Hybrid Digital-Analog Polyphonic Musical Synthesizer

Alexander Barrass, Anthony De Rosa,
Jordan Khan, Robert Rosario, and Dina Suleiman

***Abstract*—This paper presents the design and testing process of the polyphonic hybrid electronic synthesizer: Neptune. It will encompass the research, design, and testing process over the past 6 months. In depth review of PCB design and software design are covered to explain operation. System components that operate major functions for synthesis will be covered in greater detail. The engineering specifications of NepTune are also explored to present practical and impressive functionality.**

***Index Terms*—Analog processing circuits, D/A conversion, Oscillators, Embedded Systems, Low-noise amplifiers, Low-pass filters, Music Synthesizers**

I. Introduction

The earliest forms of electronic music emerged in the early 20th century, as engineers harnessed vacuum tubes and early electronic oscillators [1] to create instruments capable of producing entirely new timbres. As technology developed, the advent of the transistor paved the way for commercially viable, portable synthesizers that define the modern instrument.

A synthesizer, by its simplest definition, is an electronic instrument that generates audio signals. Synthesizers are distinct from other electronic instruments, such as samplers, in that their sounds are actively generated via function generation rather than being played back from static audio recordings. This design space has since diverged into two primary philosophies: analog and digital. Analog synthesizers are prized for their component-level non-linearities and rich sonic character, while digital synthesizers offer stability, vast polyphony, and complex modulation capabilities.

The most foundational architecture for these instruments is subtractive synthesis, which is based on a signal path of discrete functional blocks, as seen in classic analog designs. This model typically begins with a Voltage Controlled Oscillator (VCO), a waveform generator that creates an initial, harmonically rich signal. This signal is then passed to a Voltage Controlled Filter (VCF), which sculpts the timbre by attenuating or removing specific frequencies. Finally, the signal amplitude is shaped over time by a Voltage Controlled Amplifier (VCA). These blocks are often modulated by other components, such as a Low Frequency Oscillator (LFO) for cyclical changes or an Envelope Generator (EG), which provides a time-varying control voltage, typically defined by ADSR (Attack, Decay, Sustain, Release) parameters, to shape a note's contour.

In this paper, we propose the design and implementation of NepTune, a hybrid synthesizer that leverages the core philosophies of both domains. Our approach seeks to merge the tactile immediacy and warmth of analog circuits with the flexibility and control afforded by a modern digital backbone.

II. Engineering Specifications

Much consideration went into the list of engineering specifications. It was necessary to balance performance with the capabilities and budget of the team. These specifications were chosen because of their practicality to be realized in the final product.

TABLE I: Engineering Specifications

Parameters	Values	Description
Max Dimensions	13” x 6”	Approximate final size. (L x W)
Weight	$<$ 8 lbs.	Weight of system.
Cost	$<$ \$500	The total cost of the synth to the end user.
Time Delay	$<$ 7ms	Response time of system from user input.
SNR	$>$ 40dB	The signal to noise ratio (SNR) should be within an acceptable range for entry-level hobbyists.
THD	$<$ 2%	The ratio of total undesired harmonics to the fundamental harmonic should be in an acceptable range.
Filter Roll-off	12dB/octave	Rate of roll off for configurable filters with key-tracking
Number of Octaves	8	Audible range of notes
Dynamic Range	60dB	The ratio between the loudest sound and the noise floor should be in an acceptable range.
Frequency Response	30Hz – 7kHz	The range of frequencies the synthesizer can play.
Audio Output Voltage	~0.35 V_{rms}	Voltage output for audio outputs in RMS voltage (V_{rms})
Min Headphone Impedance	16Ω	Minimum impedance headphone jack supported
Power Consumption	$<$ 25W	Synth uses 12V for components and Neptune wants to limit to 25W maximum draw.

NepTune is to be a small portable synthesizer capable of good performance. A 13”x6” form factor enables the synthesizer to have plenty of room for switches and knobs but also keeps the system small enough to move around. Additionally, a cost below $500 is important to keep this product inviting for beginners but also competitive with other products on the market. The time delay, signal-to-noise ratio (SNR), total harmonic distortion (THD), and dynamic range specifications are modest but practical given the resources and constraints of the project. The dual output of both line-level and headphone support are necessary to support private and public performance, depending on the environment the user plays in. Support for 16Ω with ˜0.35Vrms on the output is important to keep the synthesizer compatible with the majority of headphones on the market.

III. System Components

A. Digital Oscillator

Among NepTune’s many functions and features, the one function that sits at the base before everything else is the oscillator generation; the main wave that must be produced so that effects such as LFO and VCF can be applied. Oscillators inside NepTune are generated digitally by the DSP library, DaisySP, and embedded within our software. It can generate many classic waveforms such as sine, square, triangle, sawtooth, and ramp (inverse sawtooth) as fast, real-time approximations using simple implementations of the mathematical functions as shown in Table II [5].

While NepTune’s oscillators are designed to generate standard waveforms with high fidelity in the digital domain, it is a naive approach to waveform generation (e.g., computing a perfect square wave by instantaneously flipping between high and low values), which leads to severe aliasing artifacts at higher frequencies. One classical solution is to bandlimit the oscillator by constructing the waveform from a Fourier series truncated to Nyquist which can produce an alias-free result [6]. NepTune’s oscillators, powered by the DaisySP library, use PolyBLEP variants (e.g., bandlimited square (WAVE_POLYBLEP_SQUARE), saw (WAVE_POLYBLEP_SAW), and triangle (WAVE_POLYPLEB_TRI)) to preserve waveform clarity at high pitches. Sine waves, being naturally bandlimited, do not require this correction [7]

B. Voltage Controlled Filter

One of the most important components that enhances the music design experience in a synthesizer is the voltage-controlled filter (VCF). A VCF behaves similarly to a traditional active filter; however, the key distinction lies in the fact that its cutoff frequency and resonance characteristics can be dynamically adjusted in real time by a control voltage. This enables musicians and sound designers to shape timbre and tone interactively, producing expressive and evolving sounds. The general topology of the filter—whether it operates as a low-pass, high-pass, or band-pass filter—is determined by the design engineer during circuit development.

In the case of Neptune, the chosen design was a four-pole cascaded low-pass filter with Q-compensation, providing a sharp roll-off and a smooth yet musical resonance peak. A low-pass filter operates by allowing all frequency components below a predetermined cutoff frequency to pass while attenuating higher frequencies. Typically, the cutoff frequency of a simple RC low-pass filter is given by:

$$f_c = \frac{1}{2 * pi * R}$$

However, in a VCF IC the filters cut-off frequency is determined by the following equation:

$$f_c = \frac{200G}{2 * pi * R}$$

When these equations are compared, it is apparent that the only modification is the multiplication by the factor 200G.In this context, 200G represents the transconductance (g_m) of each voltage-controlled amplifier (VCA) cell when the control voltage equals 0 V. Transconductance describes how the output current of the VCA varies in response to the applied input voltage.

TABLE II: Oscillation Equations

Waveform	Mathematical Function
WAVE_SIN	Internally uses arm_sin_f32() or a lookup table (LUT) with interpolation for fast sine computation.
WAVE_SQUARE	Simply checks the oscillator phase (a float between 0.0 and 1.0). If it’s less than 0.5, it outputs +1.0; otherwise, it outputs -1.0. No smoothing, no harmonics summing, just a hard switch.
WAVE_TRI	It uses a simple linear ramp-up and down: Rising from -1 to +1 over the first half of the phase. Falling from +1 to -1 over the second half. No harmonic shaping or filtering.
WAVE_SAW	A direct linear ramp from -1.0 to +1.0 over each oscillator cycle. Resets to -1.0 every 2*pi radians or phase = 1.0.
WAVE_RAMP (reverse sawtooth)	Ramps downward from +1 to -1 over the same phase range.

The control voltage that defines the cutoff frequency operates by modulating the exponential current generator within the IC. This generator produces a current that determines the effective transconductance according to the relationship:

$$g_m \propto e^{\frac{1}{2\pi R}}$$

As the control voltage changes, the transconductance correspondingly varies, causing the cutoff frequency to shift exponentially. Each change of approximately –18 mV results in either a doubling or halving of the transconductance, effectively altering the filter's cutoff frequency and, consequently, the tonal character of the sound. This direct voltage control over filter behavior allows for highly dynamic modulation effects, such as sweeps and envelopes that respond naturally to user input.

Another key aspect influenced by the VCF is resonance, which defines how strongly frequencies near the cutoff are emphasized. Resonance occurs when a portion of the filter's output signal is fed back into its input at or near the cutoff frequency. This feedback reinforces those frequencies, creating a pronounced peak in the filter's frequency response immediately before the roll-off begins.

This phenomenon modifies the tonal quality of the sound, adding harmonic richness or sharpness depending on the resonance level. In Neptune's design, resonance control is inherently available since the fourth operational amplifier in the VCF configuration is connected to a voltage-controlled amplifier within the IC. The amount of feedback—and therefore the resonance intensity—is further governed by an external control voltage applied through the Q-control pin. This voltage determines the degree of amplification in the feedback loop, enabling the user to shape how resonant or "bright" the filter response becomes.

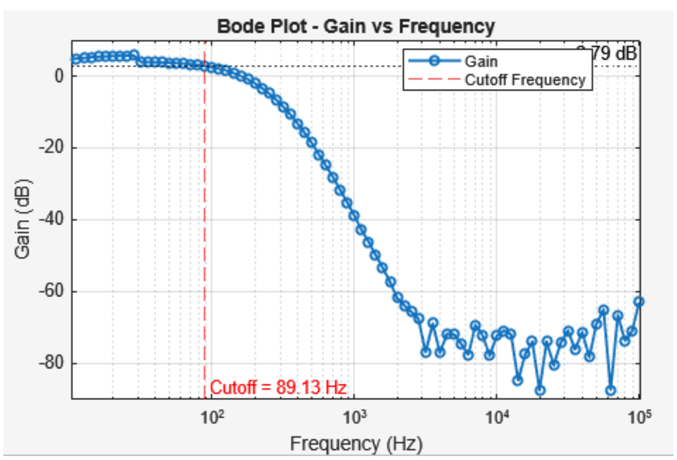

Figure 1: Frequency response of the 4- pole filter with a cutoff voltage of 5V.

As displayed in Figure 1 and Figure 2, the cutoff frequency control voltage that was chosen for Neptune was 5V, and the configuration of the capacitors and the resistors yielded a cutoff frequency of as low as 100Hz to 3KHz.

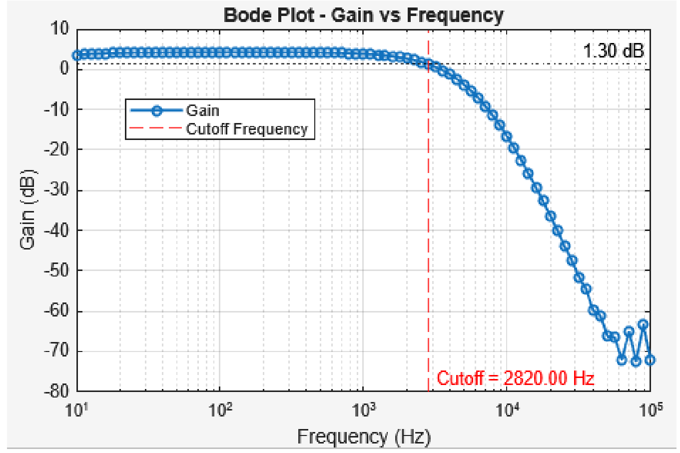

Figure 2:Frequency response of the filter with a cutoff voltage of 0V.

At 0 V, the control voltage produces a relatively low transconductance, resulting in a cutoff frequency near 100 Hz. In this state, the filter allows only bass and low-midrange frequencies to pass while effectively attenuating higher frequencies. The response is smooth and rounded, ideal for shaping warmer, darker tones or removing high-frequency harmonics from complex waveforms such as sawtooth or square waves.

At 5 V, however, the increased control voltage dramatically boosts transconductance, shifting the cutoff frequency upward to approximately 3 kHz. The filter now passes most of the audible spectrum except for the upper harmonics, resulting in a much brighter, more open tonal character. The higher cutoff also shifts the resonant peak upward, making the resonance more pronounced at higher frequencies, which can add clarity, bite, or even percussive sharpness to the sound depending on the Q-setting.

This voltage-controlled range from 100 Hz to 3 kHz demonstrates how the VCF transforms static filtering into a dynamic sound-design tool. By modulating the control voltage with an envelope generator, LFO, or keyboard tracking, the synthesizer can produce sweeping filter effects, evolving textures, and expressive timbral variation — key characteristics of analog and hybrid synthesis

IV. Low Frequency Oscillator

The LFO is used to modulate the cutoff frequency of the filters (VCF). This is a staple feature in modern synthesizers and must be implemented in NepTune. The LFO in NepTune will be supplied by Electric Druid's VCLFO10. This is an 8-bit PIC microcontroller flashed with software to function as a digital oscillator. The VCLFO10 has a 10-bit output resolution featuring 16 different waveforms and several interesting features such as the sample-and-hold (S&H) function. When the S&H function is enabled, it will sample the selected waveform at regular intervals and hold the value, producing a stepped output. Additionally, the VCLFO10 comes with open-source software, which is useful to tailor it to the needs of the project.

The team decided to use a microcontroller-based solution to realize the LFO because of the extensive hardware already included on NepTune. Having a digital solution reduces complexity and speeds up development time.

The LFO achieves oscillation through Direct Digital Synthesis (DDS). DDS is a completely digital method used to produce analog waveforms. Waveforms are digitized and stored in memory at very high resolutions. A waveform is generated by stepping through this stored data at various increments. Smaller step sizes will yield a slower output frequency, while larger step sizes will yield faster ones [2].

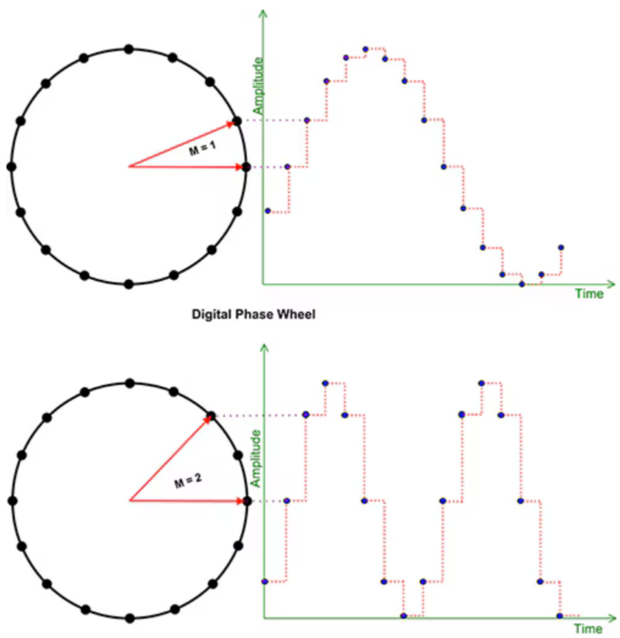

Figure 3: Illustration of a phase accumulator. Smaller step sizes yield a slower frequency, while larger step sizes yield a higher frequency. Courtesy of Digikey.

The system responsible for stepping through these waveforms is called the phase accumulator. The phase accumulator is essentially a modulo-M counter, which increases its stored number with every clock signal. The step size of each count can be increased by increasing the binary-coded input word (M). As M increases in size, the step size increases and hastens the output frequency. The relationship between the clock frequency and the output frequency of the system is given by:

$$\mathrm{f_{out}} = \frac{M * f_c}{2^N}$$

where fout is the output frequency of the signal, M is the binary input word, fc is the internal clock frequency, and n is the bit-length of the phase accumulator. For a larger bit-length, a greater range of output frequencies is possible. [3]

Traditionally, this output would be fed into a digital-to-analog converter (DAC), but since the VCLFO10 is a PIC microcontroller with only digital GPIO pins, the output is a PWM signal which must be filtered to uncover the desired low-frequency wave.

A. Voltage Controlled Amplifier and Mixer

The VCA chosen for this design was Sound Semiconductor's ssi2164. This chip has 4 available inputs meaning that it would be able to handle the 4 voices that are being considered for NepTune. With multiple configurable classes of operation, A, B, and AB, the chip is specifically designed for use in audio devices carrying low distortion and noise [4]. This is where the output of the VCF is taken and modulated by the output of the envelope generator. The output then runs into the ssi2190, another chip from Sound Semiconductors, acting as the audio mixer, which combines all of the voices into a single signal for output amplification.

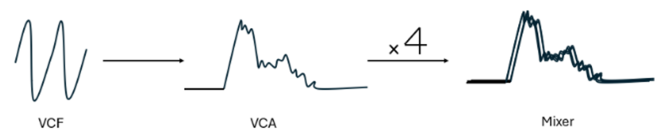

Figure 4: Outputs of the VCF, VCA, and Mixer to demonstrate functionality.

The mixed audio signal can consist of countless variations of each waveform as voices 1 and 2 can be sine waves at different frequencies and voice 3 can be ramp, voice 4 square and so on. The combined sound will be a unique note with time varying amplitude adding depth to the audio. The gain ranges of the VCA reside between –20dB and +100dB. The distortion of both the VCA and mixer is rated at 0.025%, which greatly preserves the sound quality that is required for basic operation [5].

Because the two chips were both created by a company that focuses not only on creating chips that integrate with one another easily but also for the sake of audio, meaning that these chips were easily chosen for the best available performance.

B. Envelope Generator

The envelope generator generates time-varying control voltage which feeds directly into the control port of the VCA which shapes the amplitude of the sound output by the VCF. Like the VCLFO10 described earlier, Electric Druid's ENVGEN8 is a PIC controller which has had code downloaded to operate as an envelope generator. It offers a 10-bit DAC output and 7 customizable control voltages to change the behavior of the chip output [6]. What makes this envelope generator into what is classified as ADSR, is the four adjustable control voltages: Attack, Decay, Sustain, and Release.

Each of these control voltages are operated via potentiometers available on the control board for the user to configure. Attack determines the time it takes for the amplitude to rise from 0V to the peak of the signal at 5V. Decay is the time that the peak amplitude then takes to fall down to the Sustain level. Sustain is the only value that determines voltage level rather than a time-characteristic of the envelope and can determine the voltage that the output remains at until reaching the next section: Release. Release will determine the rate at which the values decay from the sustain voltage level back down to 0V. These 4 values in combination will create a waveform resembling the following.

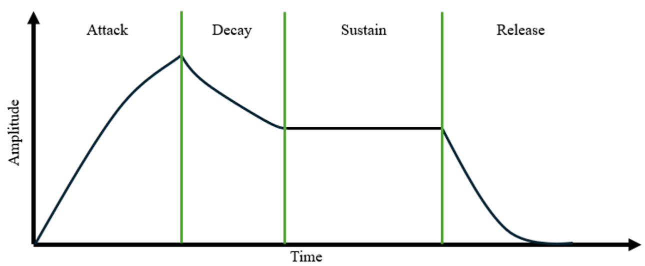

Figure 5: Magnitude representation of the output of the ADSR envelope generator with specific sections separated for clarity.

There are numerous other customizations available for the chip, such as the designer can decide whether or not a small “punch” in sound is added between the Attack and Decay sections. The Time control voltage determines the rate at which sections A, D, and R complete in (x1 - x0.01). Additionally, the chip offers options for linear / exponential control which the team vied for a more natural sounding exponential, and the mode of the ADSR can operate in a normal, gated looping, or LFO behavior.

The mathematical breakdown of how these values operate the amplitude of the wave generated is digital meaning that regular discrete component tuning is not necessary. For Attack, the following equation shows how the voltage rises over time until its peak, with α being the rate the user selects:

$$\text{env}_{\text{next}} = \text{env}_{\text{current}} + (\text{Goal} - \text{env}_{\text{current}}) \cdot \alpha$$

And the decay behavior is expressed with exponential decay as follows:

$$V(t) = V_{\text{Sustain}} + (V_{\text{Peak}} - V_{\text{Sustain}}) \cdot e^{-t/\tau}$$

Sustain simply keeps the voltage constant at the specified value and Release follows the final equation:

$$V(t) = V_{\text{release}} \cdot e^{-t/\tau}$$

Once the envelope is generated, it is directly fed into the control port of each voice input on the VCA, and each voice will require one ENVGEN8 chip. The envelope will modulate the sound signal coming from the VCF to match its shape shown in the previous figure. This creates a massive reservoir of sounds that the user can achieve with any specific set of values for ADSR. Specific to our configuration, the output of the ADSR was changed from 0-5V to –660mV – 0V to interact with the VCA correctly. This was achieved by reprogramming the chip.

One more bug that was required for us to fix was the creator of the chip seems to have only programmed the chip to respond to changes on its control voltage lines meaning that upon startup, the grounded control voltage for mode, one that we wanted to remain at 0V anyways would not actually ever activate due to no voltage change. A workaround was implemented by having the mode pin connected to a voltage divider that would set the mode within the range we still wanted (0-1.25V) meaning powering up the device would shift the voltage from 0V to 1V, changing and activating the chip without changing the behavior we expected.

C. Audio Output

The output audio amplifier features two LM386 audio drivers. These low voltage audio power amplifiers are capable of driving loads as low as 4Ω. These specific drivers were chosen to establish compatibility among as many headphones as possible. This encourages users to listen through any type of headphones they might have, enabling an easier user experience. These drivers are used to handle both the headphone output and the line-level outputs.

D. PCB Design

The physical layout and grounding scheme of a mixed-signal synthesizer are paramount for designing a high fidelity, low noise system. To address these challenges, the circuit board design of our synthesizer employs a two-board, split-grounding plane solution. The sensitive analog audio processing components have been physically separated from the high speed and noisy digital control circuitry, connected via insulated wiring for audio and control voltage signals.

The digital board houses components such as the ESP32 microcontroller used for handling control signals, and an STM32 based development board used for handling audio signal generation. Both boards generate high frequency switching noise that is isolated on a dedicated digital ground plane; however, the digital board itself contains mixed-signal elements. The audio codec used for D/A conversion and the digital potentiometers that output the analog control voltages are located on a separate analog grounding plane. The split digital/analog ground planes are joined at a single-point connection set between the two planes. This ensures that the transient digital currents from the codec and potentiometers do not contaminate the analog reference for their respective outputs. [7]

The analog board is designed purely in the analog domain. Offloading all high frequency digital signals to a separate PCB keeps the analog board’s ground plane quiet and provides a reference untainted by digital noise for all sensitive audio signals and voltage-controlled components. Both boards have been designed with a 4-layer stack-up (SIG/GND/PWR/SIG) to optimize signal integrity and power delivery. The layered approach, combined with the physical separation of digital and analog processing, are the primary effective strategies utilized for noise mitigation and signal purity in the NepTune synthesizer.

E. Controls

The user experiences the board through a series of potentiometers which are physically present as knobs for the user to tune. There are small differences between some of the knobs as the values available have different intervals of operation to be practical. For instance, the waveform selection of each voice and the waveform selection for the LFO have key distinct values that the user can switch between. With selections between sine, square, sawtooth, triangle, and more, because they are distinct, the knob that the team decided to include was a rotary switch that shifts between each selection to sufficiently provide control to operator.

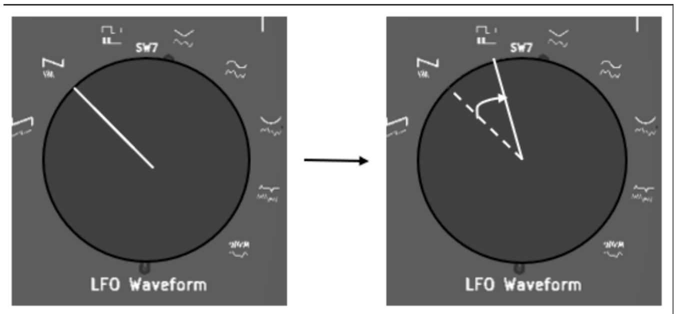

Figure 6: Example showing rotary switch changing distinct values by shifting from a set angle to the next waveform.

With detailed silkscreen descriptions of each interactable on the control board, the user can clearly observe the value that they are changing at any moment. Additionally, there have been graphics added to isolate different groups of values that have correlations to one another. This shows itself in groups like the ADSR which has four distinct values that all interact with one another, leaving the exclusion of these knobs from other operations intuitive for use.

A significant design choice that team made was to ensure a portable design as often; market parallels offer maneuverability with the instrument as well. To satisfy this design, the team decided to restrict the size of the board to 13" x 6" which will generally bring the size of the device to similar specifications. This will allow for transport of the device using everyday bags and containers. Smaller designs will also increase the ergonomics of operation, creating a more pleasant user experience. Often instruments like electronic synthesizers are required to be moved from studio to stage and so on. A board that would be overweight or cumbersome would degrade the quality of NepTune.

F. Communication Protocols

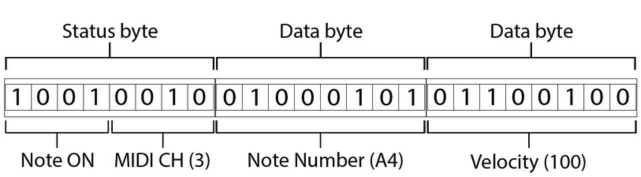

Figure 7: Bit layout of a MIDI message.

The first of the communication protocols used is MIDI. The NepTune synthesizer receives musical note data via the standard MIDI protocol using a 5-pin DIN connector. MIDI is a serial protocol running over UART at a baud rate of 31,250 bps [8]. As seen in figure 7, each MIDI message includes a status byte (high bit = 1) followed by one or two data bytes (high bit = 0), depending on the message type. The STM32's UART is configured to this MIDI-standard rate with 8 data bits, no parity, and one stop bit (8-N-1) to match protocol requirements [8]. We use the Daisy library's MIDI handler to set up USART1 with the correct configuration and parse incoming bytes into events, allowing NepTune to respond to MIDI messages in real time [9]. The MIDI interface on NepTune uses a dedicated UART channel with the MIDI-standard baud rate and an opto-isolated input, allowing an external MIDI controller or sequencer to play the synth with minimal latency and standard compliance [9][10].

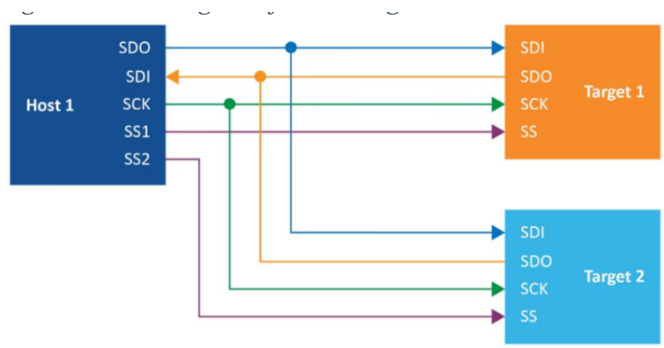

Figure 8: Basic diagram of a dual target SPI network.

NepTune's STM32 on the Daisy Seed communicates with various peripherals using SPI, SAI, and UART. SPI (Serial Peripheral Interface) is used to control digital potentiometers (digipots) that adjust analog parameters like filter cutoff and master volume. The STM32 sends numeric values over SPI to set the wiper position of each digipot, enabling real-time analog modulation while maintaining software-based control and preset recall [11]. As shown in Figure 8, each digipot is addressed via a dedicated chip-select line, and the SPI bus (configured as a master interface) operates at high speeds (often ¿10 MHz), ensuring fast, simultaneous updates to multiple analog controls with no audible lag [12]. For audio output, NepTune uses the STM32's SAI (Serial Audio Interface) configured in I²S mode to stream stereo audio at 48 kHz, 24-bit resolution to the onboard PCM3060 codec. This codec runs in hardware-controlled mode as an I²S slave, while the STM32 generates the necessary clock signals (MCLK, BCLK, LRCK) [13]. The firmware uses a double-buffered DMA scheme to continuously stream audio without CPU overhead. As one buffer is played, the next is computed, ensuring smooth, low-latency audio performance. No I²C configuration is needed for the codec, simplifying initialization and boosting reliability [13]. Meanwhile, UART is used for high-speed communication between the ESP32 module and the DaisySeed. The ESP32 acts as a secondary processor and sends parameter data to the STM32 over a dedicated UART link at 115200 baud, allowing the DaisySeed to focus exclusively on real-time audio tasks.

Unlike SPI, which is synchronous, UART allows asynchronous byte-wise transmission which is ideal for simple command and control without additional wiring complexity. This separation of responsibilities improves responsiveness and reliability while maintaining a clean communication channel between the two processors [12].

V. Performance

In previous tests, the synthesizer has shown evidence of meeting some of the engineering specifications. These specifications include the time delay, THD, and output voltage.

To calculate the time delay, the phase of the input signal was compared with the output signal. The time delay between the two signals was used to estimate the time delay of the system. When tested, NepTune yielded a time delay of about 1.25ms, surpassing the 7ms specification. A screen capture of this test is shown in Figure 9.

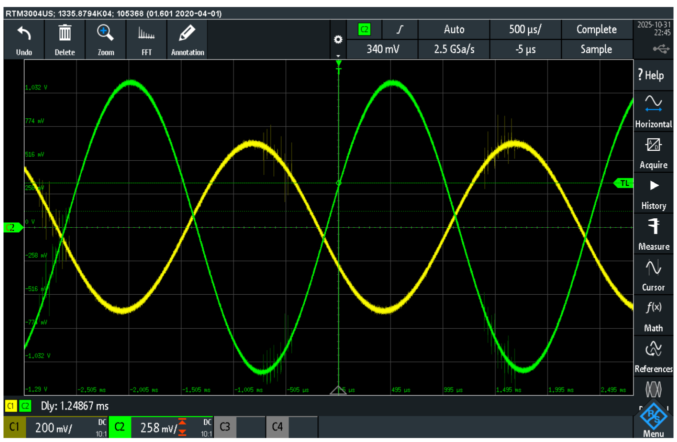

Figure 9: Screen capture of oscilloscope measuring a time delay of 1.24ms. Channel 1 (yellow) is input. Channel 2 (green) is the output.

Next, NepTune's distortion performance must be evaluated. THD can be calculated using the equation:

$$\text{THD} = \frac{\sqrt{V_1^2 + V_2^2 + \cdots}}{V_F}$$

The THD performance of NepTune was measured by sending a 440Hz sine wave at 1Vpp into the system and observing the output with a spectrum analyzer. Surprisingly, Neptune had hardly any additional harmonics. This results in a THD $<<$ 2%, meeting the THD requirement. A screen capture of this measurement can be observed in Figure 10.

However, there is still room for improvement, as the SNR is less than satisfactory. SNR was measured by observing the output of the system with a signal passing through, and without any signal passing through. The signal passed through during the test was a 440Hz sine wave at 1Vpp. The resulting SNR was 38dB, which is slightly less than the requirement of 40dB. Presently, the team is still working on a way to improve NepTune's SNR performance.

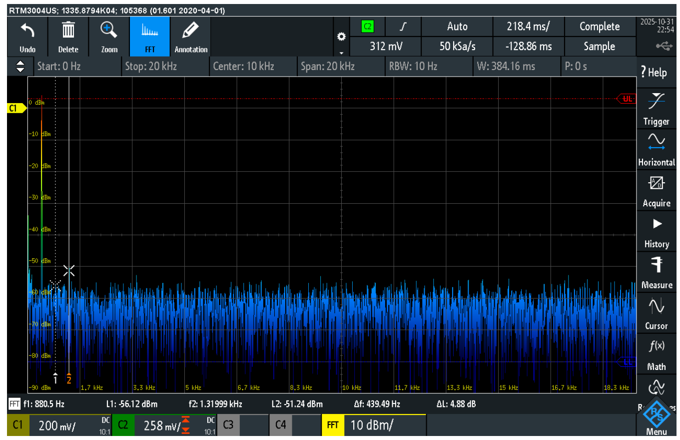

Figure 10: Frequency spectrum when sine wave is passed through the synthesizer.

VI. Future Opportunity

There are countless avenues for the team to approach a newer, better funded and longer designed NepTune if the desire/requirement appears. Due to the creative nature of designing a tool that for most intents and purposes, synthesizes music, not only is the refinement of operation possible but the expansion of configurability by the user.

There could be improvement by tuning and expanding the variety of audio chips present, as much of the semester has been spent adjusting values and integrating them for this purpose. And in the interest of deepening the toolbox of sounds that NepTune can generate, expanding the number of voices handled along with the numerous well documented techniques recorded on how to create unique sounds with techniques like the already present ADSR, LFO, and more.

In a different direction, one interesting inclusion that the team was hoping to stretch too but was not afforded the time was a digital oscilloscope chip and system to observe the waveform in real time. Because every value physically affects the wave, and that effect can be observed in the waveform of the signal, being able to show the waveform as it is at any time the user is playing, tuning, or performing would greatly improve the intrigue of NepTune's design.

VII. Conclusion

It is important to note that our team is very passionate about music synthesis and to finally get the chance to design and create an instrument that can produce music is extremely endearing. All preliminary research and work dedicated to creating a hybrid, polyphonic electronic synthesizer has been summarized in these 8 pages. There are numerous dead ends and issues that occurred during the process of creating NepTune. But through rigorous testing, designing, and fixing, the final product is coming together into a well–rounded, fully functional instrument interface. Our team was driven to include as many user options as possible to design in time for NepTune's deadline so that our design could result in the most user-controlled capability possible to expand the user to a world of opportunities with sound.

VIII. Acknowledgement

We wish to acknowledge the assistance and support of Dr. Arthur Weeks, Dr. Chung Yong Chan, and Professor Mark Maddox for their help and mentorship throughout the project. We would also like to thank Dr. Saleem Sahawneh, Dr. Sonali Das, Dr. Kalpathy Sundaram, and Dr. Justin Phelps for serving on our review committee and supporting our project.

IX. References

[1] J. Kovarsky, "History of the Synthesizer, Part 1," Yamaha Music, Apr. 04, 2022.

[2] A. Pini, "The Basics of Direct Digital Synthesizers (DDSs) and How to Select and Use Them," DigiKey, Mar. 20, 2019.

[3] "All About Direct Digital Synthesis — Analog Devices," Analog.com, 2024.

[4] "Electric Druid ENVGEN8 Voltage-Controlled Envelope Generator Datasheet, Rev.D." Electric Druid.

[5] "SSI2164 FATKEYSTM QUAD VOLTAGE CONTROLLED AMPLIFIER." SOUND SEMICONDUCTOR, Feb. 2023.

[6] "SSI2190 PROCIRCUITTM 6-INTO-1 VOLTAGE CONTROLLED MIXER*." SOUND SEMICONDUCTOR, Dec. 2024.

[7] The Sierra Team, "Reducing Parasitic Capacitance in PCB Layout — Sierra Circuits," Sierra Circuits, Feb. 23, 2021.

[8] ElectroSmith, "DaisySP Documentation – Oscillator Class," GitHub, Accessed Oct. 30, 2025.

[9] J. S. Abel and J. O. Smith, "Restoring a clipped signal," in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Toronto, ON, Canada, May 1991, pp. 1745–1748.

[10] M. P. Heins, "Bandlimited Oscillator Models using PolyBLEP," Michael Grigsby Blog, Accessed Oct. 30, 2025.

[11] J. Campbell, "MIDI Basics: Protocol & Hardware," Jack Campbell Sounds, Accessed Oct. 30, 2025.

[12] ElectroSmith, "MIDI Implementation with Daisy," GitHub Wiki, Accessed Oct. 30, 2025.

[13] N-Audio, "How MIDI Input Works: Optical Isolation & Circuit Design," N-Audio.net, Accessed Oct. 30, 2025.

[14] Digi-Key, "Using Digital Potentiometers in Audio Circuits," DigiKey, Accessed Oct. 30, 2025.

[15] ESP Boards, "SPI Communication Between STM32 and ESP32," ESPBoards.dev, Accessed Oct. 30, 2025.

[16] ElectroSmith, "Daisy Seed Hardware Manual (Rev 7)," Daisy Documentation, Accessed Oct. 30, 2025.

[17] "SSI2140 FATKEYS TM VOLTAGE CONTROLLED MULTI-MODE FILTER." Accessed: Nov. 01, 2025.