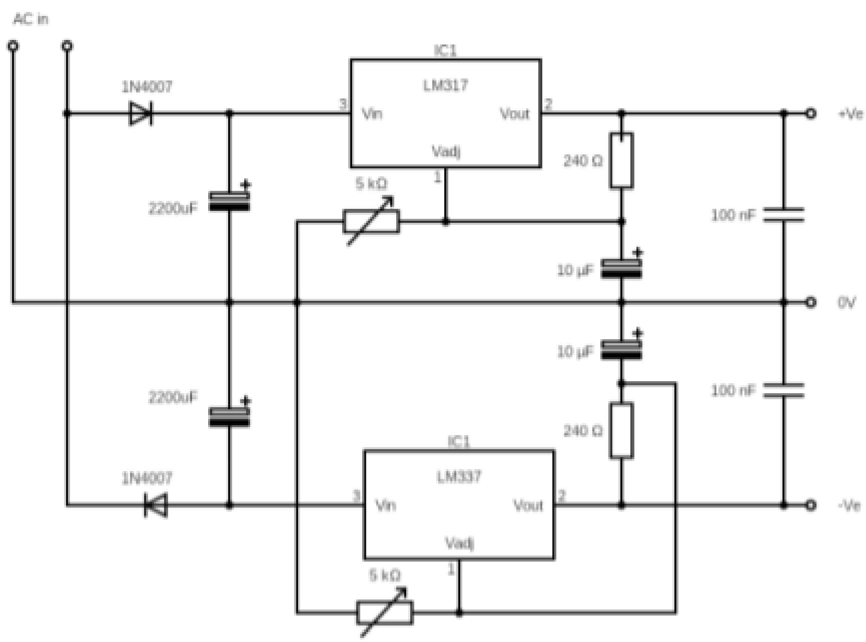


Figure 5.8: Rectification Circuit

5.6. MIDI-to-CV Conversion

The project utilizes MIDI-to-CV conversion technologies by implementing an analog-to-digital conversion system. This is done with the use of a series of microcontrollers and a printed circuit board system.

5.6.1. Protocol Explanation

Musical Instrument Digital Interface, or MIDI, is a standardization of digitizing musical instruments. MIDI works by transmitting a digital signal that represents a sound or a note. This signal contains information that is attached to each note that includes when to start and stop the note, how loud the note is, the pitch of the played note, the beats-per-minute (BPM) of the note and its following notes, and other metadata.

Control Voltages (CV) are the “opposite” of MIDI–that is they process and modify audio signals analogously. CVs, as their name reveals, controls voltages sent by adjusting how much of the voltage is allowed to pass generally in the form of various knobs, switches, or pedals. This restriction allows CVs to adjust the pitch and amplitude of the signals as seen fit and thus adjust their sounds.

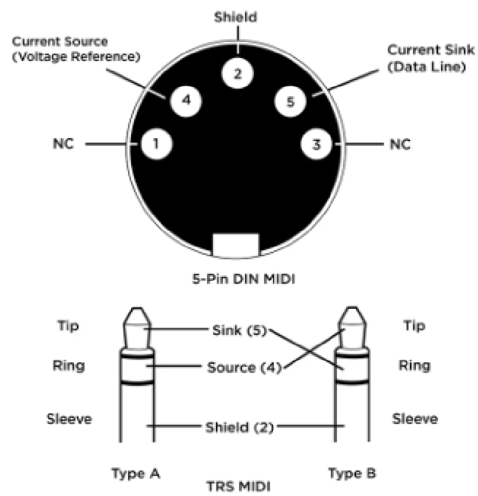


Figure 5.9: 5-Pin DIN Midi Cable Pin Reference

This conversion can be done through the use of 5-pin DIN (which gets its name from the German National Standards Organization) connections and more recently through Universal Serial Bus (USB) connections. These connections transform the digital MIDI data into analog voltages that the synthesizer can understand. The information attached to these signals are converted as well which will adjust the voltage as needed.

Control Voltage Basic Usage:
In analog synthesizers, each synthesizer component (low frequency oscillator (LFO), voltage controlled filter (`, etc.) can be connected to another component by means of a patch cable that transmits voltage. Changes in that voltage cause changes to one or more parameters of the component. This frequently involved a keyboard transmitting two types of data (CV and gate), or control modules such as LFOs and envelope generators transmitting CV data.
The table below illustrates the CV component and their definitions:

Control Voltage	Indicates which note/event to play. This is done by a different voltage for each key pressed, connected to an oscillator (or more than one, in our team's case), and thus producing the different pitches.
Trigger	This indicates when a note should start, a pulse that is used to trigger an event such as an ADSR envelope.
Gate	This is related to a trigger but sustains the signal throughout the entire event.

Table 5.0 - CV Components

The concept of CV is standardized on analog synthesizers and has two main implementations:

Volts per octave (The V/OCT Method):
This method was mainstreamed by Bob Moog, the inventor of one of the most popular analog synthesizers in the world. One volt represents one octave, as such the voltage increases by 1V per octave. Each 1V octave is divided linearly into 12 semitones.

Hertz per volt (The Hz/V Method):
This method is used and implemented in most Yamaha synthesizers. This method represents an octave of pitch by doubling the voltage.

The figure below compares notes and corresponding voltage levels:

In this example, we use 1V per octave, and 55 Hz/V.

Method	A1	A2	A3	B3	C4	D4	E4	A4	A5
V/OCT	1.000	2.000	3.000	3.167	3.250	3.417	3.583	4.000	5.000
Hz/V	1.000	2.000	4.000	4.490	4.757	5.339	5.993	8.000	16.000

Table 5.1 - Conversion Method Note-to-Voltage Ratio

The formula that links these voltages is:

$$V_{oct} = ln_2(V_{hz}) + 1 \qquad \text{OR} \qquad V_{Hz} = 2^{V_{oct}-1}$$

MIDI-CV Converter:
This is a device that accepts MIDI messages and produces control voltages based on the received messages that a modular synthesizer understands. This type of device is intended to allow MIDI control of analog synthesizers specially, which do not have dedicated MIDI interfaces. The most basic type of MIDI-CV converter will accept MIDI note on and note off messages, and will then produce two outputs: a control voltage which is directly proportional to the MIDI note contained in the “on message”, as well as a gate signal which is toggled by either of these on or off messages.

In consideration of which type of converter we would prefer, we have considered a lot of factors. MIDI to CV converters cover a wide range of features, and some work better or worse for different applications. It is very important that the converter we choose is best suited for our needs. Some of the simplest options in one convert MIDI note information to a pair of Pitch CV and gate outputs, this is a great feature in an application that uses a MIDI keyboard or sequencer. It provides excellent and straightforward pitch and timing to an analog synth voice (i.e our project). Other converters are much better suited for sending CC messages as CV signals, this is a factor we don’t really need to prioritize. Others work great as a drum machine sequencer, and so on.

Tuning and Resolution:
The tuning of the destination oscillator is extremely important when using MIDI information. You usually want to tune the oscillator to the lowest note in the sequence.

Standalone MIDI-CV Options:
Over the last few years, a lot of standalone MIDI_CV converters have come to market with the resurgence of vintage mono synthesizers and other old analog technology. These include a CV/gate integration! One of our top choices for this project is the Doepfer MCV4, which provides standalone conversion of MIDI signals into CV and gate. There are more dedicated devices if your needs go beyond just converting a single monophonic instrument line, but our device currently does not need that support. Choosing to buy a standalone device over creating one ourselves was one thing our team had to consider in the part selection summary before we could order the PCB. The current route we are

thinking of taking is creating it ourselves, but if we choose to go with a pre-built device we did agree with our top choice currently.

Doepfer MCV4:
This is our current chosen MIDI-CV Converter Interface. It is a four mode MIDI to CV gate interface. This is said to be the most ideal device for controlling monophonic lines and most vintage synthesizers. It includes two dual CV outputs and a single gate input. It also features MIDI control of CV1-CV4. This means and includes control of note on/off with pitch-bend, 1V/octave characteristic, monophonic after-touch, velocity and/or volume (controller #7), and velocity.

Figure 5.10: Doepfer Midi-to-CV/Gate Converter

5.7. User GUI Application

The idea for this is for the user to be able to interact with the device through a guided user interface (GUI). This application is built on the Java programming language and will include public libraries and various open-source application programming interfaces (API) such as the JavaX Sound API's MIDI package.

5.7.1. MIDI/API connections

[28] The application is able to read Musical Instrument Digital Interface (MIDI) data through what is known as MIDI wire protocol as a part of the JavaX MIDI package. This effectively splits the MIDI signal–which is transported by MIDI, DIN, or Universal Serial Bus (USB) cables–into distinguishable byte data.

MIDI signals are processed by the JavaX MIDI package as an abstract class (an encompassing template for Java objects). This abstract class is known as a MidiMessage which is composed of three subclasses, ShortMessages, SysexMessages, and MetaMessages. However, for the sake of this application, only ShortMessages will be used as SysexMessages contain metadata for hardware and MetaMessages contain information generally only useful for sequencers, not synthesizers. The API consists of a hierarchy that separates MIDI data into three main categories: Sequence, Track, and MidiEvent, however, these categories all reside within the same class and thus have the same inheritables and characteristics. A Sequence is a collection of Tracks which is a collection of MidiEvents.

[40] These messages are sent within MidiEvents objects (the creation based off of a class) and the MidiMessages are led by a status byte which distinguishes the type of message sent. Following the status byte are the data bytes which notify the receiver of the pitch, volume, and other adjustment levels of the MidiMessage. Each message is also sent with information detailing timing information that tells the track when to start and stop the notes played. These are known as the note-on and note-off events that are sent to the MIDI channels. MidiEvents take the arguments of a single MidiMessage obtained with the getMessage() method and obtains the current tick time of the MIDI message as a 64-bit long variable through the getTick() method.

The interfacing of MIDI devices to this API lies within the MidiDevice interface (descriptors for Java classes entailing its uses and contents). This interface can act as both MIDI input ports and MIDI output ports which can allow for exclusively digital MIDI communication, however for this device, the MidiInterface will be used as a MIDI input port using the subinterfaces Synthesizer and Sequencer to communicate with the MIDI input device.

The difference between a Synthesizer and a Sequencer is that a Synthesizer is capable of producing generated sound while a Sequencer is only capable of recording and playing back MIDI messages in a sequence. In this case, an electronic keyboard piano will act as the MIDI input (may be subject to change). This input can be read by the device using the implementable Receiver interface which is the interface responsible for generating sound and sending the sound to the synthesizer.

If the MIDI input was to be generated digitally, a respective Transmitter interface would be used. A single MIDI object consists of 16 MIDI channels, thus the synthesizer has control of 16 MidiChannels as there is only a single MIDI input device. All the aforementioned data regarding the note-on or note-off, pitch, frequency, and other parameters will be communicated to each of the MIDI channels as required. If there were to be another MIDI input device, whether analog or digital, a separate set of 16 MidiChannel objects and thus a second Synthesizer object would be required.

The Synthesizer interface is capable of storing various instruments that can be loaded and unloaded at any time to switch up which instrument is creating the sound. These virtual instruments are stored within a Soundbank interface. To load a new instrument, the interface must first communicate with the MIDI message being delivered to let the program be aware that there is a change in the current loaded instrument. If the attempted Soundbank interface to be loaded cannot be successfully loaded, the Synthesizer will throw an IllegalArgumentException.

5.7.2. User Interface (UI)

The user has the ability to visualize the sound processed by the synthesizer through a user interface (UI). The UI will include information regarding the sound's pitch, frequency, and other adjustable parameters. Alongside this information will be a waveform based off of the sound generated by the synthesizer. As the user adjusts the parameters using the analog dials and buttons, the UI will update both the display of the parameters as well as the waveform in real time on the screen.
The UI is one of the most important parts of the product design as it will be the

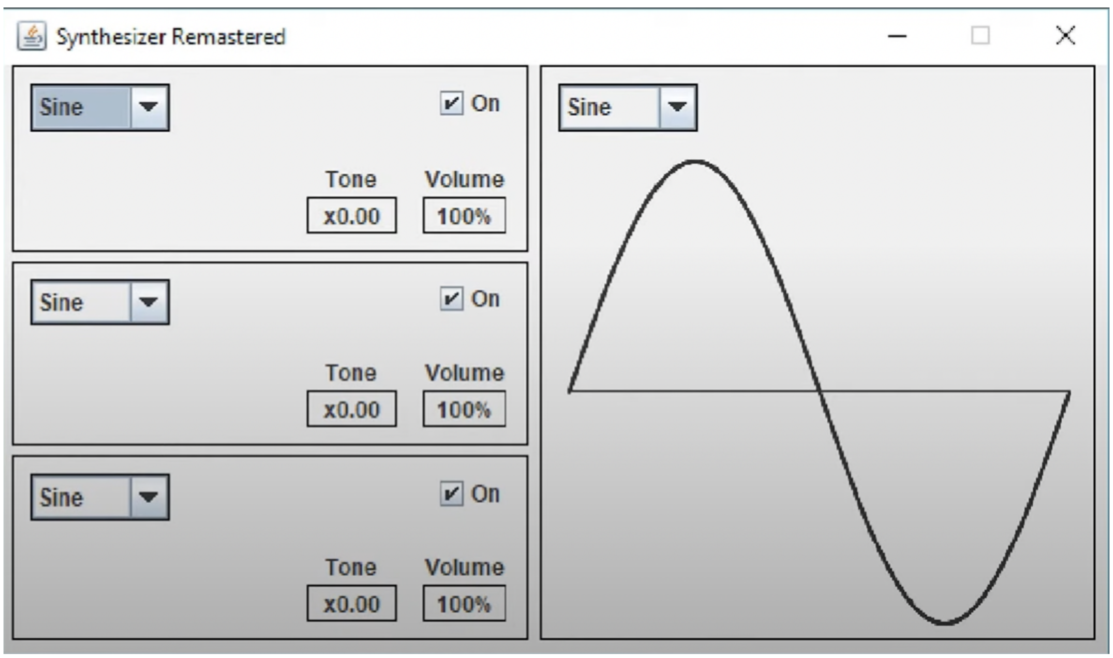


Figure 5.11 Synthesizer UI

main thing that users will be viewing. The UI displays the most important parts of the project all in one tiny screen, thus it is important that the UI is created in such a way that it is both user friendly and easily readable whilst also giving a further understanding as to what the user is interacting with.

For our UI, we want to focus on both the user experience as well as accurate readings of our waveform. Similar to the **guided user interface (GUI)** shown above, we want to shape our GUI to allow users to understand the waveform that they are creating using the synthesizer as well as what steps led them to creating that waveform. In this specific example, a sinusoidal waveform is created by an application, however, for our project, the waveform shown will be the one generated by the MIDI controller that is converted into the CV form.

However, we will still retain the core application functionality of displaying the waveform with a drop-down box option to select various waveform shapes such as sinusoidal, sawtooth, or square. Our color options are also limited as we will be working off of either an I2C or SPI display that can only display a select amount of predesignated colors. This is another design constraint for the UI but is not critical to the core functionality thus is not a major issue at this given time.

5.8. Human Interface Devices (HID)

HIDs are the components of a device that the user interacts with directly. These are physical components that require human input in order to send data, generate signals, or create output for humans to observe. For example, a person’s personal computer would generally include a mouse and keyboard as HIDs where the mouse will adjust the on-screen cursor given the user’s x and y-axis movements read through an optical laser or a sensory touchpad. Being able to see the change digitally is a core concept of HIDs.

5.8.1. Keyboards

Figure 5.11 Example of Keyboard: AKAI MPK Mini

Keyboards, also known as an electric or digital piano, is a device composed of a set of keys that are capable of producing sounds or signals representing sounds. The Musical Instrument Digital Interface (MIDI) signals that are needed to be sent to the synthesizer can be generated by both MIDI controllers or a digital piano. The difference between the two input methods is that MIDI controllers do not have a set internal sound nor external speakers unlike most digital pianos. MIDI controllers also do not have the main purpose of reproducing acoustic pianos, instead, they primarily transmit MIDI signals. The average digital piano is composed of 88 keys that contain multiple octaves of musical pitches, MIDI controllers can be as large as a digital piano but may also be smaller as their pitch can be more easily adjusted in case an octave change is needed.

For this project, albeit either device can be used to generate the MIDI signals, the MIDI-exclusive controller would be more cost effective and would be a better fit to the standards and goals originally set. Since we do not plan to produce sounds using the keyboard, there is no need for a digital piano over a MIDI controller. This MIDI controller can then be directly interfaced with the synthesizer through the use of a 5-pin DIN cable and will be able to transmit the signals as they are played.

5.8.2. Buttons and Switches

Buttons and switches can refer to any analog devices that are capable of delivering binary responses. Both are capable of completing circuits to allow the flow of electricity to pass–every switch and button has at least two terminals that it connects together. These components can only be in one state at a time, those being on or off. Albeit simple in design, there are variations that can be made to the actuation method of buttons and switches to differ them. Actuation refers to how the button or switch is interacted with in order to change its state whether it be how hard a button must be pressed or in what way a switch must be rotated, flipped, pulled, etc. Every button or switch can be maintained or momentary. Maintained buttons or switches are those where after being actuated they retain their state similar to that of a mechanical pen or a light fixture switch. Momentary buttons or switches will return to their base state, whether that be on or off, after actuation similar to a keyboard.

5.8.3. Rotary Encoders

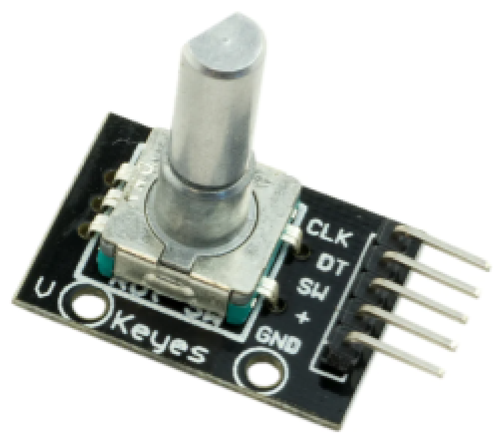


Figure 5.12 Rotary Encoders

[29] Rotary encoders are devices that are composed of a central rotating shaft whose angular position and rotation can generate electrical signals. Rotary encoders are similar to rotary potentiometers in that they are able to spin on an axis to generate an electrical signal, however, rotary encoders do not have a set limit on how far in the clockwise or counterclockwise direction they are able to spin. The average rotary encoder consists of a 5-pin setup that can be wired to any breadboard or microcontroller. These pins represent the clock (CLK), a 90 degree phase shifted clock (DT), an active low switch (SW), the supply voltage (VCC), and ground (GND).

The rotary encoder works by generating square waves based on the rotational position of the shaft on top of a common plate. This common plate is separated evenly with electrical contacts that the CLK and DT pins can trigger off of. The CLK and DT pins determine the direction the rotary encoder is being spun based on the order that they come into contact with the common plate contacts. If a pin comes into contact with the contact, the pin sends a 1, or a signal high.

Likewise if the pin leaves the contact, the signal returns to low, or 0. The output signals of CLK and DT are phase shifted by 90 degrees, this means that if CLK makes contact and sends a signal first, DT will send a following signal notifying the microcontroller that the rotary encoder is being spun in the clockwise direction. Vice versa, if the DT pin makes contact and sends its signal first with the CLK signal following 90 degrees behind, this will notify that the rotary encoder is being spun in the counterclockwise direction.

There is one rotary encoder per adjustable parameter that requires the use of one allowing each parameter to be individually adjusted. Parameters that lie on a range should be the only ones adjusted by rotary encoders, however, binary parameters (those with only two options) can still be adjusted by the encoders if requested. Knowing the direction that the rotary encoder is being spun allows for easy software integration regarding adjustments done to the MIDI parameters.

A software method (coding function) could be implemented to determine the direction of the rotation given the signal order and thus could be used to increase or decrease parameters as needed. For example, if the user would want to increase the pitch of the MIDI sound, they would simply have to rotate the rotary encoder that represents pitch in the positive direction. The software would determine which encoder is being adjusted, determine the direction of the adjustment, and would repeat this process as long as the rotary encoder retains an alternating signal.

5.8.4. Rotary Potentiometers

Similar to rotary encoders, rotary potentiometers are devices that act as a variable resistor by sliding a knob that adjusts a wiper placed underneath between contacts placed in a semi-circular pattern. These contacts are connected with a wire. As the knob is adjusted, the output voltage is also adjusted based on the location of the point of contact. The wiper under the knob will always be connected to the wire at all times. Rotary potentiometers work on a 3-pin setup. These pins are the counter-clockwise end of the resistor (CCW), the knob attached to the wiper (W), and the clockwise end of the resistor (CW).

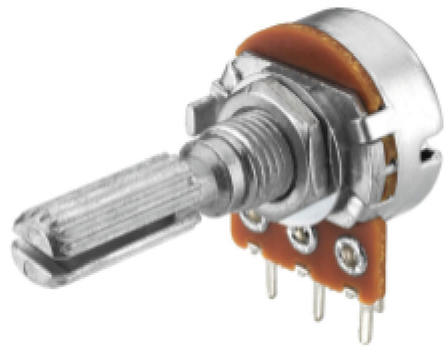


Figure 5.13: Potentiometer

Unlike rotary encoders, rotary potentiometers are more precise as they do not rely on the encoder leaving or approaching contacts to update their position. This means they are able to be more accurately be adjusted and thus are able to give more accurate output. Potentiometers also have limited rotational ability and can only provide output on the range provided.

In the case of our project, we have decided to go with potentiometers. They offer superior real-time control for users which is crucial to allow the musician to exercise control over their synthesizer. Potentiometers are more suited for analog applications like our project.

6. Project Prototype, Construction, and Coding

In this section, we will be discussing the initial prototyping plan for our analog synthesizer. We will go over plans of PCB design and practices, plan of assembly, final coding route, and enclosure details. This section serves as a preliminary strategy for our Senior Design II build phase.

6.1. Integrated Schematics

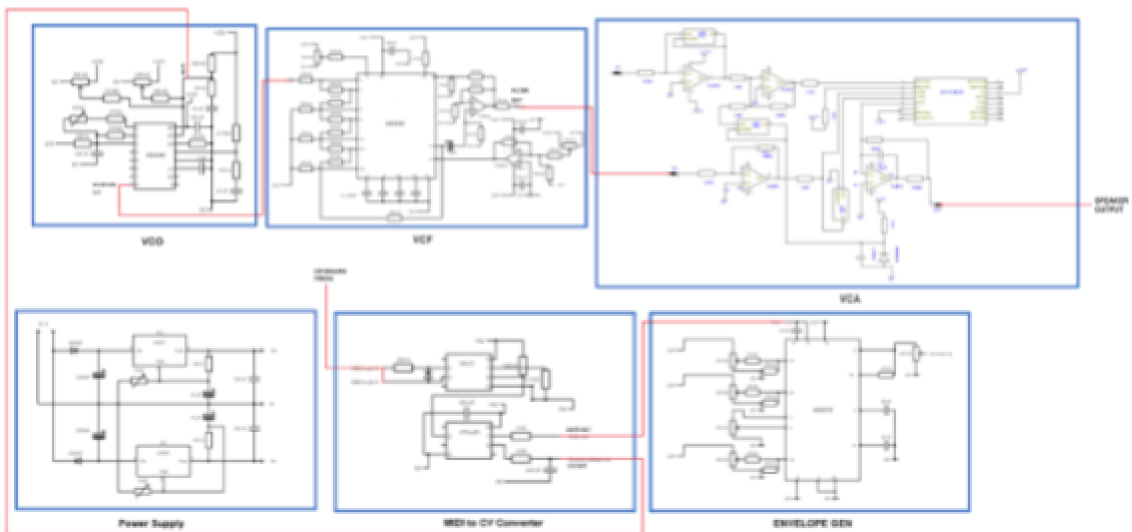


Figure 6.0: Overall Synthesizer Schematic

Above are our overall integrated schematics featuring every module of the synthesizer. This includes the power supply and MIDI to CV converter that will go on one PCB and the synthesizer sound processing modules that will go onto the next. These modules include the VCO, VCF, VCA, and envelope generator.

6.2. PCB Vendor and Assembly Plan

Our printed circuit board (PCB) supports everything mentioned in our block diagram from Chapter 2. This means are able to include the power supply, MIDI to CV converter, and all synthesizer modules including: VCO, VCF, VCA, and envelope generator on PCB. We plan to split the power supply and MIDI to CV converter on one PCB and the above mentioned synthesizers on a second PCB. Considering the power supply and MIDI to CV converter both have large footprint components like electrolytic capacitors and 5-pin DIN inputs. Each synthesizer input and output will be connected accordingly and soldered onto the audio processing PCB.

The electronic design automation (EDA) software we use is EasyEDA. EasyEDA is an online PCB design tool for students, engineers, educators, and enthusiasts to perform schematic capture for free anywhere, anytime, on any device. They are able to do so due to their online in-browser nature unlike other EDAs we considered like Eagle, Kicad, and Altium.

Considering we are building an analog synthesizer, all of our electrical design is in the analog domain. This means we have a large number of components on PCB. To divide up the workload, the two electrical engineers on our team split up the synthesizer modules into groups of VCO and envelope generator, and VCF and VCA. We plan on collaborating on the power supply and integrating our modules together. With a project that involves so much analog circuitry, we had to divide and conquer and EasyEDA makes that possible.

EasyEDA makes our schematic capture and PCB layout a breeze with its online feature so that we can build and design each of our modules separately and check in on each other's progress. If one of us needs help or encounters a design issue, we can easily open up the project folder and peer-review each other's work. We were able to work together on the same schematic and PCB in real-time such that we can share ideas on our layout.

EasyEDA has over a million free libraries and we were able to successfully import the AS integrated circuit lines into our schematics. Since we are students that often go back and forth from our home work computers to the Engineering Senior Design I laboratory, we needed a software suite that we could use on any device. Since each of our individual schematics have been laid out, we are left with PCB layout. We plan on doing this over the winter break in between Senior Design I and Senior Design II so that we are ready to order PCBs when Spring semester begins and can avoid going through multiple revisions.

Our PCB manufacturer of choice is JLCPCB. We chose this manufacturer as they work in conjunction seamlessly with EasyEDA. Upon researching many other PCB manufacturers in the USA like Advanced Circuitry International, Rush PCB Incorporated, Sunstone Circuits, PCBgogo, etc. we concluded that JLCPCB made the most logical sense for our project. JLCPCB is already integrated with EasyEDA. Since the two softwares are under the same corporate name, we can simply import the schematics and gerber files right into JLCPCB and have everything under the same hood. This PCB layout software stood out to us after extensive research of different manufacturers and comparing their reviews, location, cost per layer, and shipping time.

Since we are in the audio domain, we were not too concerned with trace width, controlled impedance, routing tolerance, etc. since we are working with relatively low frequencies (i.e. those of the human hearing range from 20Hz to 20kHz). We had plannned on saving on board dimensions and materials by choosing surface mount components when building and designing our PCB. For components like the numerous 1% tolerance through hole resistors and ceramic capacitors, we intend on switching to surface mount to save on area and materials cost. This made our synthesizer module PCB small and easily portable into a small enclosure as we intended to make our synthesizer compact and portable for the average musician and enthusiast.

After observing each of our schematics, we looked at the number of components involved and proposed that at least two layers would be required. Looking at each synthesizer module individually, there are not too many components involved in each – but after integrating them all together there were definitely a large number of components involved. We anticipate we would have to put signals and power on the top plane and dedicate the bottom layer to ground. In the case that we need more planes, we knew that we would need resort to a four layer PCB and would have more real estate to work with since we would be able to split the signals (routed internally) between the top and bottom layer and have power and ground on the middle planes.

Going with our ideal plan of a two layer PCB for cost and design constraints, we wanted to be careful to reduce EMI susceptibility, crosstalk, ground loops and preserve our audio signal integrity. Dedicating our bottom layer to a ground plane is the best method from an electromagnetic compatibility perspective. This would provide our input and output audio signals protection from EMI as mentioned before and reduce our noise and signal crosstalk.

We took into account the dimensions of each through hole component, port, electrolytic capacitor, and the numerous linear potentiometers throughout as these take up a large amount of real estate on our board dimensions. We made sure to plan in advance in choosing through hole ICs that could easily be soldered on ourselves. For surface mount components like resistors and capacitors, JLCPCB offers assembly making soldering 0602 package resistors for example easy and foolproof. We look forward to beginning PCB layout and routing our schematics for a fully analog synthesizer that withstands the modern test of time.

6.3. Digital to Analog Conversion Plan

ATtiny85 MIDI to CV

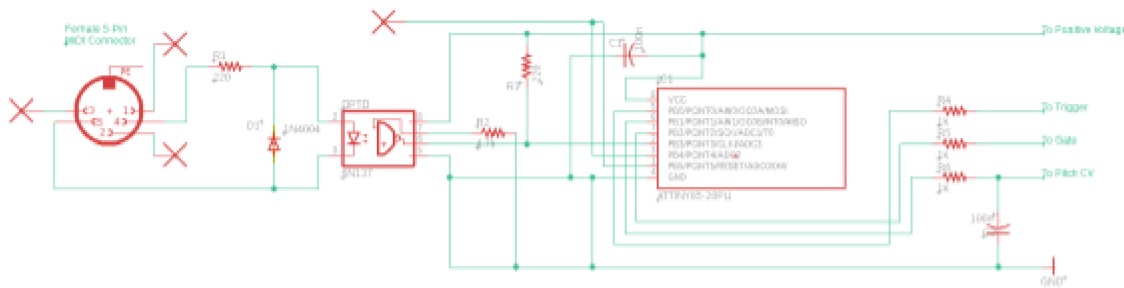


Figure 6.1: MIDI-to-CV Schematic

Outlined in this section is how the creation and testing of our 5-PIN DIN MIDI to CV function will occur. [30] A basic MIDI in circuit will be paired with the ATtiny85 CV out section based on an open source design found online.

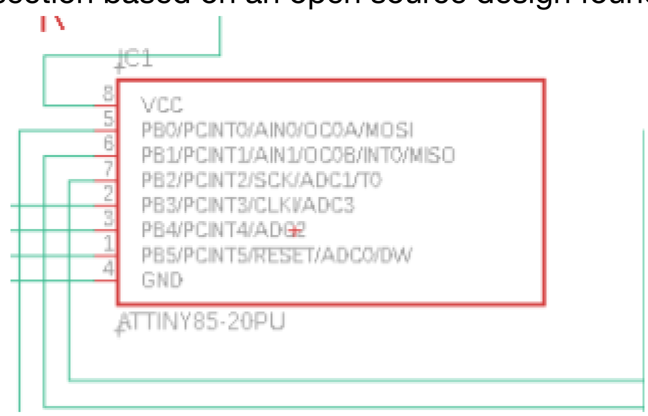


Figure 6.2: Pins of ATtiny85 Up Close

The method this code uses is an algorithm developed by Jan Ostman (whose online repository has now been retired) which denoted that if the top compare value for a **pulse width moderation (PWM)** operation is 239 then there are 240 graduations for PWM. To cover a MIDI range of C2 (note 36) to C7 (note 96) is 60, so the PWM compare value required for a linear CV voltage output is:

$$(note - 36) * 4$$

Regarding software, the pins on the board, including serial pins and GPIOs, will be the following:

Pin Usage	Notes:
SoftwareSerial receive on D3(PB3)	This is physical pin 2
Gate output on D2(PB2)	This is physical pin 7
CV output using the PWM signal that is tired to OC1A triggered off timer 1	This is D1(PB1) on physical pin 6
Trigger output on D0(PB0)	This is physical pin 5

Table 6.0: [31] ATtiny85 General Use Pins

For timer control registers, referring to the data sheet for ATtiny85 we have:

Register	Notes:
Set PWN1A	PWM based on OCR1A
Set COM1A1	Clear OC1A output line
Set CS10	Prescaler = PCK/CK - runs at clock speed
Clear PWM1B is not enabled	GTCCR = 0

Table 6.1: [32] ATtiny85 Timer Control Registers

When doing research in the different modes we could use, we opted to use the Fast PWN with the compare value in OCR1A and the maximum PWM cycle value, 239, in ICR1. The timer register setting were then chosen as follows:

Timer 1 Control Register A || TCCR1A	Timer 1 Control register B || TCCR1B
7 COM1A1 = 1 COM1A1(1); COM1A0(0) = Clear OC1A on compare match; set at TOP	7 ICNC1 = 0
6 COM1A0 = 0	6 ICES1 = 0
5 COM1B1 = 0	5 Resv = 0
4 COM1B0 = 0	4 WGM13 = 1 WGM13(1); WGM12(1) = PWM from OCR1A based on TOP=ICR1

3 Resv = 0	3 WGM12 = 1
2 Resv = 0	2 CS12 = 0 CS12(0); CS11(0); CS10(1) = Prescaler = PCK/CK i.e. run at Clock speed
1 WGM11 = 1 WGM11(1); WGM10(0) = PWM from OCR1A based on TOP=ICR1	1 CS11 = 0
0 WGM10 = 0	0 CS10 = 1

Table 6.2: [33] ATtiny85 Pulse Width Modulation (PWM) Values

Timer 1 control register C will be left at all zeros. Now in terms of pin definitions, according to the data sheet this is what we will use:

- Hardware serial receive on D0 (PD0) which is physical pin 2.
- Gate output on D11 (PB2) which is physical pin 14.
- CV output using the PWM signal tied to OC1A triggered off timer 1, which is D12 (PB3) on physical pin 15.

Now in regards to the loop, the code that we will program into the MCU will follow logic that says this (in pseudocode of course) as an algorithm:

```
IF (at least three bytes of serial data received)THEN read MIDI command value
   IF (MIDI note on received) THEN
     read MIDI note value
     read MIDI velocity value
     set CV out value based on MIDI note value
     set Gate signal HIGH
   ELSE IF (MIDI note off received) THEN
     read MIDI note value
     read MIDI velocity value
     set CV out value based on MIDI note value
     set Gate signal LOW
   ELSE
     ignore and go round again waiting for serial data
   ENDIF
 ENDIF
```

Figure 6.3: [34] ATtiny85 Pseudocode

The reason we are following this type of algorithm is because we grabbed information from the MIDI spec data sheet that describes “running status” bytes and how they are sent and received. This algorithm that we implemented therefore includes proper handling of the running status of MIDI. Using “at least three bytes of serial data" received means that if things get out of sync, eventually bytes are skipped until there are three bytes that equate to a note on/off message. We finished this testing environment by using an 8MHz internal clock for the ATtiny85. We were able to finally test all of this using a MIDI keyboard, controller, or any type of testing hardware. The full code is not allowed to be posted in our documentation but will be posted in our github.

Connections and receiving MIDI data:

[35] With our Arduino Uno we first test if we are actually receiving the MIDI data. Our testing plan for this part of the project was to set up our MIDI controller with a DAW system running on a computer, and then build the breadboard to make a connection with our breadboard 5-PIN Socket to the Arduino Uno connection. Following this successful connection, we wrote code into the Arduino IDE and import the correct available MIDI library to check if we are receiving MIDI signals. To do this, we made sure the MIDI controller has a standard MIDI out port included on the system to send the controller the signals. We also needed to make sure we wire all the software and hardware correctly to successfully receive signals and begin our debugging and testing process. We also ran into a few issues with what types of cables we needed to use along with the ports. To make sure we do this correctly to duplicate results and continue with our testing plan, we created a flow chart with correct assembly for the MIDI-to-CV conversion.

The following diagram was created to show the flowchart of our specific MIDI-to-CV conversion plan. It also shows all the correct in and out ports needed from each piece of hardware to correctly make these connections with our breadboard setup.

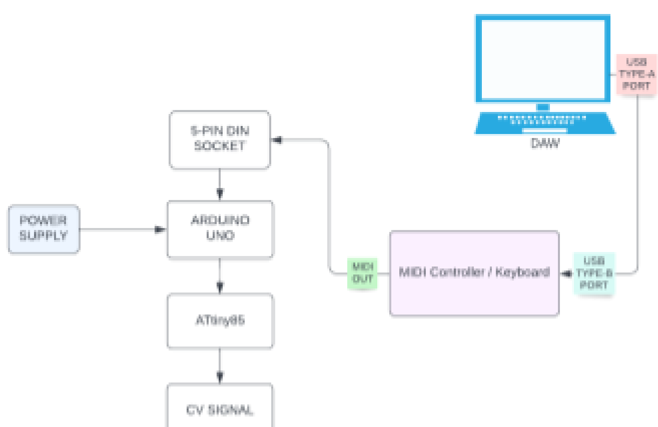


Figure 6.4 Receiving MIDI Data

Once we have established a successful connection and are receiving MIDI data to our Arduino, we program the Arduino to communicate with the ATtiny85 and then convert the signal to CV for the analog synthesizer to successfully work.

Enclosure and Housing Assembly

An analog synthesizer enclosure is a physical container that houses the various components of an analog synthesizer. It typically includes knobs, switches, and other controls to adjust sound parameters such as pitch, volume, attack time and so on. It also usually contains jacks for connecting external devices like pedals or keyboards. The enclosure is designed to protect the internal components from dust and moisture while providing access to all necessary controls and connections.

Building an analog synthesizer enclosure requires careful planning of the layout of all its parts in order to ensure that everything fits properly within it and is accessible when needed. The first step was deciding what type of material would be used for the construction; this could range from wood panels to metal sheets depending on personal preference or budget constraints. Once a suitable material has been chosen then measurements can be taken according to how much space each component needs in order to fit inside without any obstruction or conflict with other items.

The next step would involve cutting out the pieces using either hand tools like saws or power tools like jigsaw machines based on how precise one wishes their cuts to be; this should include holes for ventilation purposes if necessary as well as slots for mounting screws which will hold together different sections during assembly later on.

Afterward, pieces can then be sanded down smooth before being painted with a protective layer of varnish in order to prevent corrosion over time due to regular use near electrical circuits/components inside it - this will also give it more aesthetic appeal if desired too!

Finally once all these steps have been completed one must assemble their newly created synth case by attaching hinges on doors (if applicable) before finally installing any additional hardware such as potentiometers/knobs etc., making sure they are securely fastened into place via nuts & bolts so they don’t come loose during operation later down the line.

Our team has identified cost and material as key considerations when selecting an enclosure for our PCB. To optimize both, we have decided to pursue laser cutting of wood in the TI lab as our chosen solution. This approach will provide a cost-effective option while still allowing enough space for our PCB design.

We ended up using an acrylic sheet for the front panel and a 3D printed part for the knob assembly. We planned to also include rubber feet on all four corners of the enclosure in order to keep it steady while playing with it.

The housing assembly of our analog synthesizer were carefully considered due to factors such as cost, material, and size of PCBs used. To create a cost-effective solution, we have opted to laser cut wood using a TI lab machine.

Apart from wood, there are a variety of materials that could be used for the enclosure of an analog synthesizer.

One option is plastic; this material is relatively inexpensive and can be easily sourced in large quantities. Plastic enclosures also provide excellent protection against dust, moisture and other environmental factors. However, plastic does not offer much structural support and may require additional reinforcement or bracing to ensure its durability over time.

Aluminum is another viable option for an analog synthesizer enclosure; it offers good heat dissipation properties as well as improved strength compared to plastic. Aluminum enclosures are usually more expensive than their plastic counterparts, but they will last longer due to their higher quality construction. Additionally, aluminum enclosures often have better shielding capabilities than those made from other materials due to their greater conductivity.

Finally, stainless steel can also make a suitable choice for an analog synthesizer enclosure because it provides superior strength and corrosion resistance despite being heavier than aluminum or plastic options. Stainless steel is generally more expensive than both aluminum and plastic though, making it less attractive in terms of cost efficiency depending on the project’s budget constraints.

7. Project Prototype Testing Plan

The first project prototype was conceived during our time in Senior Design 1 into our transition into Senior Design 2. We prototyped each block on the block diagram from Section 3.4 on breadboard first and ensured success before designing our PCB. Hardware testing was done on the audio processing modules, software testing was done on MIDI to CV conversion and application usability and overall testing was the integration of everything just mentioned. The most difficult part of testing was when attempting to integrate the various devices and components together.

7.1. Hardware Testing

Hardware testing consisted of building the power supply, VCO, VCF, envelope generator, and VCA. We breadboarded these modules separately on breadboard and tested each individually in order before connecting them together. Our hardware test environment took place in the Senior Design Lab at UCF's Engineering Building I. This is the location where we spent most of our time building the circuitry, testing, and integrating. We met here regularly as we do not have the following equipment at home.

Equipment Needed:

- Oscilloscope: Rohde & Schwarz RTM 3004
- Function Generator, Tektronix AFG 3022B
- Digital Multimeter, Tektronix DMM 4050
- Triple Output Power Supply, Agilent E3630A
- Solderless breadboards

Synthesizer Modules

VCO

The VCO was the starting point for testing our sound processing modules as it is the heart of any synthesizer. In our case, our VCO is to be capable of producing ramp, sawtooth, pulse, and triangle waveforms. During this testing phase, we wanted to ensure the functionality of our AS3340 VCO integrated circuit. At the component level, we could not simply power the chip and probe pins alone – we had to flush out the surrounding external circuitry per the data sheet to see anything meaningful. In doing so, we were able to observe both if our IC was working and if our schematic design was successful.

There was no way that we could have tested the IC alone without any surrounding circuitry. Test engineers at the manufacturer (Alfa in this case) of the AS3340 would be the only ones able to test the IC itself. They have the advantage and knowledge of the exact chip implementation internally and are the ones responsible for delivering a working product. Though we would ideally be faithful in that we were sent a working chip, we had no choice but to be on the side of caution.

Considering that the AS3340 has complex internal circuitry taking care of the building blocks of the VCO: input adjustment, linear to exponential conversion, oscillation, and buffering, we were only be able to ensure this was all working correctly by building the foundation of our VCO and testing if the chip worked by observing the output. In other words, we were able to knock out two birds with one stone.

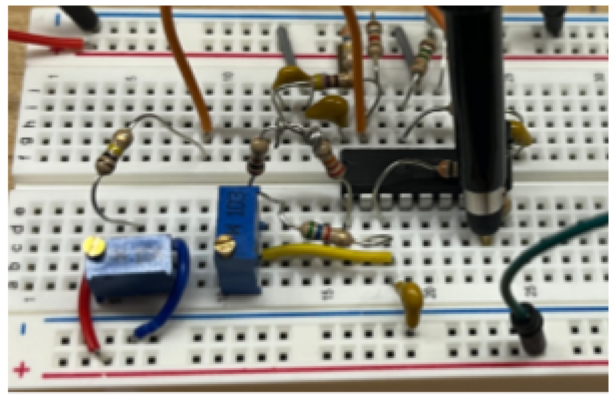


Figure 7.0: VCO Testing

Given that we wanted to test VCO off one wave type alone, we started building the circuit to achieve a ramp output. We chose to build the ramp output as this was the foundation of our VCO schematic mentioned prior in Chapter 5 on the project design. To the right is the breadboarded prototype for the VCO.

We encountered some issues with our linear potentiometer knob and had to resort to using a second trim potentiometer (the leftmost one) to successfully change the frequency (i.e. pitch) of the oscillator. The linear potentiometer we were using was causing issues with our dual +/-12V voltage and preventing us from supplying the AS3340 its power rails. In order to troubleshoot, we checked every connection, probed the pins, checked our power supply configuration, and made sure that every component was seated correctly. It was upon touching the leads to our linear potentiometer that we felt one of the jumper wires connecting to its three

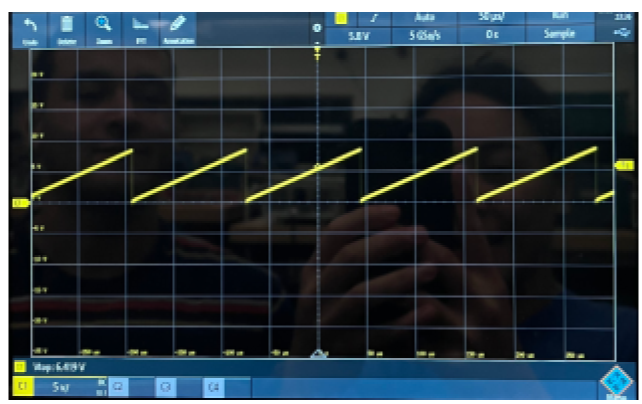


Figure 7.1 Sawtooth Waveform from Testing IC

terminals feeling hot to the touch. Upon removing it, our power supply voltage shot straight up to +12V for the positive rail. After this discovery, we switched to a trimmer potentiometer and resolved our issue. Other than that, we were able to successfully test the AS3340 and achieve the ramp wave output that we desired.

Seen on the right is our successful output. Even using the 2% resistors and having to interpolate a capacitor value using components provided in the lab we were still able to achieve a clean ramp wave output that changed with our varying voltage input. This measurement was taken on pin 8 of the AS3340.

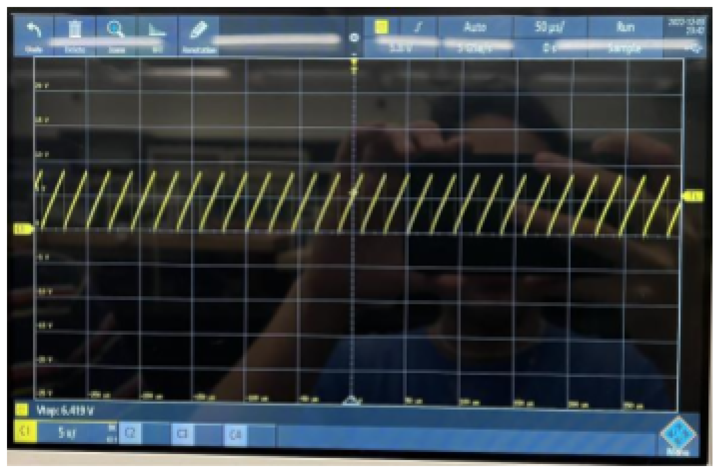


Figure 7.2: Sawtooth Waveform Adjusting Pot

As we turned the trimmer potentiometer, we were able to adjust the frequency (i.e. pitch) of the VCO. As we increased the resistance via the trimmer potentiometer, we got an increase in voltage and a higher pitch heard by the user. You can see the increase in frequency clearly working correctly which solidified we will be able to use this in our final design. Eventually, used a speaker to plug into our design and hear the oscillator itself to judge its sound quality but right now we are solely focused on the waveforms.

VCF

Next in the audio signal processing chain was the VCF. This module uses the output of the VCO to adjust its cutoff and resonance characteristics. We were able to breadboard the entire VCF successfully and verify correct +/-12V voltages but unfortunately misoriented our TL082 operational amplifier's supply polarity. This caused the TL082 to burn out thus causing a halt to our VCF testing. We ordered more TL082s (with extras this time) to continue the testing. We plan on connecting the output of the VCO ramp wave from pin 8 to the VCF input on pin 1.

Once we feed the output of the VCO to the filter were able to probe the VCF output from pin 10 – Output Stage 4. We should were able to see the output of the ramp wave completely and be able to modulate it via the cutoff and resonance potentiometers on the breadboard.

VCA

After verifying the VCF, were able to breadboard and test the VCA. The VCA gets its audio signal output fed from the VCF directly to the input pin. The output of the VCA will be the output of the synthesizer itself. Considering it should amplify the voltage signal to an appropriate output level, this module was fairly straightforward to determine if it is meeting its criteria or not. We needed to connect the oscilloscope in such a way that the input signal is output on setting the relevant potentiometer. If we turn the potentiometer to reduce the resistance, we should see an increase in amplitude and increase in gain.

The VCA is a special module as it will be integrated to work in conjunction with the following module – the envelope generator. We want the envelope generator to work on both the filter and the amplifier such that the user can manipulate both envelopes. Applying the envelope generator to the VCA makes the sound more or less bright and applying to the VCA controls the volume over time. Essentially, it will serve as a complex volume control. Not only getting louder or softer but shaping the contour of how long the sound is loud for, or how fast it decays, etc.

Envelope Generator

After verifying the VCA and VCF are able to successfully modulate the oscillator's waveform, we can move onto manipulating the sound envelope itself. In order to test the envelope generator, we needed to breadboard the circuit discussed in section 5 and probe the IC with the oscilloscope to observe each characteristic that defines the loudness of sound: attack, sustain, decay and release (ADSR) on both the filter and amplifier envelope. If needed, recall these definitions in Chapter 2.

Upon viewing the envelope waveforms on the oscilloscope, we are able to see and interact with the contour shaping. A large part in play will be the trigger of the design. We will determine the success of the envelope generator by its responsiveness to key presses and exponential sloping. To do so, we bring in our MIDI compliant keyboard that one of our team members will be providing. This way we hook up the keyboard and hit keys on the synthesizer we would by the end of our project while observing the envelopes on the scope.

Power Supply

Testing the power supply involved our breadboarded power supply module and 15V AC to AC output wall wart. We will integrate the two together and identify the correct voltages and currents on the oscilloscope. We ensured that these values did not exceed the maximum ratings of our AS3340, AS3320, and

AS3310 of +24V combined. We were able to observe not only from a quantitative perspective but also a qualitative one such that we minimize any ripple voltage in the power supply for a clean supply for our synthesizer modules.

Keyboard

Testing the keyboard was as simple as validating that human input would generate MIDI signals. To test this, a connection was made from the keyboard using a MIDI-to-USB cable to a computer. We downloaded a program developed by BandLab called Cakewalk to verify the keyboard's inputs. Cakewalk is a music production software similar to Apple's Garageband that allows for the reading of MIDI data from applicable devices connected to the hardware running the program. In this case, the keyboard for testing was selected as the input device and the speakers of the computer were used as the output. When pressing keys, audio feedback was instantly granted and thus we knew that we were successfully getting MIDI signals from the keyboard.

Microcontroller

The microcontrollers that we will utilize in our project are very versatile and very programmable. Testing the ATmega328P on the Arduino Uno R3 board can be done by downloading the Arduino IDE from the Arduino.cc website. Once the IDE has been downloaded and installed, we can verify that the board works as it will be an available option within the setup for the IDE. We can select an Arduino board from a list of provided options and then select the output as the generic COM3 output port.

MIDI-to-CV Converter: MIDI output receiver testing

Our team opted to build the MIDI-to-CV converter from scratch, and the first step of prototyping and testing this is to make sure our breadboard setup is actually receiving the MIDI. We were able to accomplish this by setting up a solderless breadboard as, again, this is for testing purposes only. The breadboard takes MIDI input from a keyboard through a male-to-male MIDI cable connected to a 90-degree female MIDI port on the breadboard. There are three active pins on the MIDI cable which are the center three, known from left to right as pins 4, 2, and 5. Pin 2 is grounded while pin 4 is set to the positive voltage pin on the Arduino Uno in series with a 220 Ohm resistor. Pin 5 is the "data" pin of the MIDI port, this is set to the serial pin TX. The TX pin is also the pin used to connect to the PC used to program the board, thus while testing this will need to be disconnected while the code is pushed to the board. The opto coupler/isolator is placed behind the MIDI port in order to isolate and send the digital MIDI signals to the ATtiny85. This is placed next to, but not in parallel, to the opto coupler/ isolator.

The code was programmed using the native Arduino integration development environment (IDE) in combination with a library that was downloaded from the Arduino website known as SoftwareSerial. The SoftwareSerial library specifically enables the Arduino Uno to interface and communicate alternative serial connections onto the digital pins of the Arduino, hence the name SoftwareSerial.

Figure 7.3: Midi to CV Testing

MIDI signals are included and thus we are able to send them through the Arduino's general purpose input output pins (GPIOs), specifically the serial pin 0 RX and serial pin 1 TX. The code enables the pins and sets them to parse the MIDI information sent in. For testing purposes, we verified that the Arduino, and in turn the ATmega328P, was able to read the MIDI input by toggling the LED associated with the TX pin. In this case if a key was pressed on the keyboard and a MIDI note was generated, the LED would toggle. If no information is sent, the LED would deactivate.

To start the process, the Arduino IDE needs to know which board and which port we are using, these are the Arduino Uno R3 and the serial COM port respectively. After using the in-program validation, any serial connections must be severed before pushing the code. Once the IDE has verified that there are no issues, the code can be compiled and delivered to the board. At this point, the serial pin connections can be remade and thus the code will begin executing.

Once verified that MIDI input was correctly being read, we could utilize an infinite loop to wait for MIDI data as long as the board was powered. This section of the testing almost entirely relies on the SoftwareSerial library's midiSerial method which enables us to direct the pins on the board to the gate, pitch, and trigger values.

7.2. Software Testing

Software testing heavily relies on reading hardware input. To verify the integrity of the boards we have purchased, a test can be run on every digital pin in order to determine whether or not they are able to read data or output signals. For example, if we were to code a program that relies on the use of the serial clock pin (CLK), we can time the intervals of change to determine whether the CLK pin is accurately timed. Likewise we could read gate input to determine whether the generic pin PB2 is being used correctly.

Besides validating that the hardware works correctly, software testing is relatively simple thanks to modern day debuggers that are built into most of the commonly used IDEs such as Eclipse or IntelliJ. We can validate that our code is executing correctly if we are able to produce the guided user interface (GUI) we implemented without any errors. Along with the GUI, the software must be able to accurately (to an extent) display the generated waveform as well as display the parameters that give the displayed waveform. We can validate that the parameters are accurately displayed by adjusting the physical knobs on our synthesizer and verifying that they reflect correctly on the UI.

7.3. Integration

The software-hardware integration will predominantly reside within the MIDI-to-CV converter as this is the source of the switch from digital to analog data. The software side of things will be programmed into the microcontrollers both on the prebuilt microcontroller board as well as the created printed circuit board (PCB). Most of the hardware will reside within the housing for the device which will include the synthesizer itself along with all the components that make it up–the VCO, VCF, VCA, and power supply.

When fully integrated, the device should be able to be powered by an external wall source or battery source and be able to complete its programmed tasks such as reading MIDI data and displaying information for the user on the UI. The device will be able to read in digital MIDI data generated by an electric MIDI keyboard. These digital signals will be processed by the hardware through a MIDI-to-CV converter into analog signals.

After being processed as control voltages, the hardware will be able to manipulate the voltages to create differences in sound, tone, intensity, or other defined parameters. These changes will be made with various dials, switches, or buttons as seen fit. The CVs generated should be able to be audibly heard after their processing through the use of the synthesizer either through external speakers or built-in speakers.

If all the core functionality of the device is working as intended, more quality-of-life features or user experience features can be worked with. These include, but are not limited to, a more modern UI design, additional parameters to change with the CVs, or a richer sound output from the synthesizer. However, as they are quality-of-life features and not critical to have completed, they will be executed on a "do-we-have-time" basis.

In our conceptualized and practiced design using our borrowed MIDI controller, with the synthesizer fully working out, a stretch goal was a working integration to a graphic user interface that generates visualizations based on specific parameters passed from the MIDI controller. This design can be integrated through a simple process using a Java based tech stack, and a simple USB connection from our MIDI controller.

Since we will be using a single device, and our team has followed our milestone planner according to their designated goals we may have a very great chance of achieving this goal. I will summarize the specific case of how we will be connecting to a MIDI controller, as described by documentation by Oracle and known coding practices. Please note that all interfaces will be highlighted in blue, only on its first instance.

[36][37] The Java Sound API has a very flexible message-routing architecture that gives developers and engineers very free control over almost all MIDI data, which also means it will be easier to manipulate this data. Configuration of the MIDI system is handled in the "javax.sound.midi.spi" package. The abstract classes in this package allow service providers to supply and install their own MIDI devices, MIDI file readers and writers, and soundbank file readers. The base module in the API's architecture can be called with "MidiDevice".

This interface includes synthesizers which generate sounds when triggered by MIDI messages. The functionality required of the MIDI ports which we will use is also described in this library. The "Synthesizer" interface extends this interface which describes its additional functionality with sequencers and synthesizers, respectively.

The diagram on the following page shows the interfaces that the Java Sound API uses and how we can convert the data for our synthesizer.

The "Receiver" and "Transmitter" represent the "plugs" that connect these devices together and permit data to flow bidirectionally. Our specific device has multiple ports, but we will only be using one that will represent the "transmitter" - which propagates the incoming messages. This interface and its method are used for setting and querying the receivers so it can send its "MidiMessages".

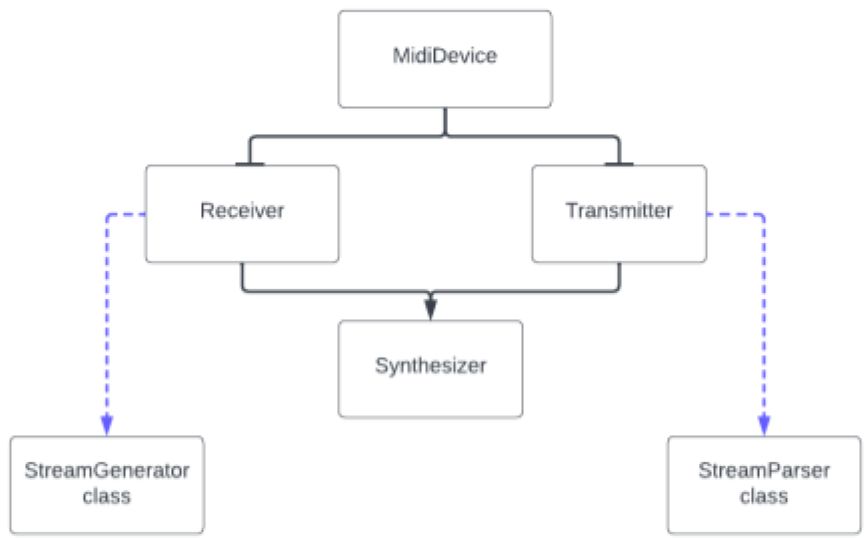


Figure 7.4: Java Interface Outline

The transmitter sends events that it generates itself to its MIDI out receivers. Our API includes concrete classes for then converting between the MIDI objects and the raw byte stream used in the long standing MIDI wire protocol. This is why you see in the diagram the "StreamGenerator". This is a receiver that accepts MidiEvent objects and outputs a raw MIDI byte stream. This is also not far different to what the "StreamParser" does. This is a transmitter that accepts a raw MIDI byte stream and writes the corresponding MIDIEvent object to its receiver.

Dealing with "Real Time" Latency:

In our extensive research with MIDI specification and protocol, we have learned that the Java API actually can help us deal with and optimize any latency issues if this visualizer were to perform eloquently in real time. In the specification document describing MIDIFiles - which are messages stored as events in "sequences" - it is tagged with a timing value. By contrast, messages in MIDI port protocol are always supposed to be processed immediately, so they have no accompanying timing values. This created a real issue for the computer engineers on the team, because we did not know how we were going to measure the absolute timing of the messages. We soon found through documentation that while the timing values that are presented in the objects of the interface of “MidiEvent” are stored in sequences, as said in the specification, the sound API completely bypassed this problem in one of their earlier updates. The timing concept and values are based on musical concepts such as beats, tempo, etc. Each event’s timing measures the time elapsed since the previous stamp.

In contrast to that technique, the sound API using the “Receiver” object always measures absolute time in microseconds. It more specifically measures the number of microseconds elapsed since the device connected to the receiver was opened. This kind of stamp is designed to compensate for latencies that may be introduced by the device used, the operating system, or the application program we are building. We must note that these time stamps should be used for minor adjustments and can not handle complex queues.

Presenting the information passed as a visualization:

Our team’s engineers have tested and understand how to grab data directly from a MIDI device and manipulate it, but one of the largest parts of the stretch goal is learning and understanding how to present it in a functional way.

This is where our team has room to learn and understand how to appeal to the client and user to manipulate this data in a meaningful way. [38] The main resource we have found that has much artistic and technical appeal is the ps.js library that creates visualizations and has previous projects that work with synthesizers, but since our tech stack is based in the Java language this might be difficult to achieve. This is an example of what kind of data art we hope to achieve:

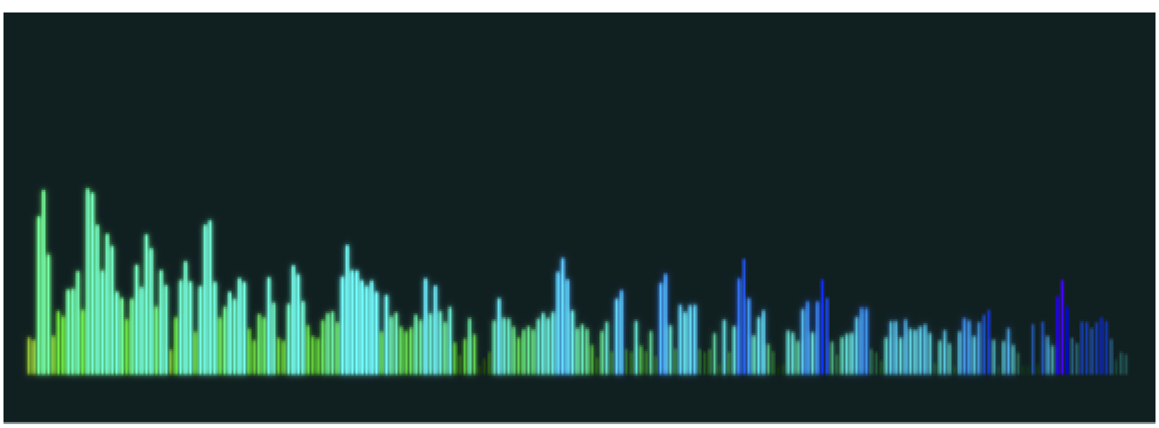


Figure 7.5: Potential Output Visualization

The lines would represent dynamically changing (real-time) parameters such as bpm, pitch, etc.

8. Administrative Content

The following section details administrative content including, but not limited to, milestones and date planning, budgeting and financial decision making, and how to move forward with the progress obtained in Senior Design I in order to better prepare for Senior Design 2.

8.1. Milestone Discussion

The milestones listed below pertain to our time within Senior Design I. The majority of our time taken was spent planning for the final project and in drafting our 120 page report. The deadlines for the report updates are hard set whereas our other preparations follow an "as it's done" schedule as most of the production of the device will reside during Senior Design 2.

Number	Tasks	Start	End	Status	Assignee
Senior Design I					
1	Brainstorm	08/22/2022	08/26/2022	COMPLETE	ALL
2	Project Selection + Role Assignments	08/30/2022	09/20/2022	COMPLETE	ALL
3	Initial Document - Divide and Conquer Version 1.0	09/06/2022	09/16/2022	COMPLETE	ALL
3a	High level overview	09/06/2022	09/16/2022	COMPLETE	ALL
3b	Narrative description	09/06/2022	09/16/2022	COMPLETE	ALL
3c	Requirement specifications	09/11/2022	09/16/2022	COMPLETE	ALL
3d	Block diagrams	09/11/2022	09/16/2022	COMPLETE	ALL
3e	Project budget and financing	09/15/2022	09/16/2022	COMPLETE	ALL
4	Presentation with Dr. Wei/Dr. Richie	09/20/2022	09/20/2022	COMPLETE	ALL
5	Divide and Conquer Version 2.0	09/20/2022	09/30/2022	COMPLETE	ALL
6	Schematics / Breadboard Initial Design	10/03/2022	N/A	COMPLETE	Hardware
7	Software environment setup	10/03/2022	N/A	COMPLETE	Software
8	Halfway Milestone	9/20/2022	11/04/2022	COMPLETE	ALL

	(60pgs)				
9	Parts comparison	10/01/2022	10/15/2022	COMPLETE	ALL
9	Parts list - draft / final budget overview	TBD	N/A	COMPLETE	ALL
10	Parts - Ordered	11/08/2022	N/A	COMPLETE	ALL
11	Test individual parts	11/15/2022	11/22/2022	COMPLETE	ALL
12	Test signal waves/milestone basic goals	TBD	TBD	COMPLETE	ALLL
11	Software API Testing	TBD	N/A	TODO	Software
12	Second to last Milestone (100pgs)	11/04/2022	11/18/2022	COMPLETE	ALL
13	Final Milestone (120 pgs)	TBD	12/06/2022	IN PROGRESS	ALL

Table 8.1: SD1 Milestone Tracker

8.2. Budget and Financing

Outlined in section 8.2 is the relative (non-final) estimate of our cost for this project. The major parts are listed below such as the power supply or microcontroller. Smaller parts are either negligible in cost either due to some team members owning products or there being a readily available supply at the various labs located within the university's engineering building. These smaller parts include resistors, capacitors, and jumper wires. Miscellaneous costs can refer to any non-aforementioned products that may be needed such as glues for emergency repair or additional parts needed for the final product.

Description	**Name**	**Manufacturer**	**Quantity**	**Price (Est)**
Hardware	VCO		3	$8.00
Hardware	VCA		1	$10.00
Hardware	VCF		1	$11.65
Hardware	Power Supply		1	$70.00
Hardware	MIDI Controller		1	$0.00
Hardware	MCV4 MIDI to CV Gate Interface		1	$166.00
Hardware	Bread Board		2	$0.00
Travel Expense	Shipping		-	N/A
Hardware	Custom PCB		1-5	N/A
Hardware	Microcontroller		1	$35.00

Table 8.2: Project Financing

8.3. Project Design Problems

The major problems that could impact project design include availability of parts, the compatibility of various parts from various manufacturers working in tandem, and in some cases, the pricing of components. We strived to acquire a set of parts that would both be beneficial to our success while maintaining a level of financial balance. One such example is opting to create our own MIDI-to-CV converter instead of relying on a pre-manufactured device that would cost much more.

As of the completion of Senior Design I, the parts we have procured are able to work on their own. There were two stages of testing: enabling the reading of MIDI data from human input to convert these to CV signals and the translation of CV signals into waveforms utilizing the VCO.

8.4. Conclusion and SDII Plan

This project allowed us to have full control over the type of monophonic synthesizer we build and how it interacts with the channels, notes and software.This will also allow us to experience using our own acquired software and hardware knowledge by applying concepts learned in linear circuits, electronics, differential equations, embedded systems, and computer science in a real world application. It is a great way to understand the connection music has in the engineering world and its roots, as well as how we can push the limits of what we already know we can manipulate and control regarding sound waves and transmission between channels. Overall, we want to use this as a learning experience to hone our skills that we have acquired over the years.

Senior Design II					
14	Begin creation of PCB and prototyping	~January 2023	~January 2023	TODO	Hardware
15	Initial interfacing of software to hardware	~January 2023	~January 2023	TODO	Software
16	Begin drafting final document	~January 2023	~January 2023	TODO	ALL

17	Heavy testing phase - software bug fixing	~February 2023	~March 2023	TODO	Software
18	Heavy testing phase - hardware alterations	~February 2023	~March 2023	TODO	Hardware
19	Finalize prototype	~February 2023	~March 2023	TODO	Hardware
20	Revise final document	~February 2023	~April 2023	TODO	ALL
21	Deliver final project presentation	~May 2023	~May 2023	TODO	ALL

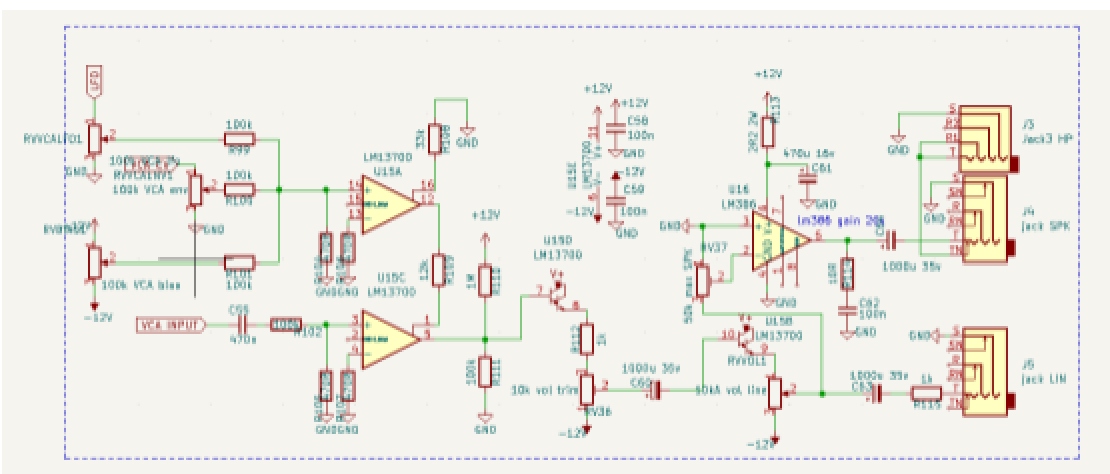
Table 8.3: SD2 Milestone Tracker

9. Reflections

The following section was written after the completion of Senior Design 2, our final presentations, as well as the Senior Design Showcase. They will include our finalizing thoughts, our affirmations and successes, as well as our frustrations or oversights during the process of our project's lifecycle.

9.1. Hardware

When working with some of the modules, it is not uncommon to encounter issues that require modifications to the original design. In some cases, these modifications were necessary to get the module to work properly in conjunction with the other modules. For example, when testing the VCA module, it became apparent that while testing the IC we struggled and were not able to make it work properly while prototyping; certain components need to be replaced and added however the output from the AS3330 was not what we desired. While making careful considerations we adjusted the VCA design to use the LM13700. The new VCA design is pictured below:



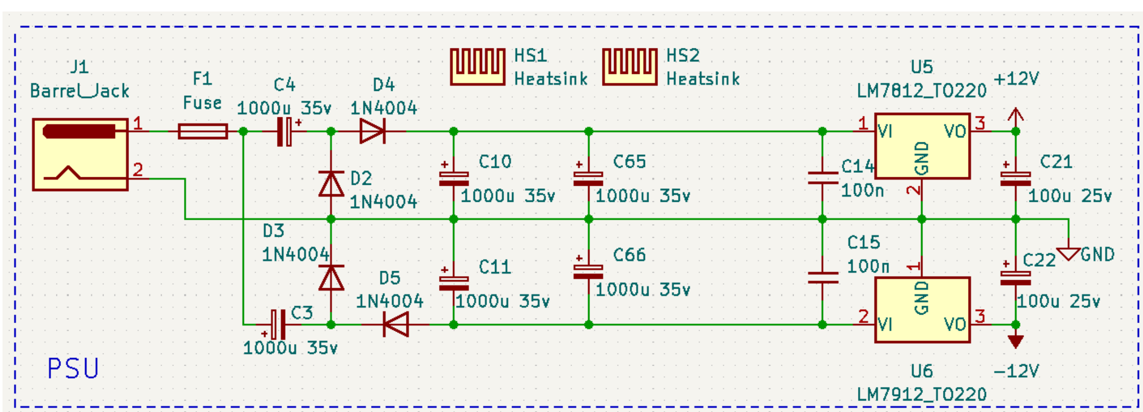
We may note that the LM13700 is not as great as the AS3330 in terms of specs which is compared in Table 3.6 however this OTA (Operational Transconductance Amplifier) was available to us at the senior design lab and plenty of material was available for us to create a VCA using this component. Something to note is that this design was created with a Eurorack "Vintage VCA" in mind.

By carefully evaluating the design and conducting thorough testing, it was possible to make the necessary modifications and achieve the desired performance from the module.

Another issue that was encountered was with the power supply. We encountered a number of issues while prototyping the first design of the power supply which used the LM237 and the LM317T voltage regulators. First while testing we realized that we used 500mA wall wart which didn't meet the specific load for the system which then led us to change the wall wart to one that supports 1 A. After correcting this we realized that the ripple was too large and on the oscilloscope and that it never reached a steady state. In the process we blew a capacitor which led us to consider a change in the design.

In the second design a linear power supply was made using an LM7812 and LM7912. Contrary to the first design this new one contains a full wave rectifier additionally the LM7812 has a built in thermal shutdown feature that protects the device from overheating and damaging the circuit which in other terms saves time while testing since we know that this voltage regulator will not damage the circuit.

Below is the new design of the power supply used:



Overall, the success of the prototyping stage of this power supply was likely due to the thorough testing, experimentation, and material available that went into developing the circuit. By carefully evaluating and adjusting the design at each stage of the prototyping process, you were able to create a robust and reliable power supply that met the requirements to power each module of the synthesizer.

9.2. Software

Overall, the software was not difficult to implement. The MIDI-to-CV software, on a very high level, consists of code that awaits for a MIDI signal before it kicks into action. Once it gets the signal, it breaks it down and looks for its tone. This tone pertains to a certain voltage which is then outputted by the ATTiny85–the IC chip responsible for the MIDI-to-CV conversion. The software specifically for the MIDI-to-CV conversion did not give us too many issues. The only numbers we had to adjust were the MIDI note bounds which were defined at the start of the code. All the ICs in our design had dedicated slots soldered into the final PCB design which allows us to easily use through hole-based ICs as we could easily program them externally and then place them into their slots. This helps us skip the step of soldering the IC in order to program it.

Our biggest issue we ran into for the MIDI-to-CV conversion was actually forgetting to bend away the unused MIDI port pins. The MIDI port consisted of five pins, hence the name 5-Pin DIN, however only pins 2, 4, and 5 are used. In order to not have unwanted flow of electricity we needed to bend away pins 1 and 3. While testing the breadboard, we forgot this part which delayed the completion of testing the MIDI-to-CV conversion on the breadboard. However, this did not affect the PCB design or final PCB print in any way.

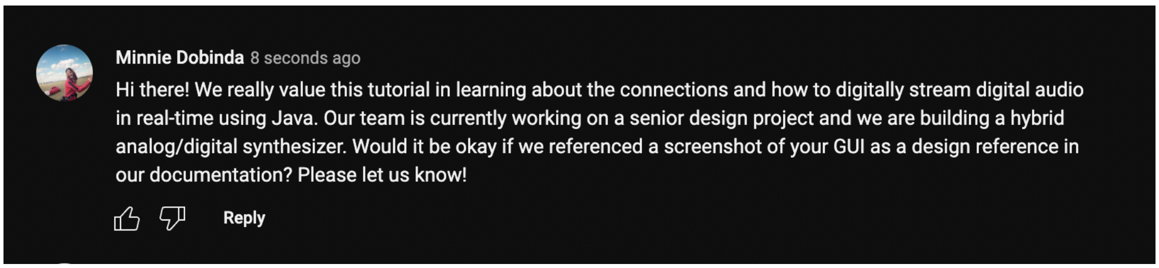
The rest of the software lied within stretch goals. Our intent was to have our users be able to utilize a Guided User Interface (GUI) in order to visualize the values of the voltages they are outputting with their MIDI notes. The final PCB design consisted of multiple potentiometers that all pertained to a different value that they adjusted whether that be their pitch, frequency, oscillation, etc. If we were able to successfully incorporate our GUI into the design, a separate screen would be attached to the device with various jumper wires attached at different junctions where the values of frequency, amplitude, and other variables can be measured. As will be explained in the next section for another component, incorporating this design into the final PCB design would have been problematic for our team.

The GUI would also incorporate a live-time spectrum analyzer; this would display the waveform generated by the given waveform type as well as the values that affect the output waveform. The spectrum analyzer was successfully built and tested before the final PCB design, however, due to the complex nature of combining several parts together with a grand total of over 200 components, inserting the spectrum analyzer into the final design would have added potential complications when we were close to deadlines. Thus, it was in our team's best interest to scrap the design.

10. Appendices

The following section outlines all available copyright permissions for written work or at least the request to utilize them, a reference of used datasheets throughout the document, and a list of references that have helped us write our document. All original works belong to their respective authors.

10.1. Appendix A: Copyright Permissions

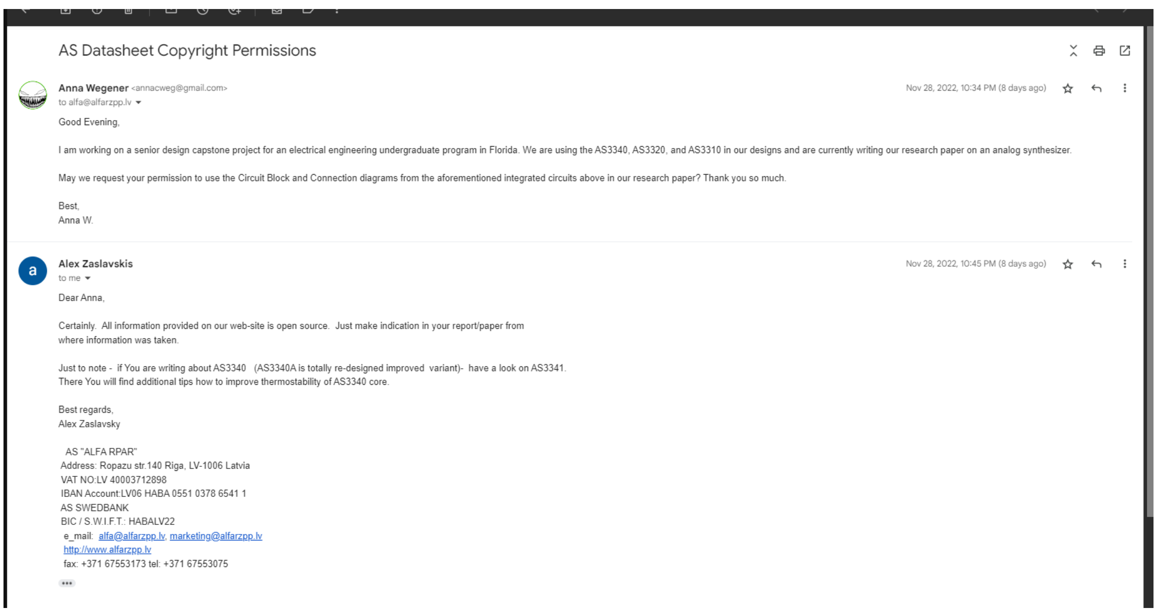


Minnie Dobinda 8 seconds ago

Hi there! We really value this tutorial in learning about the connections and how to digitally stream digital audio in real-time using Java. Our team is currently working on a senior design project and we are building a hybrid analog/digital synthesizer. Would it be okay if we referenced a screenshot of your GUI as a design reference in our documentation? Please let us know!

Reply

Copyright permission request for GUI example



AS Datasheet Copyright Permissions

Anna Wegener <annacweg@gmail.com>
to alfa@alfarzpp.lv
Nov 28, 2022, 10:34 PM (8 days ago)

Good Evening,

I am working on a senior design capstone project for an electrical engineering undergraduate program in Florida. We are using the AS3340, AS3320, and AS3310 in our designs and are currently writing our research paper on an analog synthesizer.

May we request your permission to use the Circuit Block and Connection diagrams from the aforementioned integrated circuits above in our research paper? Thank you so much.

Best,
Anna W.

Alex Zaslavskis
to me
Nov 28, 2022, 10:45 PM (8 days ago)

Dear Anna,

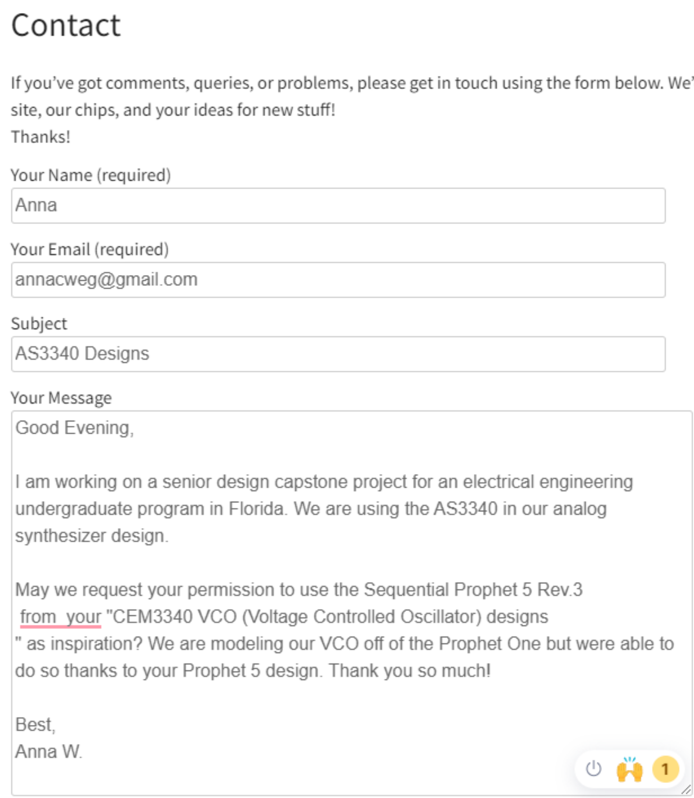
Certainly. All information provided on our web-site is open source. Just make indication in your report/paper from where information was taken.

Just to note - if You are writing about AS3340 (AS3340A is totally re-designed improved variant)- have a look on AS3341. There You will find additional tips how to improve thermostability of AS3340 core.

Best regards,
Alex Zaslavsky

AS "ALFA RPAR"
Address: Ropazu str.140 Riga, LV-1006 Latvia
VAT NO:LV 40003712898
IBAN Account:LV06 HABA 0551 0378 6541 1
AS SWEDBANK
BIC / S.W.I.F.T.: HABALV22
e_mail: alfa@alfarzpp.lv, marketing@alfarzpp.lv
http://www.alfarzpp.lv
fax: +371 67553173 tel: +371 67553075

Copyright permission request for datasheet circuit block diagrams


Contact

If you've got comments, queries, or problems, please get in touch using the form below. We'd love to hear your feedback about the site, our chips, and your ideas for new stuff!
Thanks!

Your Name (required)
Anna

Your Email (required)
annacweg@gmail.com

Subject
AS3340 Designs

Your Message
Good Evening,

I am working on a senior design capstone project for an electrical engineering undergraduate program in Florida. We are using the AS3340 in our analog synthesizer design.

May we request your permission to use the Sequential Prophet 5 Rev.3 from your "CEM3340 VCO (Voltage Controlled Oscillator) designs " as inspiration? We are modeling our VCO off of the Prophet One but were able to do so thanks to your Prophet 5 design. Thank you so much!

Best,
Anna W.

Copyright permission request for Sequential Prophet VCO model (still pending)

10.2. Appendix B: Datasheets

AS3340 / AS3345, AS3340A / AS3345A - Voltage Controlled Oscillator (VCO) by "ALFA RPAR" Joint Stock Company ALFA

AS3320 - Voltage controlled filter (VCF) by "ALFA RPAR" Joint Stock Company ALFA

AS3310 - ADSR Voltage Controlled Envelope Generator by "ALFA RPAR" Joint Stock Company ALFA

10.3. Appendix C: Works Cited

[1] Edwards, Nate. "Synthesizer Basics Explained - with Audio Samples." *InSync*, 7 Oct.2022, https://www.sweetwater.com/insync/synthesizer-basics-explained-with-audio-samples/.

[2] Mma. Summary of MIDI 1.0 messages. Retrieved December 6, 2022, from https://www.midi.org/specifications-old/item/table-1-summary-of-midi-message

[3][4][5] Edwards, Nate. “Synthesizer Basics Explained - with Audio Samples.” *InSync*, 7 Oct.2022, https://www.sweetwater.com/insync/synthesizer-basics-explained-with-audio-samples/.

[6] *Sample-based synthesis*. Apple Support. Retrieved December 5, 2022, from https://support.apple.com/guide/logicpro/sample-based-synthesis -lgsife418f0c/10.7.3/mac/11.0

[6] Douglas, A. (2021, October 21). Monosynth vs. Polysynth: Which Should I Buy? reverb.com. https://reverb.com/news/monosynth-vs-polysynth -which-should-i-buy

[7] Gaston, R. (2019, February 28). *Learning synthesis: Filters*. Perfect Circuit. Retrieved December 6, 2022, from https://www.perfectcircuit.com/ signal/learning-synthesis-filters

[8] Douglas, A. (2021, October 21). Monosynth vs. Polysynth: Which Should I Buy? reverb.com. https://reverb.com/news/monosynth-vs-polysynth -which-should-i-buy

[9][10] Rod Elliott - Elliott Sound Products. *Class D audio amplifiers - theory and Design*. Class-D Amplifiers. Retrieved December 5, 2022, from https://sound-au.com/articles/pwm.htm

[11] Raspberry Pi Ltd. “Raspberry Pi Datasheets.” *BCM2711 ARM Peripherals*, 18 Jan. 2022, https://datasheets.raspberrypi.com/bcm2711/bcm2711-perip herals.pdf.

[12] Stoffregen, Paul J, and Robin C Coon. “Teensy® 4.0 Development Board.” *PJRC*, https://www.pjrc.com/store/teensy40.html.

[13] Atmel Corporation. “Atmel 2586 AVR-8 Bit Microcontroller ATTINY25 - Microchip Technology.” Atmel ATtiny25, ATtiny45, ATtiny85 Datasheet, 2013, https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-2586-AVR-8-bit-Microcontroller-ATtiny25-ATtiny45-ATtiny85_Datasheet-Summary.pdf.

[14] Atmel Corporation. “Arduino.” *ATmega328P - 8-Bit AVR Microcontroller with 32K Bytes In-System Programmable Flash*, 2015,

https://content.arduino.cc/assets/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf.

[15] *USB MIDI 2.0*. The MIDI Association. Retrieved December 6, 2022, from https://www.midi.org/specifications/midi-transports-specifications/usb/usb-midi-2-0-2

[16][17] *Modular synthesizer power supplies and distribution: A thorough introduction*. Rabid Elephant. Retrieved December 5, 2022, from https://rabidelephant.com/blogs/general/modular-synthesizer-power-supplies-and-distribution-a-thorough-introduction

[18] *Home*. chillibasket. Retrieved December 5, 2022, from https://wired.chillibasket.com/2020/06/dual-power-supply/

[19] *USB MIDI 2.0*. The MIDI Association. (n.d.). Retrieved December 6, 2022, from https://www.midi.org/specifications/midi-transports-specifications/usb/usb-midi-2-0-2

[20] *Universal asynchronous receiver/transmitter (UART) for Keystone Devices UG*. Retrieved December 6, 2022, from https://www.ti.com/lit/ug/sprugp1/sprugp1.pdf

[21] *Official MIDI Specifications*. MIDI. Retrieved December 6, 2022, from https://www.midi.org/specifications/midi-transports-specifications/usb

[22] *USB MIDI 2.0*. The MIDI Association. Retrieved December 6, 2022, from https://www.midi.org/specifications/midi-transports-specifications/usb/usb-midi-2-0-2

[23] *Official MIDI Specifications*. MIDI. Retrieved December 6, 2022, from https://www.midi.org/specifications/midi-transports-specifications/usb

[24] *Package javax.sound.midi*. javax.sound.midi (Java Platform SE 8). (2022, October 11). Retrieved December 6, 2022, from https://docs.oracle.com/javase/8/docs/api/javax/sound/midi/package-summary.html

[25][26][27] Rod Elliott - Elliott Sound Products. *Class D audio amplifiers - theory and Design*. Class-D Amplifiers. Retrieved December 5, 2022, from https://sound-au.com/articles/pwm.htm

[28] *Trail: Sound*. Trail: Sound (The Java™ Tutorials). Retrieved December 6, 2022, from https://docs.oracle.com/javase/tutorial/sound/index.html

[29] Last Minute Engineers. (2022, July 21). *In-depth: How rotary encoder works and interface it with Arduino*. Last Minute Engineers. Retrieved December 5, 2022, from https://lastminuteengineers.com/rotary-encoder-arduino-tutorial/

[30] Kevin. (2021, July 12). *Attiny85 MIDI to CV*. Kevin's Blog. Retrieved December 5, 2022, from https://emalliab.wordpress.com/2019/03/02/attiny85-midi- to-cv/

[31][32][33] Atmel Corporation. "Atmel 2586 AVR-8 Bit Microcontroller ATTINY25 - Microchip Technology." Atmel ATtiny25, ATtiny45, ATtiny85 Datasheet, 2013, https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-2586-AVR-8-bit-Microcontroller-ATtiny25-ATtiny45-ATtiny85_Datasheet-Summary.pdf.

[34][35] Kevin. (2021, July 12). *Attiny85 MIDI to CV*. Kevin's Blog. Retrieved December 5, 2022, from https://emalliab.wordpress.com/2019/03/02/attiny85-midi- to-cv/

[36] Kula, K. (2020, February 11). *Web MIDI API-sending notes from JavaScript to your synth*. Medium. Retrieved December 5, 2022, from https://kulak.medium.com/web-midi-api-sending-notes-from-javascript-to-your-synth-1dfee9c57645

[37] *Package javax.sound.midi*. javax.sound.midi (Java Platform SE 8). (2022, October 11). Retrieved December 6, 2022, from https://docs.oracle.com/javase/8/docs/api/javax/sound/midi/package-summary.html

[38] Lacey, L. (2015, October 30). *Create Your Own MIDI-Generated Realtime Visuals with openFrameworks*. https://ask.audio/articles/create-your-own-midi-generated-realtime-visuals-with-openframeworks

[39] Johnson, N. M. + J. (2021, July 21). *Creative control strategies: MIDI TO CV*. Perfect Circuit. Retrieved December 5, 2022, from https://www.perfectcircuit.com/signal/midi-to-cv

[40] Kula, K. (2020, February 11). *Web MIDI API-sending notes from JavaScript to your synth*. Medium. Retrieved December 5, 2022, from

https://kulak.medium.com/web-midi-api-sending-notes-from-javascript-to-your-synth-1dfee9c57645