Monophonic Hybrid Analog/Digital Synthesizer

Magdalena Dobinda, Anna Wegener, Sebastian Carrion, and Devin Vanzant.

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816, U.S.A.

***Abstract* — This paper presents the design methodology and stages of building a monophonic analog synthesizer that is playable through digital Musical Instrument Digital Interface (MIDI) devices, converting digital signals to analog. A synthesizer is an electronic music instrument that generates audio signals. This hybrid form of a synthesizer bridges together vintage-style electrical design techniques of implementation of synthesis while adapting to the modern digital era of music. We will discuss the construction and materials we use to achieve a base synthesizer and the process of integrating a custom digital-to-analog converter into it.**

***Index Terms* — Synthesis, MIDI, Monophonic, DAW, FFT Analysis, V/OCT Method, Subtractive Synthesis, Additive Synthesis.**

INTRODUCTION

Synthesizers have revolutionized music production by enabling musicians to create a vast range of sounds and textures that were previously impossible to achieve with traditional instruments. Originally synthesizers were marketed towards academics, and musicians did not start utilizing them until the late 1960's when they became smaller with the use of transistors - and more user friendly due to the integration of built-in keyboards. Among the various types of synthesizers, monophonic analog/digital synthesizers have garnered popularity due to their ability to generate rich, warm, and dynamic sounds. These synthesizers are designed to produce a single note at a time, allowing musicians to create melodies, basslines, and lead sounds.

The most widely used form of synthesizers currently is digital. Unlike analogue synthesizers, which produce music using real analogue circuitry, digital synthesizers emulate analogue sounds with digital signal processing techniques. The issue most users have with this type of synthesizer is that it can never exactly mimic the raw sound produced by a real vintage analog synthesizer. Our group aims to solve this by combining the two.

Our paper will divide the sections between analog and digital for clarity in following the goals and results of each part of the project.

I. ANALOG SYNTHESIZER

An analog synthesizer is a synthesizer that uses analog circuits and signals to generate sound electronically. The voltage coming from the power source generates a waveform by oscillating electrons.

A. Subtractive Synthesis

Almost all analog synthesizers operate using a method known as subtractive synthesis. Essentially, you subtract parts of the frequency spectrum, consisting of the fundamental tone and associated harmonics. This is a staple and the foundation of how almost all synthesis and electronic instruments operate to this day. You begin with a waveform generated by the oscillator, such as a square wave or sawtooth wave. The signal is sent from the oscillator to a filter that represents the frequency-dependent losses and resonances in the body of the instrument. This is used to "subtract" the oscillator's harmonics. A low-pass filter removes frequencies above the cutoff frequency, which dulls the sound. Inversely a high pass filter cuts frequencies below the cutoff point to tin the tone. Modulating the cutoff frequency has a large impact on the sound you create, and it will be completely unique to the user. This is the real strength of subtractive synthesis that you cannot recreate with other types of instruments. This is the type of synthesis we will be utilizing in our project. [1]

B. Waveforms

In most analog oscillators, one waveform is produced, usually either a triangle or sawtooth wave, and then run through the filters to alter the waveform into the other shapes. This is where the terms triangle core and sawtooth core come from.

The shapes of the actual waveforms produced by synthesizers are often approximations of the standard waveforms and sometimes they don't even look close to what they claim to represent. This is part of what creates the unique character of analog waveforms.

1. Triangles and Squares

Triangle waves and square waves both contain only odd harmonics, but their amplitude decreases with different rates. Each successive harmonic in a square wave is a third as low in amplitude, and about a tenth as low in a triangle wave. Note that all of these waves are measures in Hertz. Both a square wave and a triangle wave at 100Hz would have harmonics at 300Hz, 500Hz, 700Hz, continuing at every odd multiple, but because they decrease at different rates the shape is affected differently. The same note from a flute will be a triangular wave.

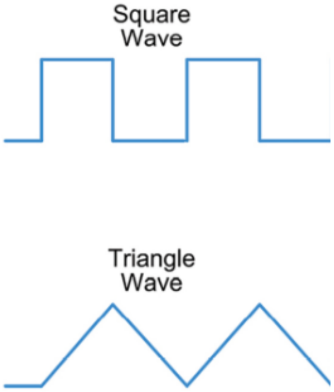

Fig. 1. Square and Triangle waves

2. **Sawtooth**

The sawtooth wave (or saw wave) is a kind of non-sinusoidal waveform. While a square wave is constructed from only odd harmonics, a sawtooth wave's sound is harsh and clear, and its spectrum contains both even and odd harmonics of the fundamental frequency. It is one of the best waveforms to use for subtractive synthesis of musical sounds, and it is the core of vintage sounds. The sawtooth wave has almost uniform integer harmonics on its fundamental frequency. The same note played on a trumpet will look like a saw-tooth wave. [2]

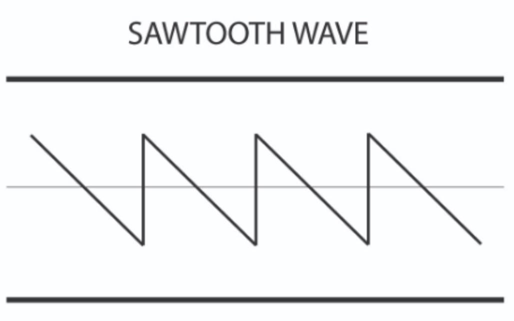

Fig. 2. Sawtooth wave.

C. Oscillators

This is the primary source of sound in our project. It works the same way a guitar string pluck works. When a musician plucks a string, the string vibrates enough to produce a signal at a constant rate. Oscillators also control the pitch or frequency, or vibration speed, measured in Hertz.

D. Filters

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 low-pass, band-pass, or high pass. 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.

E. Amplifiers and Envelope Generators

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.

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. [3]

F. Requirements for our Analog Synthesizer

Such an analog synthesizer described in this section will satisfy the following requirements:

1. **Musical Instrument Digital Interface (MIDI) to Control Voltage (CV):** MIDI digital notes are shown to be analog by reading voltage output after pressing a note. *Notes must go up 1V for each octave.*
2. **Voltage Controlled Oscillator (VCO)**: A dual Voltage Controlled Oscillator 20Hz to 20kHz. By using two, the versatility of our product doubles as you can get 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

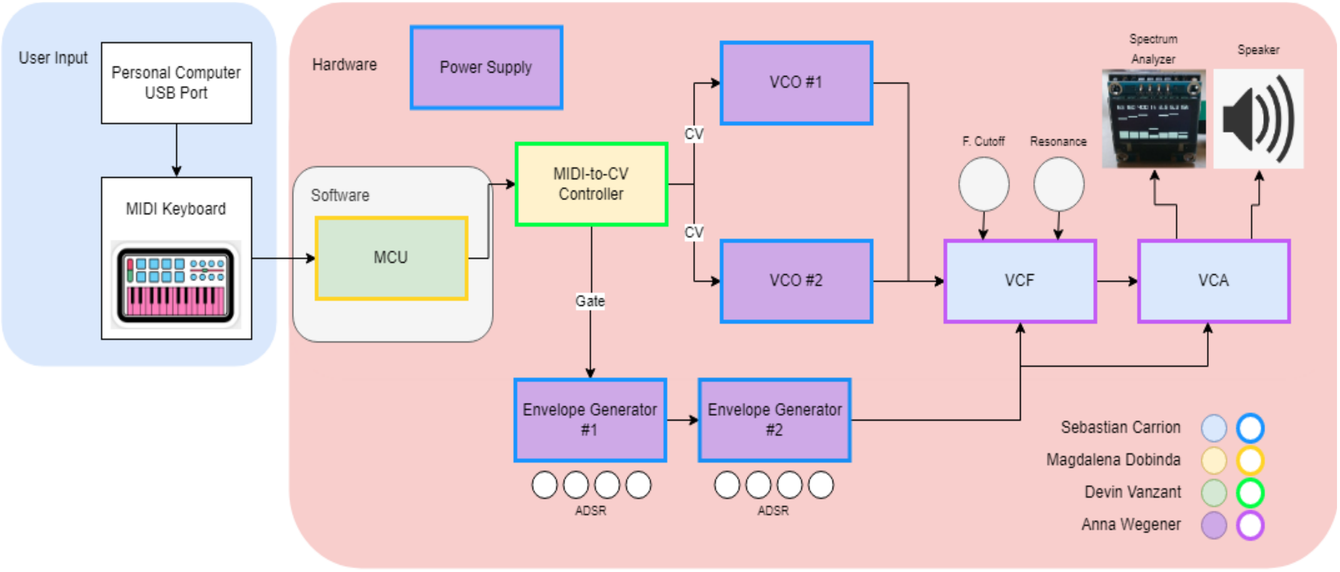

Figure 3 - Overall Block Diagram

for sub bass. We must see this *produce sawtooth, square and triangle waves.*

3. **Voltage Controlled Amplifier (VCA):** Used to increase the control voltage to amplify the signal to an audible level
4. **Voltage Controlled Filter (VCF):** Used to adjust cut-off frequency through an input-controlled voltage for a frequency range of 20Hz to 21kHz. We must prove that *we are able to control cutoff frequency and resonance.*
5. **Envelope Generator:** Manages the loudness contours in our synthesizer. All four components of Attack, Decay, Sustain and Release should show results in the characteristics of the sound produced.

These requirements and results will be discussed in depth in the results section of this paper.

G. Design Overview

Figure 3 shows a functional block diagram of the system demonstrating the control voltage and audio signal paths. The synth will take input through a MIDI cable that can either be connected to a MIDI keyboard or Personal Computer (PC) with a Universal Serial Bus (USB)-MIDI interface. For our project, we decided to use a MIDI keyboard as the main input source. You can also see in this block diagram the way in which the workload for the synthesizer project was split up in the key. The main filled color of the boxes indicates that the person had the primary role of that component, and the outline of the box is the person assigned this component as a secondary role.

The MIDI data will be sent over a 5-pin "Deutsches Institut fur Normung" (DIN) cable to the microcontroller unit (MCU) which we have programmed to convert MIDI data into the respective voltages that correspond to the correct notes. The MIDI-to-CV controller outputs this voltage (known as a "control voltage") to the VCO which controls the pitch of the oscillator.

When a key is pressed, the MIDI-to-CV controller also outputs a gate signal of 5V to enable the 2 Envelope Generators– when a key is not pressed, the MIDI-to-CV controller outputs 0V. These Envelope Generators also send a separate control voltage to both the VCF and VCA to control the frequency cutoff of the filter and amplitude of the amplifier.

The audio output of the VCOs feeds into the VCF to allow the user to control both filter cutoff and resonance. The VCF then sends this audio signal into the VCA where it is then amplified to an audible level.

We were able to pick and choose designs to replicate a vintage style analog synthesizer that withstands the test of time in today's digital world operating on MIDI. The majority of the audio module designs were based on the manufacturer Alfa's AS chip line datasheets. This AS series of ICs was a reissue of the popular CEM series that was used in the most famous synthesizers in the 70s and 80s that we were able to base our modules on. The amplifier was based on a vintage style LM13700 design. The power supply was based off a dual output 12V wall wart design. The MIDI to CV converter was based off an open-source design. We were able to test these modules on breadboard individually, integrate them, and finally build them on PCB.

II. DIGITAL

Our synthesizer is also digital in nature considering it is controlled via a protocol called "MIDI." This is in contrast to the first analog synthesizers in the 1960s and 1970s that relied strictly on control voltage paired with limited sequencers. Many synthesizers nowadays are purely digital synthesizers through digital signal processing and digitally controlled oscillators [4]. Our synthesizer is combination of both domains in that the subtractive synthesis modules are purely analog and controlled via digital MIDI protocol.

A. MIDI Protocol

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 [6]. 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. It 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. MIDI supports up to 16 channels of simultaneous sound. Each channel can have its own melody, rhythm, and instrument.

B. MIDI Messages

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. 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.

C. Control Voltage

Control Voltages 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.

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.

TABLE I

SUMMARY OF CV COMPONENTS

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 I illustrates the different CV components and their definitions. The concept of CV is standardized on analog synthesizers and has two main implementations:

Table II compares notes and corresponding voltage levels. In this example, we use 1V per octave, and 55 Hz/V. There are two formulas that link these voltage methods. They are represented in equations (1) and (2).

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

TABLE II

CONVERSION METHOD NOTE-TO-VOLTAGE RATIO

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

voltage increases by 1V per octave. Each 1V octave is divided linearly into 12 semitones [5].

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 [5].

$$V_{Oct} = \ln_2(V_{Hz}) + 1 \tag{1}$$

$$V_{Hz} = 2^{V_{oct}-1} \tag{2}$$

D. MIDI-CV Conversion

A MIDI-CV Converter 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 specifically, 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. Our team decided to build a MIDI-CV from scratch rather than buying a standalone one to allow the4 computer engineers full control over the MCU code, and vital parameters such as having a DIN-MIDI port rather than a USB-MIDI port for successful conversion [6].

E. Goals for Digital Section of Synthesizer

The goals we had for this part of the project are simple, we wanted to make sure that we:

1) Are receiving MIDI data in their full packets to the MCU

2) Once communication with the MIDI device was successful, we wanted to check that the code we are writing and the circuit connections made are successfully converting the digital signals to CV for the analog synthesizer to receive.

We tested this with the V/OCT Method by measuring the voltage being passed through on the VCO each time we played a higher note. We achieved this goal but should note for clarity that this converter does not work bidirectionally, and we customized it for the purpose of MIDI-CV conversion and not the other way around.

For quantifiable specification:

1. All 128 Notes of the MIDI spectrum were successfully received by the MCU by printing the correct note on the terminal console on our system.
2. MIDI digital notes are shown to be analog by reading voltage output after pressing a note. Notes go up *1V for each octave.*

III. TESTING & RESULTS

In this section we will discuss the testing and results of the project in order:

A. 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 design with additional circuitry to see anything meaningful and achieve our desired waveshapes. In doing so, we were able to observe both if our IC was working and if our schematic design was successful.

After proving the basic sawtooth wave, we were able to add the additional surrounding circuitry to produce square and triangle waves as per our requirements. The following waveforms can be seen in figure 4.

Fig. 4. Oscilloscope Results

B. VCF

In the audio signal processing chain, the VCF was next in line. This module used the output of the VCO to adjust its cutoff and resonance characteristics.

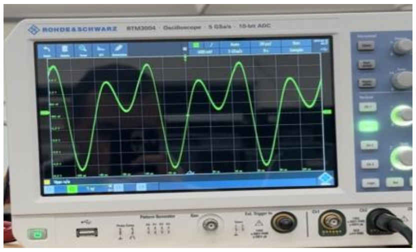

Fig. 5. VCF Oscilloscope Results

The team successfully tested the entire VCF and verified correct +/-12V voltages. Once this was proved in conjunction with the VCO we could see the output of the ramp wave completely and were able to modulate it via the cutoff and resonance potentiometers. See Figure 6 for the results on an oscilloscope.

C. VCA

The VCA testing was a success as we were able to verify that it amplified the signal. We used a sawtooth wave input to and we used a potentiometer to control voltage input while using an audio input to the VCA signal input. We then varied the control voltage and observed a corresponding increase or decrease in the amplitude of the audio output signal, indicating that the VCA was functioning correctly. We also observed that the VCA output signal was able to reach both negative and positive voltage ranges as expected.

D. Envelope Generator

After verifying the VCA and VCF can successfully modulate the oscillator's waveform, we moved onto manipulating the sound envelope itself. To test the envelope generator, we created the circuit and probed 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.

E. Keyboard (Receiving MIDI Data)

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 an audio 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.

F. MIDI-CV Testing

Our team decided to create a MIDI-to-CV converter from scratch. We started by testing our breadboard setup to ensure that it was receiving the MIDI signal properly. Weused a solderless breadboard for testing purposes and connected it to a male-to-male MIDI cable which was plugged into a 90-degree female MIDI port on the breadboard. The MIDI cable had three active pins, which were pins 4, 2, and 5, from left to right (see figure 6).

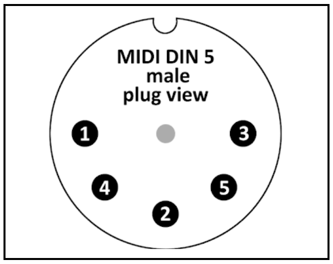

Fig. 6. 5-Pin DIN Layout

Pin 2 was grounded while pin 4 was connected to the positive voltage pin on the Arduino Uno using a 220 Ohm resistor. Pin 5 was connected to the serial pin TX, which was also used to connect to the PC used for programming. We added an optocoupler/isolator to isolate and send the digital MIDI signals to the ATtiny85 [6].

Using the Arduino IDE and the SoftwareSerial library downloaded from the Arduino website, we programmed the code to enable the Arduino Uno to interface and communicate alternative serial connections onto the digital pins of the Arduino. This allowed us to parse MIDI information sent through Arduino's general purpose input output pins (GPIOs), specifically the serial pin 0 RX and serial pin 1 TX. We tested the code by verifying that the Arduino and ATmega328P were able to read the MIDI input and toggled the LED associated with the TX pin when a key was pressed on the keyboard generating a MIDI note.

When testing if the MIDI notes were converted to analog, we saw successful results as the Voltages correctly went up in octaves and the oscilloscope read the signals correctly. See figure 7 for the results of the analog conversion.

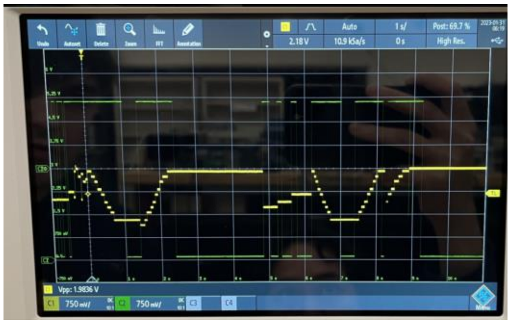

Fig. 7. MIDI-CV Voltage Test Results. Green = Gate Signal, Yellow = CV Out.

G. PCB Testing

We designed our PCB in Kicad due to its free accessibility and rich functionality. The AS chip series is not easy to find footprints for but they were readily available in the Kicad Audio library for our VCO, VCF, and two envelope generators.

Our PCB is a two-layer board with the user-interface devices like potentiometers and switches on the top layer and all connectors, ICs, variable resistors, and large through-hole components on the bottom layer. Doing so and using surface mount resistors and capacitors allowed us to keep a hardware intensive project onto one 5” x 10” board. See Figure 8 for PCB Layout.

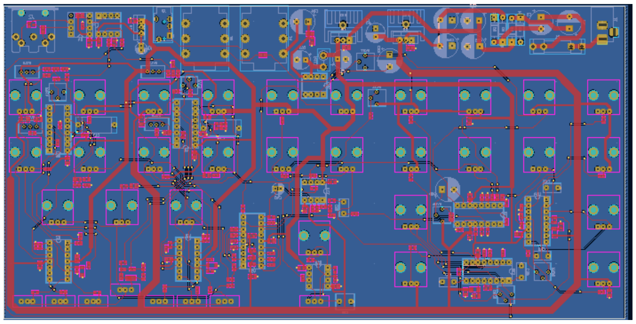

Fig. 8. PCB Layout

IV. STRETCH GOALS AND BEYOND

We consider stretch goals to be anything pertaining to our project that was not a part of the original plan. Whether due to time constraints, monetary issues, or overall over-complications, these goals were not met, but did not take away from the core device. This means that outside of the aforementioned components, the following will be considered a stretch goal.

1. Spectrum Analyzer

One such stretch goal we aimed to incorporate into our synthesizer is a spectrum analyzer. Spectrum analyzers are devices that measure the magnitude of an input signal given a frequency range. In our case, our input signal is the generated control voltage from our MIDI keyboard or computer-generated MIDI notes.

The frequency range can be defined in the code set on the microcontrollers by adjusting the sampling frequency. This is how often a sample of sound is taken within one second. For our purposes, we utilized a sampling frequency of 38.64KHz. According to Nyquist’s Theorem, the sampling frequency is actually double of what the highest possible frequency of the control voltage can be, thus our actual frequency displayed on the spectrum analyzer will only be 19.32KHz, however, this is more than enough for our needs.

We converted the OLED to display 64 frequency bin bars by using fast Fourier transform analysis (FFT). See figure 9 pertaining to this explanation below.

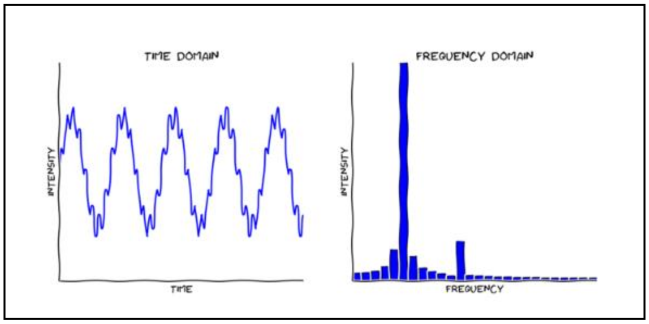

Fig. 9. Fourier transform application

The graph on the left represents a complex signal in the time domain, like what a microphone might produce. This signal is the sum of two sine waves. You can see a low frequency sine wave with high intensity gives the signal its overall up and down shape. However, a higher frequency sine wave with a lower intensity is added to the signal so it has small rough edges that protrude out as it rises and falls. The graph on the right is the result of running a Fourier transform on the signal at the left. You can see the Fourier transform output as a histogram, or bar graph, of the intensity of each frequency. It's apparent that two frequencies - the two spikes in the graph - have much stronger intensities than the others. These frequencies represent the frequencies of the two sine waves which generated the signal. The output of the Fourier transform is nothing more than a frequency domain view of the original time domain signal [7].

We implemented this analysis through use of the Arduino built in library named *“fix_fft”* and applying it to the methods described in the documentation. We could not implement this stretch goal in our final project as the PCB had already been ordered, but we learned a lot about the capabilities of our synthesizer.

V. CONCLUSION

In conclusion, this paper on the monophonic hybrid analog/digital synthesizer was the result of a collaborative effort by a team of individuals with diverse perspectives and areas of expertise. Through effective communication, sharing of ideas, and constructive feedback, we were able to produce a comprehensive analysis of this type of synthesizer. Our research explored the history, components, and impact of the monophonic hybrid analog/digital synthesizer, providing valuable insights into its unique sound qualities, versatility, and ease of use. This project has been an excellent opportunity for the team members to develop our research and writing skills while learning from each other. Our collaborative effort highlights the importance of teamwork and how working together can lead to a more in-depth analysis and understanding of complex subjects. We hope that our paper will serve as a useful resource for anyone interested in learning more about monophonic hybrid analog/digital synthesizers.

ACKNOWLEDGEMENT

The authors wish to acknowledge the assistance and support of the University of Central Florida ECE Community for all the help and support to make this project possible.

MEET THE TEAM

Anna Wegener

Anna Wegener is a graduating Electrical Engineering student at the University of Central Florida. She is an ambient electronic music enthusiast leading her inspiration to build an analog synthesizer. Upon graduation, she will be pursuing a career in the commercial avionics industry at Collins Aerospace.

Magdalena Dobinda

Magdalena is a graduating Computer Engineering student at the University of Central Florida. Upon graduating she will be pursuing a career in microservices architecture and machine learning at BNY Mellon. She enjoys discovering and curating playlists of new and old music genres, inspiring her to pursue a project that pertains to this interest.

Sebastian Carrion

Sebastian Carrion is graduating with his degree in Electrical Engineering from the University of Central Florida he currently works with Qorvo as a Test Engineering Intern. After graduating he plans to continue his career at Qorvo working as a full-time Test Engineer. He hopes to continue a career path with a focus in RF & Communications.

Devin Vanzant

Devin Vanzant is graduating with a degree in Computer Engineering at the University of Central Florida. He is an avid programmer and ex-collegiate esports player competing for UCF. Previously having interned with the Department of Defense and Lockheed Martin, he will continue to work as a software engineer wherever his career may take him.

REFERENCES

[1] “Digital Synthesizer.” Wikipedia, Wikimedia Foundation, 16 Apr. 2023, Website Link

[2] “Sawtooth Wave.” *Wikipedia*, Wikimedia Foundation, 17 Feb. 2023, https://en.wikipedia.org/wiki/Sawtooth_wave.

[3] “How Subtractive Synthesizers Work.” *Apple Support*, https://support.apple.com/guide/logicpro/how-subtractive-synthesizers-work-lgsife41a22f/mac.

[4] Nugent, James. “What Is a Synthesizer, How It Works, Types and Uses.” *Higher Hz*, 24 Feb. 2021, https://higherhz.com/what-is-synthesizer/.

[5] *What Are the Differences between V/Oct and Hz/V of CV Output?* Website Link

[6] Kevin. “Attiny85 MIDI to CV.” *Kevin's Blog*, 12 July 2021, https://emalliab.wordpress.com/2019/03/02/attiny85-midi-to-cv/.

[7] DiCola, Tony. “FFT: Fun with Fourier Transforms.” *Adafruit Learning System*, https://learn.adafruit.com/fft-fun-with-fourier-transforms/background.