SONOMA STATE UNIVERSITY | ENGINEERING SCIENCE

**Senior Design Project Progress Report**
**EE 493 Senior Design Project**

# **C**ustomizable **H**ardware-based **O**pen-source **R**eal-time **D**igital **S**ynthesizer

# C.H.O.R.D.S.


By:

Bjorn Lavik
Julius Faller
Madison McIntyre


**May 2024**


Faculty Advisor: Dr. Farid Farahmand, SSU
Industry Advisor: Mr. Spencer Scott, SSU
Client: Mr. Cameron Bartoloma


Project Website: https://chordsynthesizer.com

## Acknowledgments

*Thank you to*
*our Faculty Advisor, Dr. Farid Farmand,*
*our Industry Advisor, Spencer Scott,*
*and Lead Industry Advisor, Dr. Chris Stewart,*
*for all their help throughout the process of this project.*

## Abstract

The Customizable Hardware-based Open-source Real-time Digital Synthesizer (CHORDS) project aims to address the lack of affordable modular synthesizers with true polyphonic capabilities. Traditional modular synthesizers are constrained to monophonic or paraphonic sound production, often requiring costly and complex workarounds to simulate polyphony. This project introduces an innovative system based on microcontrollers to deliver real-time, 8-channel polyphony using a newly developed Digital Modular Synth Protocol (DMSP). With a focus on cost-effectiveness, portability, and modularity, the synthesizer integrates MIDI input and features like oscillators, filters, and envelopes. Each module is powered by a dedicated microcontroller and synchronized through a global clock, ensuring low-latency performance. This design offers musicians enhanced creative freedom while maintaining the tactile experience of hardware synthesizers, bridging the gap between high-cost analog solutions and purely virtual instruments.

## Table of Contents

## List of Figures



## List of Tables

# 1. Introduction

Musicians and sound designers who use modular hardware synthesizers are often limited to monophonic sound production unless they invest in costly polyphonic systems, which can range from hundreds to thousands of dollars [1]. For many users, this expense is prohibitive—and even when affordable, most polyphonic synthesizers are pre-built, non-modular machines. Modular synthesizers, particularly those following the Eurorack standard, offer unparalleled flexibility in sound design, but they do not support true polyphony natively [2].

Current workarounds to simulate polyphony—such as using multiple oscillators, arpeggiators, or delay lines—are often complex, unreliable, and only partially effective. These techniques focus on either voice generation or note triggering, not both, and tend to suffer from poor timing accuracy and voice allocation conflicts [3]. As a result, users must compromise between flexibility and usability, making it difficult to achieve rich, multi-voiced arrangements within modular environments.

This project addresses that limitation by introducing true polyphonic capability into modular synthesizers through a cost-effective, microcontroller-based system. Polyphony, the simultaneous generation of multiple distinct audio frequencies, is typically constrained in modular systems to monophonic or paraphonic operation—where only one frequency can change dynamically at a time, regardless of how many oscillators are present.

The system is built around the RP2350 microcontroller, chosen for its cost efficiency and sufficient processing capability for real-time digital signal processing tasks. The RP2350 features a 32-bit Arm Cortex-M3 core operating at 72 MHz, with 128 KB of flash memory and 20 KB of SRAM, providing an adequate balance between performance and memory for embedded audio applications [4]. Although it offers fewer computational resources than high-end DSPs, the modular nature of our design compensates for this limitation by assigning a dedicated microcontroller to each module. This distributed architecture allows the system to maintain low-latency, parallel audio processing across multiple voices without overloading a single processor.

The system utilizes MIDI (Musical Instrument Digital Interface) as the primary control protocol. MIDI is a well-established, digitally encoded communication standard designed in the 1980s to interconnect musical devices and has been widely adopted due to its low-bandwidth, serial design and real-time performance [5]. MIDI messages are interpreted by the microcontroller and routed to various synthesis modules including oscillators, envelopes, and filters. Additional modules may be incorporated as development progresses.

Processed digital signals are converted to analog via an I2S (Inter-IC Sound) interface, which allows high-fidelity audio transfer with low jitter and supports stereo or multi-channel output formats [6]. This digital-to-analog pathway is essential for recording and interfacing with other analog equipment in music production environments.

To facilitate efficient communication among multiple modules, we developed the Digital Modular Synth Protocol (DMSP). DMSP uses a global clock signal to maintain synchronization between control data and audio channels. Each frame starts with a synchronization bit, followed

by 3 bits used to address the 8 channels. The remaining bits are allocated for MIDI and audio data payloads. By assigning each module its own microcontroller, memory and processing resources are distributed across the system, allowing for parallel execution and eliminating typical limitations related to RAM or storage bottlenecks in embedded systems.

Ultimately, our project aims to provide an affordable, modular, and scalable polyphonic synthesizer platform. With this digital communication system, musicians gain access to polyphonic synthesis with the flexibility of modular control—all at a fraction of the cost of commercial solutions.

## 2. Literature Review and Previous Works

Currently there are no simple alternatives to modular polyphonic synthesis. A variety of alternatives have been developed, such as virtual instruments, multi- chord oscillators, and sequence-passed polyphony, like arpeggios and delay. These alternatives emulate the idea of polyphony, while sacrificing price, hardware, and creativity.

Virtual instruments can provide the polyphonic control, yet are limited to software application. For instance, the VCV Rack 2 Virtual Eurorack [7] is a software synth that has the capabilities of a eurorack module while providing the ability of polyphonic output. Although these are excellent features, being chained behind a computer diminishes the immersion provided with a hardware modular synth. having the ability to control knobs and move faders to control parameters allows the musician a tactile experience, and can bring them close to their music. These virtual synths are typically the cheapest options out there, since they are only software. Our product intends to target the musician who is looking for more out of their digital synthesizers, without having to buy expensive analog gear.

Analog modular synths have many workarounds to fix the issue of polyphony. The main one is using a paraphonic or multi oscillator system. This is done with either a specific module or multiple single oscillators patched together. This can be done on a small scale like the Behringer Proton [8]. This synth utilizes two oscillators to create harmonics and chords. This solution limits the control of the harmonics and leaves the musician with the same amount of options and a monophonic synth. For musicians creating their own Eurorack they can use multi chord oscillators such as the a-111-D [9]. This module allows up to four oscillations to create even larger harmonies. These modules can help generate beautiful sounds, however they are limited in versatility. With this project we intend to have true polyphony with our synthesizer, meaning the musician can play up to 8 notes on a MIDI keyboard and change them at their will. This allows for more creative freedom, while utilizing the beauty of modular synthesizers.

Lastly, there are countless modules that can sequence notes or delay harmonics to create the illusion of polyphony. A module like the Black Module from Erica Synths [10] allows the musician to sequence multiple oscillators to create harmonics and patterns over one another. For synth hobbyists who may not know how to play the piano, this allows them to lay out melodies and musical ideas. This varies from our product as we intend to allow for MIDI control, which can allow the musician to play the piano or any other compatible instrument. Other musical techniques and FXs can be used to create harmonies, such as arpeggios, delay effects, or ring

modules. Relying on playing single harmonies near one another, the sense of polyphony can be created. Although these techniques are used they are note limited to one type of instrument and can be implemented even in our synthesizer.

In our research, One synthesizer has come close to what we intend to achieve. The PO modular 400 from Teenage Engineering [11] is a low budget modular synth that allows for creative control for the musician. For around $300 this synth provides versatility and power with various modules such as oscillators, filters, envelopes, and a sequencer. What this synth does not provide is polyphonic capabilities or reprogrammability of its modules. Although it has a sequencer and the ability to be externally triggered it will not generate more then one frequency at once. Our goal is to build a synth with similar capabilities. Instead of using a sequencer or other paraphonic modules, we intend to have true polyphony.

With our research we have found many alternatives to, or approximations of, polyphony when in modular synthesis. Many of these options require more cost with the addition of modules for sequencing or generating multiple oscillators. This can be intimidating for a musician looking to get into hardware synthesizers. Other cheaper others forgo the hardware entirely and keep everything virtual. We intend to meet in the middle of these areas and provide the hardware experience for the amateur musician looking to get away from their computer.

## 3. Methodology

Our team intends to solve the problem of low cost polyphony by using a microcontroller based system. When creating a digital synthesizer, we are able to utilize more complicated techniques within synthesis all at a low cost. This system will allow for 8 frequency channels to be sent through the microcontroller using our Digital Modular Synth Protocol (DMSP). Outlined in this proposal is our DMSP which will utilize a global clock to synchronize the incoming data in real time for fast feedback to the musician. This system starts with the synchronization bit followed by 3 bits for controlling the 8 frequency channels. The remaining bits will be used for additional MIDI and audio data. This is the basis for our project and is intended to help control the synthesizer. This approach will create a low cost polyphonic synth that allows for 8 frequency channels to be sent through the system, and achieve our intended goal for the project.

## 4. Project Requirements

### Marketing Requirements (MR)

1. CHORDS is a hardware synthesizer, utilizing a new digital communication protocol
2. CHORDS will have polyphonic capabilities, with a minimum of 4 voice channels
3. CHORDS and the DMSP will be compatible with MIDI commands for versatile user control
4. CHORDS is modular, and comes equipped with Lookup Table Oscillator, Biquad Filter, ADSR, Line-Out, and MIDI in modules
    a. CHORDS oscillator module is a Lookup Table Oscillator
    b. The CHORDS Line-Out module will utilize a ¼” audio jack to send a Line-Out signal at to the user's audio interface or speaker system
    c. The CHORDS Biquad Filter module will have a control knobs for the musician to

change cutoff frequency, Q, and morph between low and high pass
   d. The CHORDS ADSR modules will allow the musician to set Attack, Decay, Sustain, and Release time with the control knobs
5. CHORDS will have a durable custom case
6. CHORDS is a portable system that operates on battery power.
7. CHORDS will have low latency, ensuring that each musical note or control input is reflected quickly for a responsive playing experience.
8. CHORDS provides a more versatile musical experience at an affordable price

**Engineering Requirements (ER)**

ER-1. CHORDS modules will have knobs, switches, and buttons, to control parameters (MR
ER-2. CHORDS will be Polyphonic, allowing for a minimum of 4 voice channels to be played at one time. DMSP has 3 channel bits, which supports a maximum of 8 voice channels (MR 2)
ER-3. Each CHORDS Module shall introduce a maximum latency of no more than 5 milliseconds and system latency shall scale linearly with the number of modules in the signal chain (MR 7)
ER-4. CHORDS will be controllable with a standard MIDI controller. Using the IC (6N138) MIDI information will be converted into DMSP signals containing pitch and gate data (MR 3)
ER-5. CHORDS oscillator module will be tuned to 440 Hz for the A4 note on MIDI keyboard (MR 4a)
ER-6. CHORDS Inter-Integrated Circuit Sound (I2S) module will run at a sample rate of 40 kHz at 24-bits and Total Harmonic Distortion (THD) shall be no higher than 1%, ensuring high audio fidelity with minimal distortion.(MR 4b)
ER-7. CHORDS Biquad Filter will operate from 15 Hz-20 kHz (MR 4c)
ER-8. CHORDS case will be 3d printed in PETG (MR 5)
ER-9. CHORDS will utilize a 5 V supply powered by a rechargeable battery for portability and shall last at least 8 hours on one charge(MR 6)
ER-10. Utilizing the RP 2350 Microcontroller, CHORDS is estimated to cost $200 or less (MR 8)

## 5. Implementation

The implementation of CHORDS revolves around the integration of modular hardware and a newly designed Digital Modular Synth Protocol (DMSP). The core of the system is a global clock operating at 5.28 MHz and sync bus operating at 160 kHz, which ensures precise synchronization across all modules. Each module—such as oscillators, filters, and envelopes—is powered by an individual microcontroller, enabling parallel processing and minimizing latency. The system begins with MIDI input, which is processed to generate corresponding pitch data. This data is transmitted via DMSP for modules to send control and audio sample frames to each other in real time. Audio output is managed by an Inter-Integrated Circuit Sound (I2S) DAC, converting digital signals into high-quality analog audio. This modular approach allows for scalability, low production costs, and the versatility needed to meet the demands of musicians seeking true polyphonic capabilities.

### System Architecture

The diagram below shows the system overview of our Synthesizer. The busboard module provides clock and sync signals, and power to the system's modules. Each module is implemented on an RP2350 microcontroller and provides one function of an audio synthesizer. Our initial design implements MIDI in, oscillator, filter, and envelope, and Line-Out modules. The system is controlled via MIDI (Musical Instrument Digital Interface) which is the standard for electronic instruments. The output of the synthesizer is a module that converts our protocol to I2S at 24-bit 40kHz analog signal by a built-in DAC.

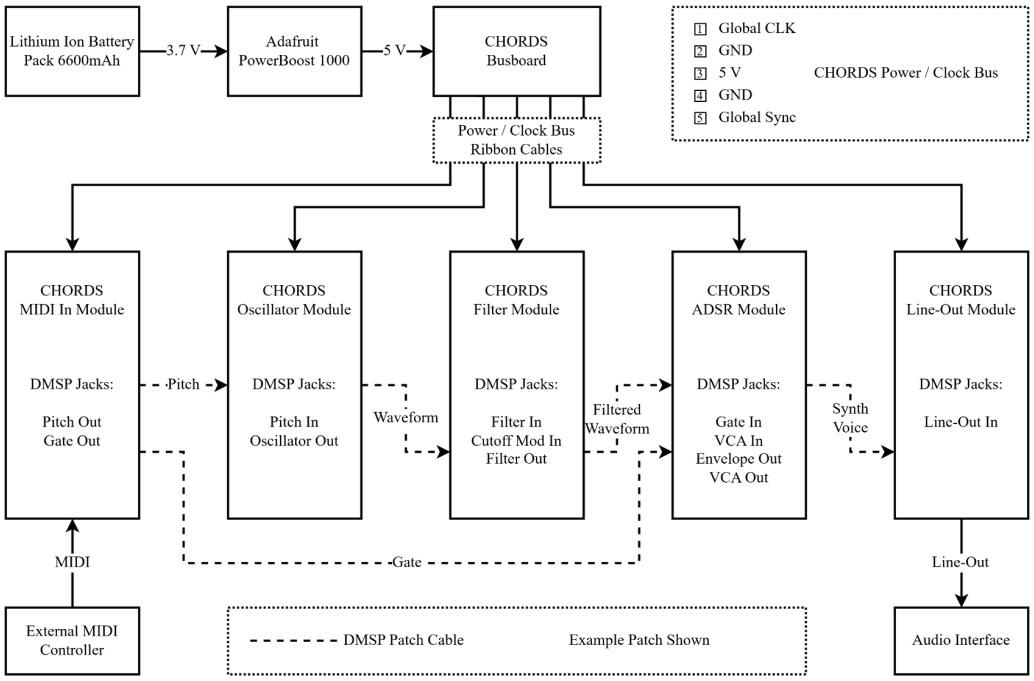


**Fig. 5.1: Synthesizer System Overview**

### Software

In order to send multiple channels of data, our goal was to develop a DSP program to create polyphonic sounds. The Digital Modular Synthesizer Protocol (DMSP) uses a 5.28 MHz Global Clock to sync data every 33rd bit. The Global Sync channel will start the transmission of a new frame. The figure below shows the timing diagram for the DMSP.

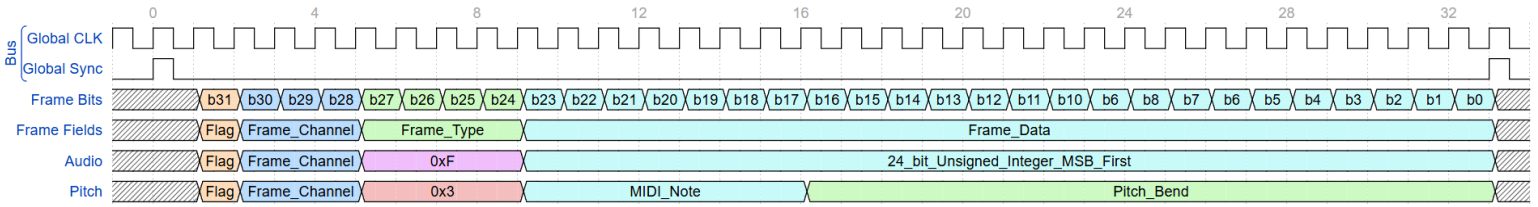


**Fig. 5.2: Digital Modular Synthesiser Protocol (DMSP) timing diagram**

The DMSP is transmitted through the TX and received through the RX or the RP2350. The next set of diagrams show where the DMSP is transmitted and received in the microcontroller.

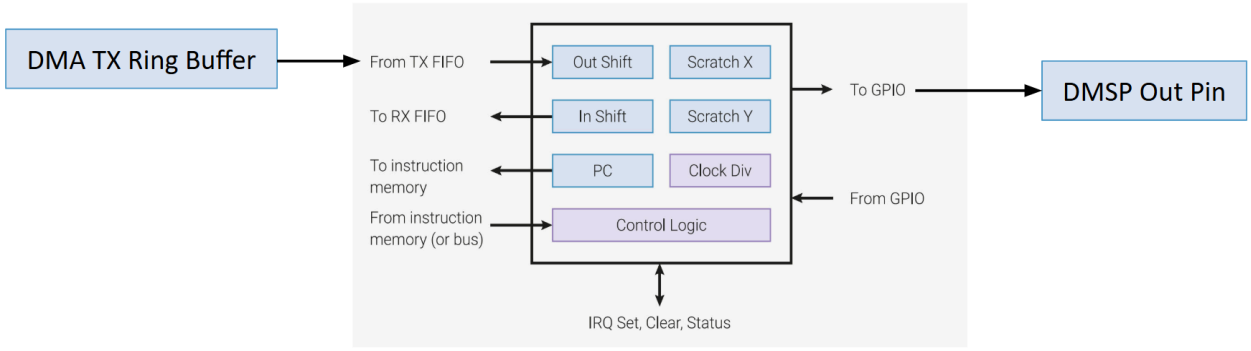


**Fig. 5.3: DMSP Transmitter diagram**

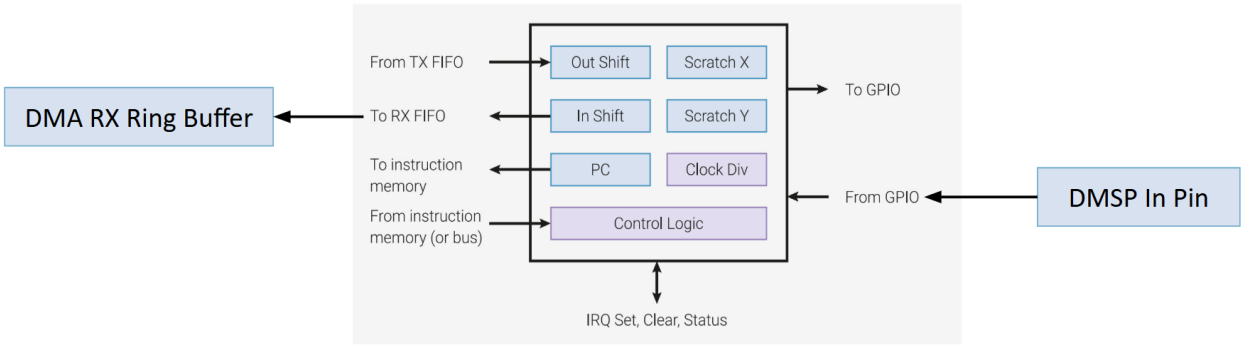


**Fig. 5.4: DMSP Receiver diagram**

In order to maintain the flow of data through CHORDS, ring buffers were used. This allows the data to be received in the RX of the RP2350, in the order it was transmitted from the TX. The ring buffer utilized can be seen below

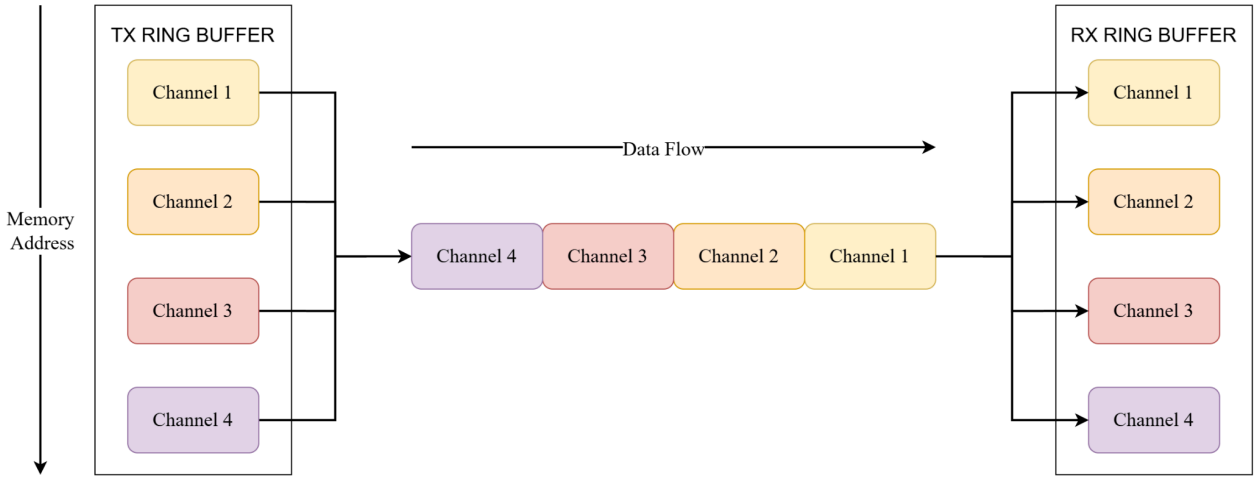


**Fig. 5.5: DMSP Ring buffer implementation**

The ring buffers improve consistency but introduce latency into the system. At a sample rate of 40 kHz, a buffer size of 256 frames, and 4 channels the maximum latency introduced by the ring buffers is 3.2 ms.

- **Sample Rate:** $f_s = 40\,\text{kHz}$ (1 sample every $25\,\mu\text{s}$)
- **Buffer Size:** 256 frames (64 samples/channel)
- **DMSP Channels:** 4 (frame-based, similar to TDM)

$$\begin{aligned}
\text{Samples per channel per buffer} &= \frac{256}{4} = 64 \\
\text{Buffer duration per channel} &= 64 \times 25\,\mu\text{s} = 1.6\,\text{ms} \\
\text{Per-module latency (worst-case)} &= \text{RX buffer} + \text{TX buffer} \\
&= 1.6\,\text{ms} + 1.6\,\text{ms} = \boxed{3.2\,\text{ms}}
\end{aligned}$$

The Digital Modular Synthesizer Protocol is a unique PIO program that is used to send data between each module. This flowchart outlines how the DMSP is transmitted and received between modules. The program is initialized at the falling edge of Global Sync bit, then waits 33 bits before sending another sync pulse. In those bits the data is either transmitted or received from the FIFO.

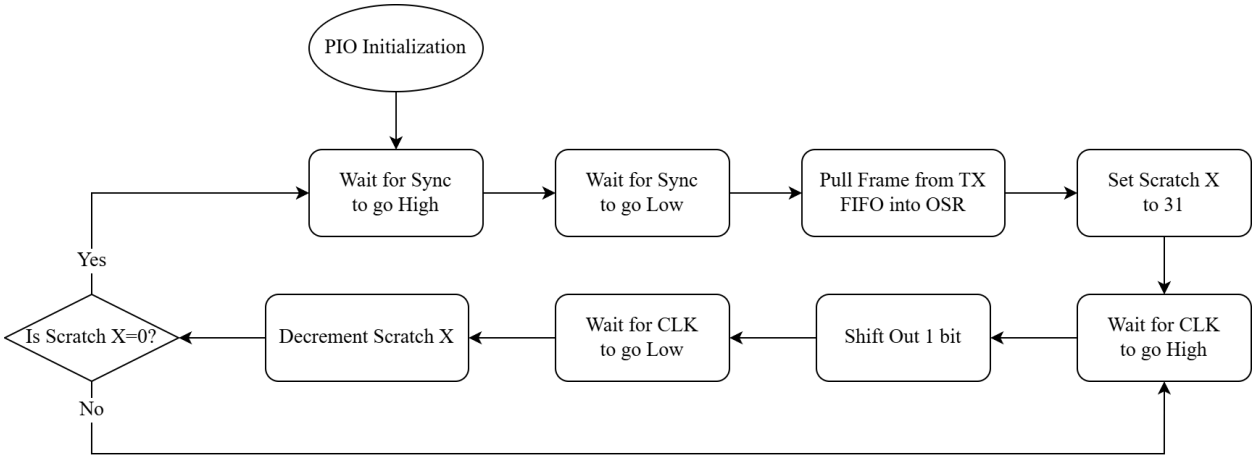


**Fig. 5.6: DMSP Transmitter flowchart**

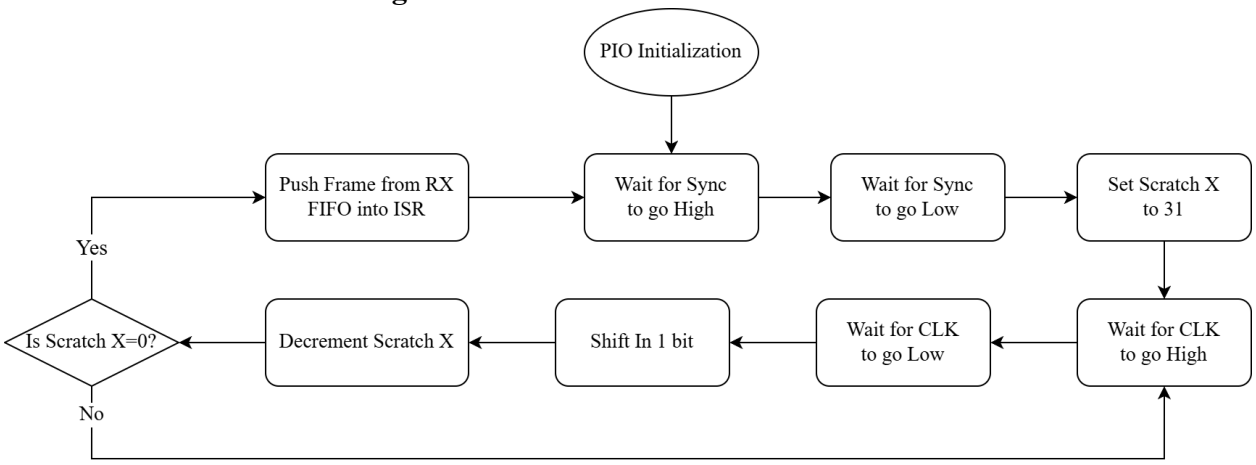


**Fig. 5.7: DMSP Receiver flowchart**

## Hardware

CHORDS is designed to be portable and cost effective for musicians to use and hobbyists to create. This is achieved with a 3D printed case and simple PCB designs. In this section the process of developing these components will be explained.

### PCB Design

Our PCB design is built to be adaptable to different types of modules. Each PCB has 8 aux jacks, 6 slots for knobs and 2 buttons. Depending on the module the PCB can be adapted to be a filter, oscillator or envelope. Below shows the KICAD schematic and PCB prototype for our module

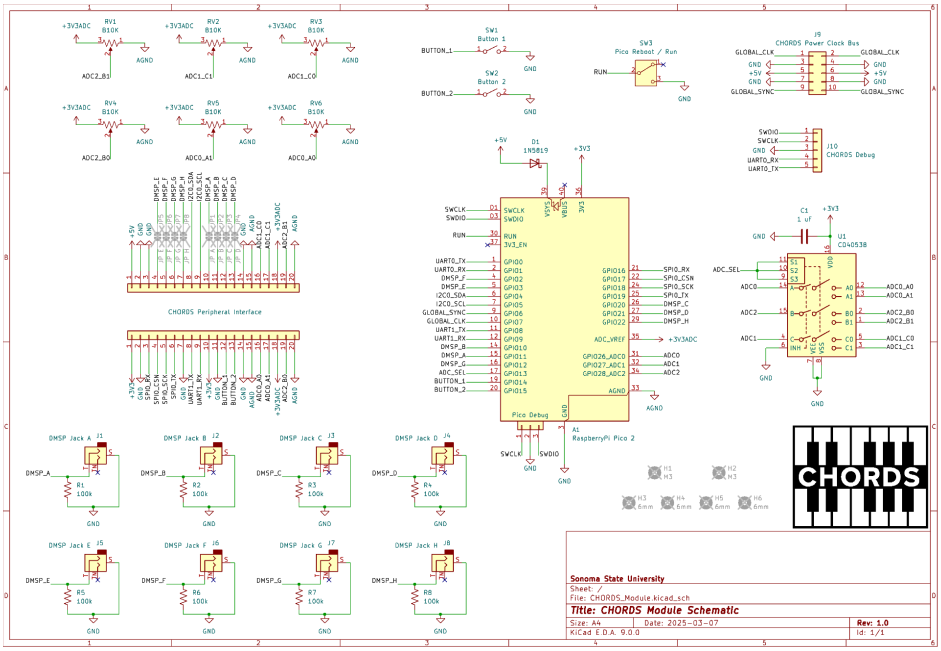


**Fig. 5.8: Module PCB Schematic**

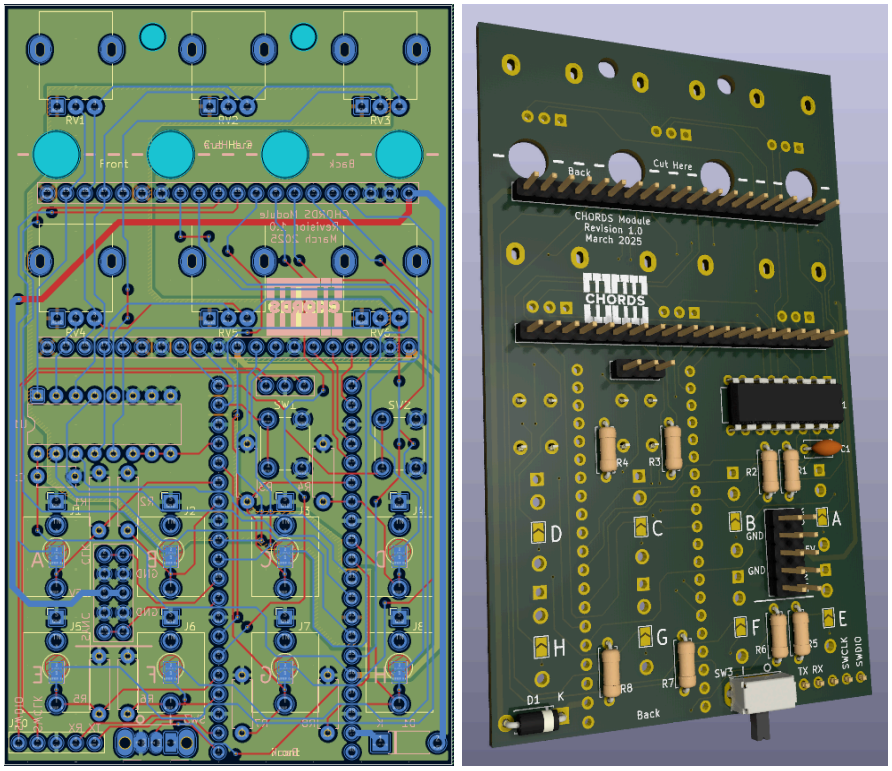


**Fig. 5.9: Module PCB prototype**

The PCB Busboard is designed to allow 10 modules to be connected and in sync with each other using our DMSP. The figures below the KiCAD schematic and prototype for the busboard

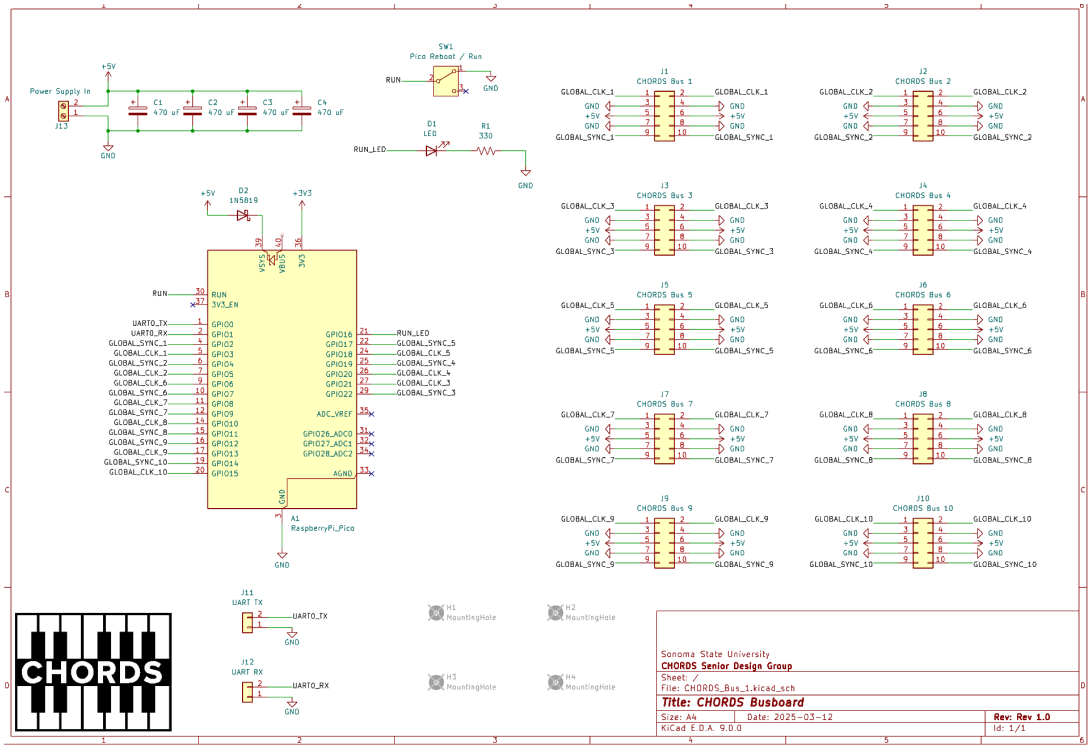

**Fig. 5.10: Busboard PCB Schematic**

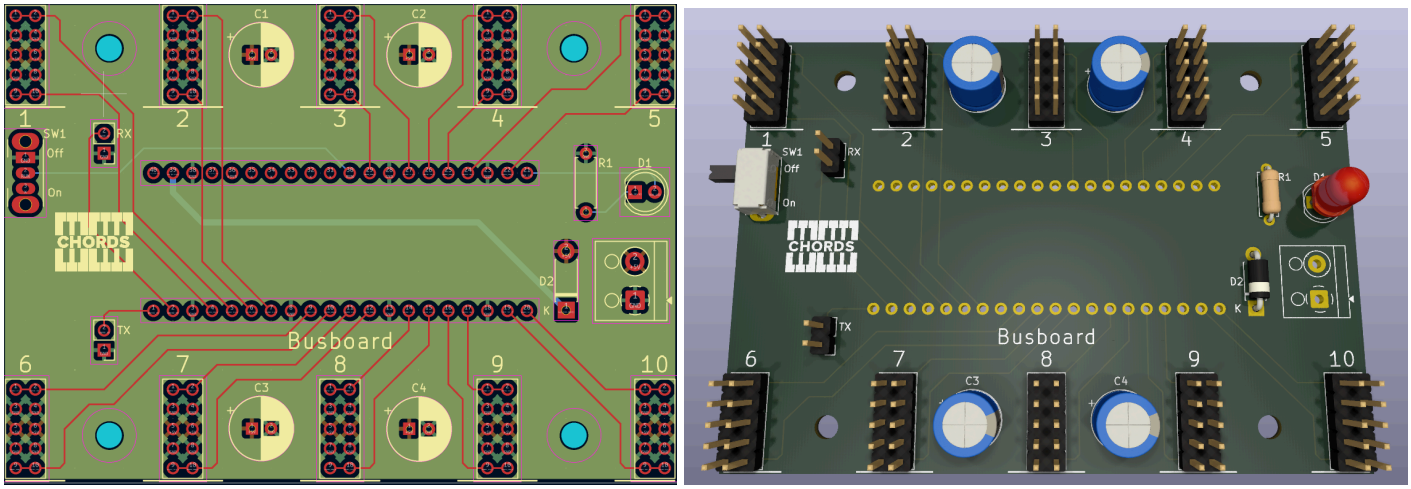


**Fig. 5.11: Busboard PCB prototype**

## 3D Printing

For the case, faceplate, knobs, and stand the SSU makerspace allowed us to 3D print various components for CHORDS. Using Fusion 360 all external components of CHORDS were able to be designed and inevitably printed.

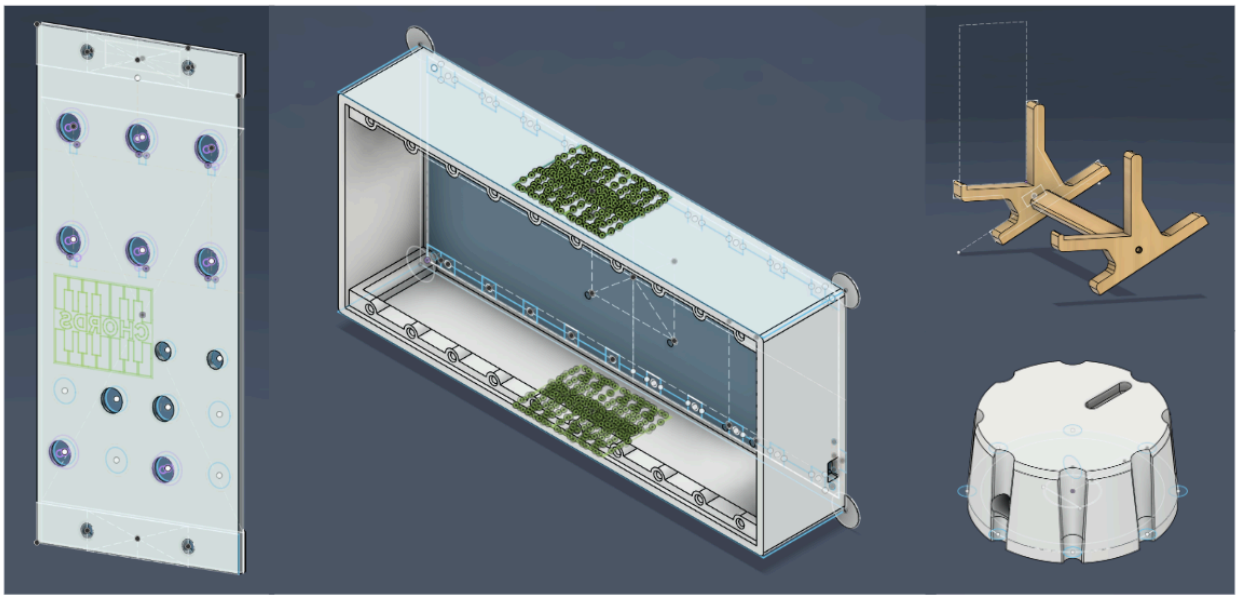

**Fig. 5.12: Fusion360 prototype (left to right: Faceplate, Case, Stand, and Knobs)**

**Prototyping**

To test different module designs, a prototype module was created. This prototype had every audio jack and knob attached. This allowed us to test different layouts for every type of module. The figures below show the completed test module prototype that was used.

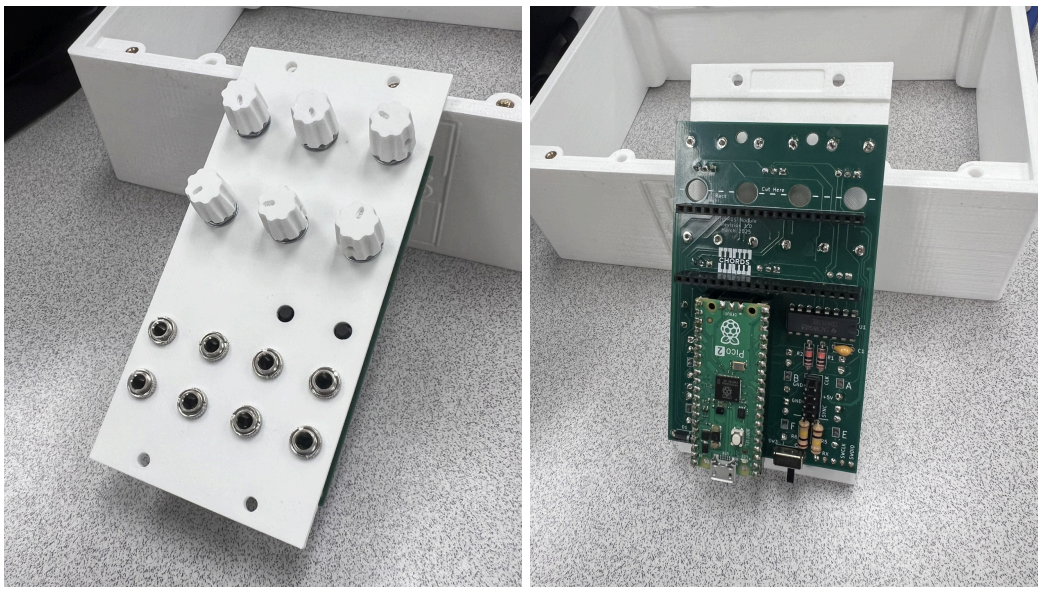

**Fig. 5.13: Complete Module prototype**

Once all modules software was tested on the CHORDS module prototype, unique faceplates and layouts were created to customize our CHORDS synth to what we wanted. The photo below shows the final version of the CHORDS prototype.

**Fig. 5.14: Complete CHORDS prototype**

**Alternative design matrix for synth microcontroller:**

The main component used in CHORDS is the Pico-2 RP 2350. This microcontroller is ideal due to its low cost, high clock speed, and memory capacity. One of our main goals is to build this system at an affordable cost compared to other modular synthesizers. Currently, each module in a modular synthesizer can run a musician over $200. We intend to build the whole system for less than that. The main way to sync each module and implement the DMSP is by manipulating the clock speed of the Pico-2. By manipulating the 150MHz clock, we can send a clock sync through a main busboard module that will sync the addiction modules together with a low latency.

**Table I: Microcontroller alternate design matrix**

| | RP2350 (pico 2) | RP2040 (pico 1) | Teensy 4.0 | ESP8266 |
|---|---|---|---|---|
| Cost (0.5) | 0.3243 (5.00$) | 0.4054 (4.00$) | 0.0676 (23.80$) | 0.2027 (7.99$) |
| Clock Speed (0.25) | 0.2442 (150 MHz) | 0.1250 (100 MHz) | 0.4769 (600 MHz) | 0.1538 (80 MHz) |
| Memory (0.25) | 0.2942 (520 KB) | 0.1541 (264 KB) | 0.4703 (1024 KB) | 0.0813 (112 KB) |
| Score | 29.68% | 27.25% | 27.06% | 16.01% |

**Alternative design matrix for audio output module:**

When deciding on audio output modules, a similar focus on cost is calculated. We wanted to ensure that the cost was low while maintaining a sample rate of 24-bit at 40 KHz. Compared to other DAC audio output modules at the same sample rate, the I2S Stereo Decoder from AdaFruit comes at a price of less than $7. With this price we are able to maintain our goal of less than $200 for the entire system while achieving our audio output quality expectations.

**Table II: DAC alternate design matrix**

| | I2S Stereo Decoder (AdaFruit) | I2S Audio Breakout (Sparkfun) | ESP32 with I2S | DAC (HiFiBerry) |
|---|---|---|---|---|
| Cost (0.45) | 0.45 ($6.95) | 0.22 ($17.95) | 0.28 ($22.99) | 0.05 ($58.58) |
| Output Quality (Sample Rate: 24-bit 44.1KHz) (0.33) | 0.175 (24-bit 44.1KHz) | 0.175 (24-bit 44.1KHz) | 0.20 (24-bit 44.1KHz, less noise) | 0.45 (24-bit 192KHz) |
| Capability & Additional Features (0.22) | 0.225 | 0.10 | 0.275 | 0.40 |
| Score | 30.975% | 17.875% | 25.25% | 25.9% |

# 6. List of Tests

### Summary of Tests

Below, we present a summary of tests that are conducted so far.

**Table III: Summary of conducted tests**

| Test Number | Objective | ER to address | Status | Notes |
|---|---|---|---|---|
| ST. 1 | Global CLK and Sync Alignment Over Time | ER. 2, 3 | Pass | CLK = 5.28 +/- 0.01 MHz |
| ST. 2 | Full System Test on Breadboards | ER. 1, 2, 3, 4, 5, 6, 7 | Pass | Recorded Demo |
| ST. 3 | Average Power Draw | ER. 9 | Pass | Average current = 180.60mA, Calculated 25.7 hours of battery life |
| ST. 4 | Full System Latency | ER. 3 | Pass | Average system latency of 13.40ms |
| FT. 1 | Oscillator Output | ER. 2, 5 | Pass | Tuned to A= 440.32 Hz |
| FT. 2 | MIDI Input | ER. 1, 2, 3, 5 | Pass | < 5ms of Latency |
| FT. 3 | Total Harmonic Distortion | ER. 6 | Fail | THD 5.93% |
| FT. 4 | Filter Module | ER. 1, 3, 7 | Pass | Filter from 15 Hz - 20 kHz, with resonance knob |
| FT. 5 | I2S Module | ER. 3, 6 | Pass | 40 kHz sample rate at 24-bit |

### 1. ST 1: Global Clock Accuracy and Stability

**Objective**: The purpose of this test is to evaluate the long-term timing stability and phase consistency of the Global Clock (Global CLK) and Synchronization (Sync) signals distributed via the CHORDS Busboard. This test specifically targets slow phase drift across modules rather than short-term jitter, which is critical for ensuring that all modules on the Digital Modular Synth Protocol (DMSP) remain synchronized during operation [13].

**Set-up**: The test system consisted of a MIDI controller connected to the MIDI module of the synthesizer, which acted as the master clock source. The Busboard module distributed a 5-MHz Global CLK and Sync signal to all slave modules via the CHORDS Busboard. Each module contained its own RP2350 microcontroller, configured to interpret the clock signal, parse DMSP frames, and activate a digital output upon receiving valid data. The oscilloscope was connected to both the clock line and the Sync signal line to visually confirm timing alignment over time [14].

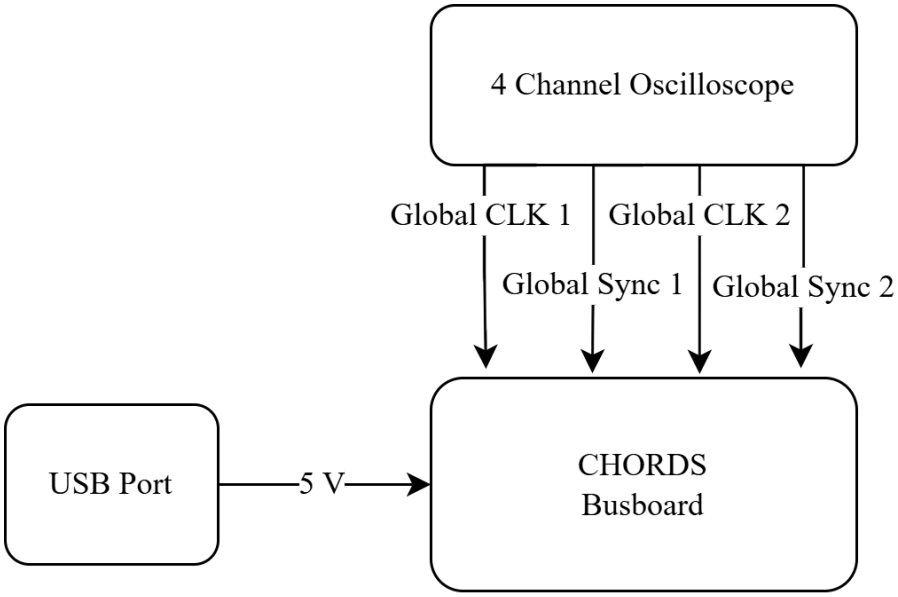


**Fig. 6.1: Global Clock Accuracy and Stability Test setup**

**Procedure**: A MIDI signal was sent from the controller to the central module to initiate data transmission.

Each module on the CHORDS Busboard was programmed to:

- Receive the Global CLK and Sync lines
- Parse the synchronization bit and relevant channel address,
- Activate a corresponding LED if the received address matches the module's

assigned channel.

The oscilloscope was used to measure:

- The frequency of the Global CLK signal,
- The phase relationship between CLK and Sync,
- Stability of these signals over an extended observation window.

**Results:** The oscilloscope confirmed a stable 5 MHz square wave on the Global CLK line with minimal variation in duty cycle. Sync pulses were consistently aligned with the leading edge of the Global CLK cycle. No observable drift or skew was detected between modules during the test duration. All functional modules responded in real-time to incoming MIDI events, confirming successful parsing of the DMSP frame and channel identification.

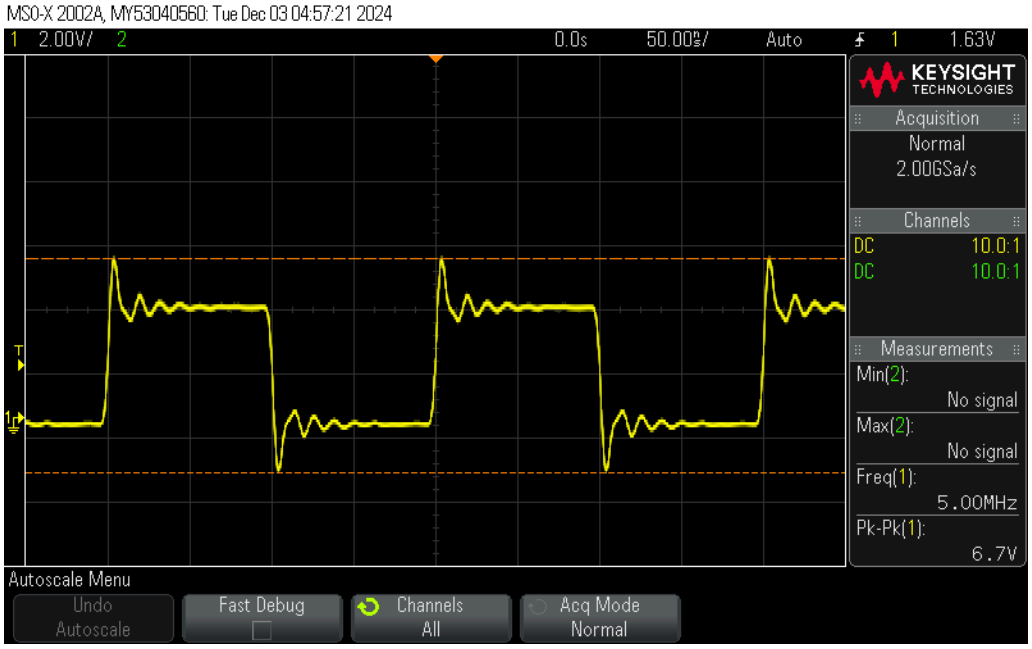


**Fig. 6.2: Global Clock Accuracy and Stability oscillator Output showing a consistant 5MHz clock signal**

**Conclusion**: The CHORDS Busboard's distributed clocking system demonstrated excellent phase stability and consistent performance across modules. The 5 MHz clock provided a reliable baseline for transmitting multi-channel DMSP frames. The lack of observable drift suggests that the RP2350-based modules can maintain synchronized behavior even under prolonged use [4], [14]. Importantly, this clock rate is scalable: increasing the frequency could

theoretically support higher channel bandwidth, enabling more voices or finer resolution in control signals [13]. This result confirms the core viability of the DMSP framework for low-latency, polyphonic digital modular synthesis.

### 2. ST 2: Full System Test on Breadboards

**Objective**: To evaluate the CHORDS system's capability to process and output audio signals from MIDI input and confirm its usability in a digital audio production environment. The test focuses on verifying that the entire system, comprising MIDI input parsing, signal synthesis, audio output, and DAW recording, is functioning as a cohesive musical tool.

**Set-up**: For this test, a MIDI sequence was created within a DAW (Ableton Live) and transmitted via a USB-MIDI interface to the CHORDS synthesizer. The synthesizer, assembled on breadboards using multiple RP2350 microcontrollers, parsed the MIDI messages and routed the frequency data through its DMSP-controlled audio modules. Each module contained core synthesis components including oscillators and envelopes. The system's audio output was routed through an I²S-to-analog DAC circuit and connected to an audio interface, which returned the signal to the DAW for monitoring and recording [6], [4].

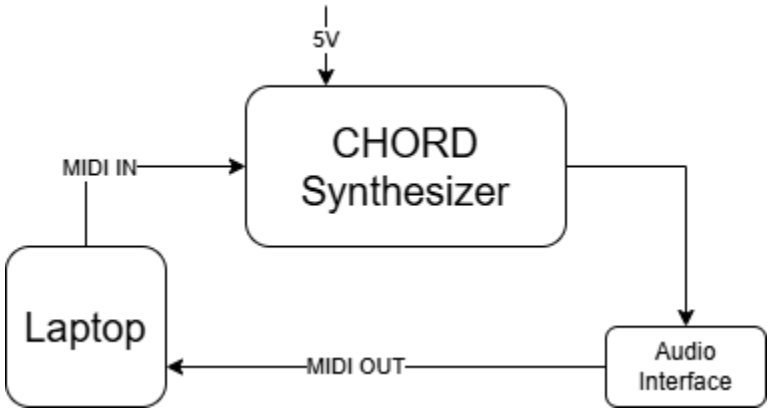


**Fig. 6.2: Full System Test on Breadboards setup**

**Procedure**:

- The DAW was used to send a test MIDI note (A4 = 440 Hz) to the system.
- The oscillator module was configured to generate a sine wave based on the frequency data.
- Output from the DAC was connected to the audio interface line-in port.

- The DAW was used to monitor and record the audio signal.
- Waveform analysis was performed in the DAW to verify frequency accuracy, signal stability, and phase alignment.

**Results**: The system successfully interpreted MIDI input and generated a consistent audio signal at 440 Hz. The waveform displayed in the DAW had minimal harmonic distortion and matched the expected sine profile, indicating proper digital-to-analog conversion and oscillator accuracy. Latency was not perceivable in playback, consistent with embedded systems operating below a 10 ms audio threshold, which is generally considered imperceptible to human listeners [13].

The output was stable across multiple modules operating in parallel, confirming the integrity of the CHORDS Busboard and global synchronization protocol. Real-time performance was validated by the responsive behavior of the oscillator and filter modules under rapid MIDI changes.

Audio recordings from the test are available at:
https://chordsynthesizer.com/Documentation/demo.php

**Conclusion**: This full-system test demonstrates that the CHORDS synthesizer is capable of polyphonic audio generation from MIDI input and can be successfully integrated into DAW environments for recording and production. The accurate 440 Hz response, clean waveform output, and DAW synchronization confirm that the digital signal chain from MIDI parsing through DMSP control to analog audio output is operating correctly. These results support the system's practical use as a modular, digitally controlled polyphonic synthesizer for modern audio workflows.

### 3. ST 3: Average Power Draw

**Objective**: To characterize the power consumption of the CHORDS synthesizer system during normal operation. This test is designed to capture both the steady-state and dynamic power behavior of the system while processing MIDI data and generating audio, providing

insight into the system's energy efficiency and expected runtime under battery power.

**Set-up**: The Nordic Power Profiler Kit II was connected in-line between the CHORDS system and its power source to measure current draw. The CHORDS system was configured for full operation, with all core modules (including MIDI input, DMSP control, oscillators, and audio output) running continuously. MIDI data was streamed from a DAW to simulate real usage conditions. The system was powered by a standard 5V lithium-ion battery pack rated at 4400 mAh, commonly used in embedded applications.

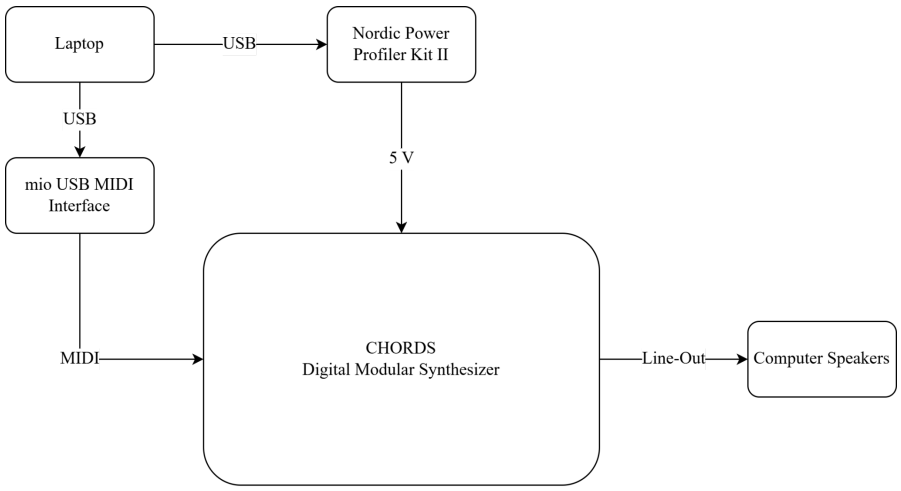


**Fig. 6.3: Average Power Draw test setup**

**Procedure**:

- The Power Profiler was calibrated and connected to the CHORDS power input.
- The DAW sent a continuous MIDI signal to the CHORDS system to engage all modules during operation.
- Current measurements were logged over a 5-minute period using the Power Profiler software.
- The average current draw was recorded and used to estimate expected battery life.

**Results**: The Power Profiler reported a consistent average current draw of 180.60 mA during active use. This value remained stable with minimal fluctuation, suggesting efficient load balancing across modules and limited power spiking. Using the average current and the battery capacity (4400 mAh), the expected battery life can be calculated as:

$$\text{Battery Life} = \frac{6600 \text{ mAhr} \times 3.7 \text{ V} \times 0.95}{180.60 \text{ mA} \times 5 \text{ V}} = 25.7 \text{ hr}$$

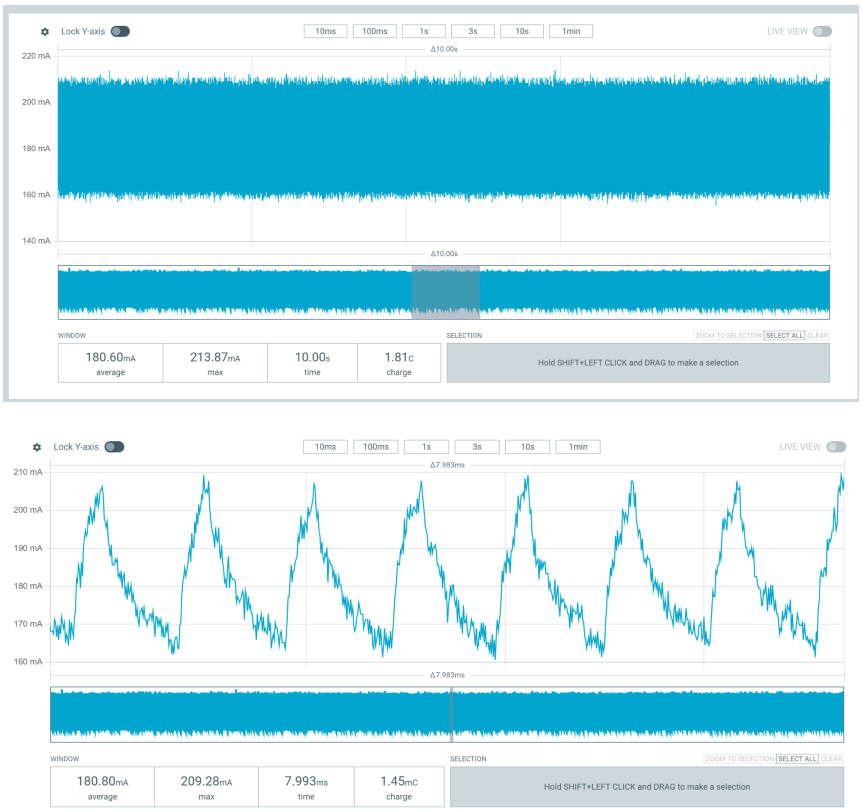


**Fig. 6.5: Average Power Draw test output**

**Conclusion**: The CHORDS system demonstrates excellent power efficiency for a multi-module, real-time audio synthesizer. With an average current draw of approximately 180 mA, the system can operate continuously for ~25 hours on a standard portable battery. This level of performance supports the system's use in live performance, mobile recording, or field experimentation without requiring frequent recharging. [4], [13] (ER 9)

### 4. ST 4: Full System Latency

**Objective**: To measure the end-to-end latency of the CHORDS system from MIDI input to DAC audio output. This test evaluates the cumulative delay introduced by DMSP buffering, per-module frame processing, and digital-to-analog conversion. A total of 8 DMSP transmit (TX) and receive (RX) ports were included in the signal chain, each contributing to system-wide latency.

**Set-up**: Two oscilloscope probes were used to measure latency across the full CHORDS system:

- Probe 1: was placed at the MIDI input line to detect the beginning of the MIDI signal transmission.
- Probe 2: was placed at the I²S audio output DAC to detect the start of the corresponding analog audio waveform.

The test involved 5 active CHORDS modules, each running on an RP2350 microcontroller and operating with synchronized clocks via the CHORDS Busboard. MIDI notes were triggered from a DAW, and the oscilloscope was set to capture both probe signals in a single acquisition window with time resolution in microseconds.

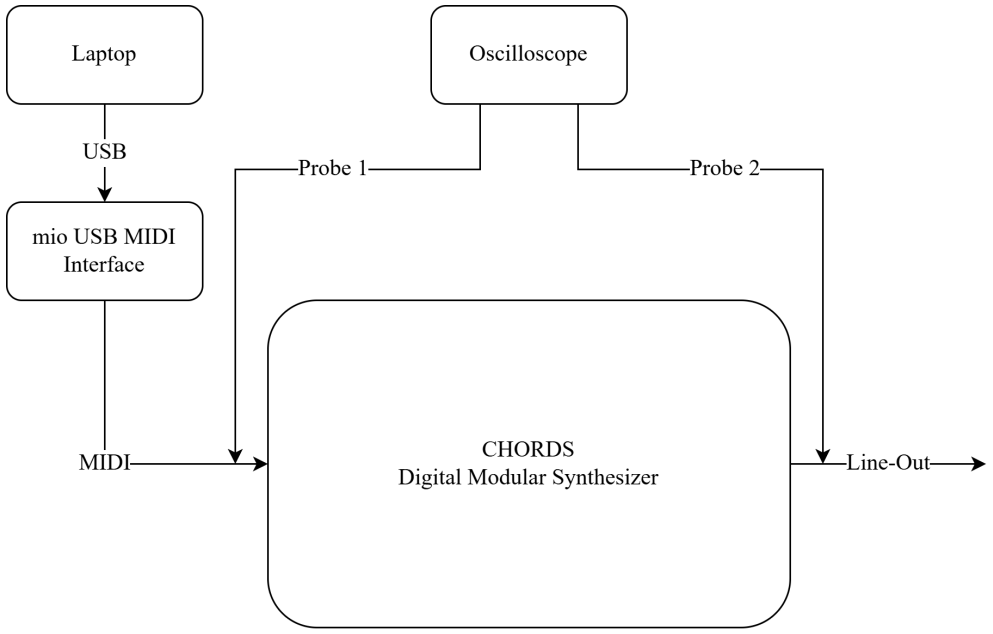


**Fig. 6.6: Full System Latency test setup**

**Procedure**:

- A single MIDI note (A4 = 440 Hz) was transmitted repeatedly from the DAW.
- For each trial, the time delta between Probe 1 (MIDI trigger) and Probe 2 (audio output) was measured using the oscilloscope.
- This process was repeated nine times to capture average latency under consistent processing conditions.
- Latency measurements were tabulated and averaged to characterize overall system response time.

**Results**: The average system latency was calculated to be 13.03 ms, well within the acceptable range for musical instrument response. According to human perception studies, latency below 15–20 milliseconds is generally imperceptible to most musicians and does not interfere with timing or rhythmic perception during live performance [15].

| Test Number: | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Latency (ms): | 15.48 | 10.22 | 13.96 | 13.04 | 15.32 | 10.88 | 16.20 | 12.24 | 13.28 |

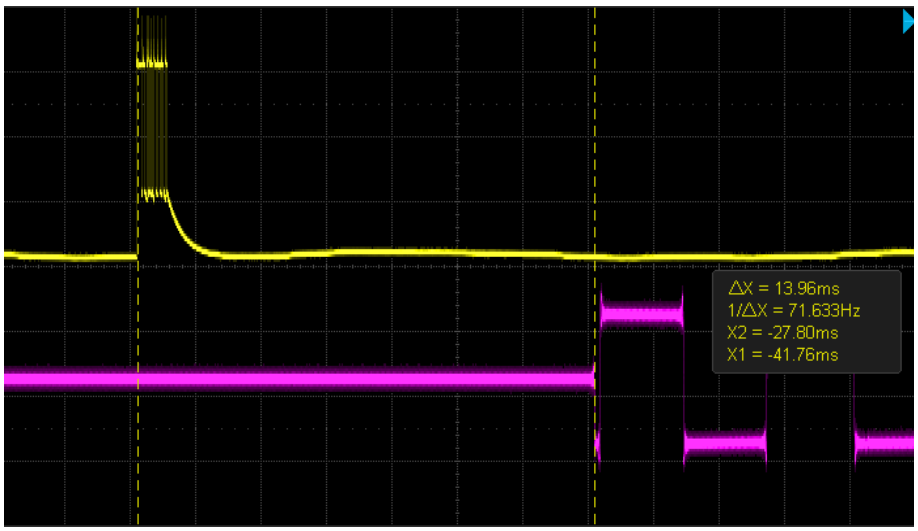


**Fig. 6.7: Full System Latency oscilloscope output measurement**

**Conclusion**: With an average full-system latency of 13.03 ms, the CHORDS synthesizer performs within acceptable bounds for real-time audio applications. The test confirms that despite multiple frame-processing stages and buffered communication between modules, CHORDS maintains a responsive user experience. The result is consistent with expectations for embedded audio systems utilizing I²S audio output and serial MIDI input [5], [6]. This validates the architectural viability of DMSP and confirms that the system can be reliably used in live or studio environments with no perceptible delay.

Each TX or RX stage introduces:

$$\text{Latency}_{\text{per DMSP}} = \frac{256}{4} \times 25\,\mu\text{s} = 1.6\,\text{ms}$$

**Configuration:**

- 3 full modules (TX+RX): $3 \times 3.2 = 9.6$ ms
- 1 TX-only module (MIDI): 1.6 ms
- 1 RX-only module (Line-Out): 1.6 ms

$$\text{Total theoretical latency: } 9.6 + 1.6 + 1.6 = \boxed{12.8\,\text{ms}}$$

**Measured latency:** 10.2 ms to 16.2 ms (average: 13.2 ms)

## 5. FT 1: Oscillator Output

**Objective:** The objective of this test is to validate the Oscillator module's ability to generate a sine waveform tuned to A4 (440 Hz), which serves as the standard tuning reference in Western music [1]. This signal is essential for musical accuracy when processing MIDI input and routing audio through subsequent CHORDS modules. The oscillator's output must be accurate and stable to ensure compatibility with musical standards synthesis workflows.

**Set-up:** The test involved three active CHORDS modules: the MIDI Interface, Oscillator, and Audio Output module. MIDI data was sent from a DAW to trigger the oscillator via the CHORDS Busboard. An oscilloscope was used to measure:

- The Global Clock signal,
- The Data line carrying the DMSP payload, and
- The analog output from the I²S DAC following waveform generation.

The RP2350-based oscillator module contained a precomputed wavetable used to synthesize the 440 Hz tone in response to the incoming MIDI pitch data.

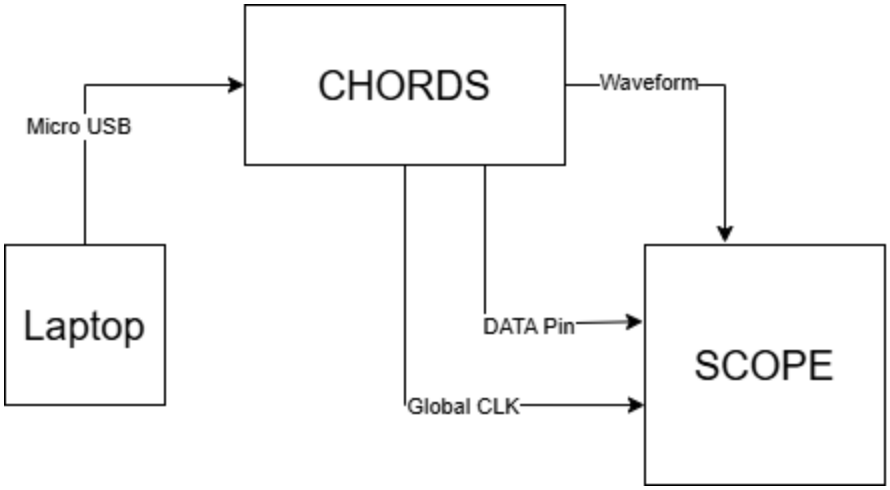


**Fig. 6.8: Oscillator Output test setup**

**Procedure:**

- A MIDI note corresponding to A4 (note number 69) was transmitted from the DAW into the CHORDS system.
- Upon powering the oscillator, it began waveform generation using its internal wavetable oscillator.
- The oscilloscope's measurement cursors were used to examine the analog output waveform.
- Frequency was calculated based on the period (T) of the waveform, using the relation:

$$f = 1/T$$

**Results**: The oscillator successfully generated a clean sine wave at 440 Hz, as measured by the oscilloscope. The waveform was stable and periodic, indicating a correctly implemented lookup table (LUT) oscillator structure, a common method in embedded DSP applications due to its balance of performance and memory efficiency [13].

Additional waveforms (square and triangle) were also confirmed to be selectable via software control, though 440 Hz sine wave output was the focus of this functional test.

**Conclusion**: This test confirms that the Oscillator module reliably generates standard musical tones based on incoming MIDI pitch data. The accurate 440 Hz output provides a verified reference point for further testing and usage, ensuring the CHORDS system is usable for musically correct synthesis. This functionality is foundational for any modular or polyphonic

synthesizer and verifies the oscillator's core role within the CHORDS architecture.

### 6. FT 2: MIDI input

**Objective**: To validate the correct operation of the CHORDS MIDI input module by verifying that MIDI messages sent from a standard controller are accurately received, interpreted, and passed to the RP2350 microcontroller for further processing. This test ensures that the isolated input circuitry is functioning correctly and that the system meets latency requirements for real-time musical interaction.

**Setup**: The MIDI input circuit was assembled on a breadboard, consisting of a 5-pin DIN MIDI connector, a 6N138 optocoupler, and a resistor network designed according to the MIDI 1.0 electrical specification [5]. The optocoupler provides galvanic isolation between the MIDI controller and the CHORDS system, a standard safety and noise mitigation measure in audio electronics.

The optocoupler output was connected to an RP2350 microcontroller, configured to interpret serial MIDI data at the standard baud rate of 31.25 kbps. Two oscilloscope probes were used to monitor:

- Probe 1: The input MIDI data line before entering the optocoupler.
- Probe 2: The digital output from the RP2350's flag pin, triggered once a complete MIDI message was received.

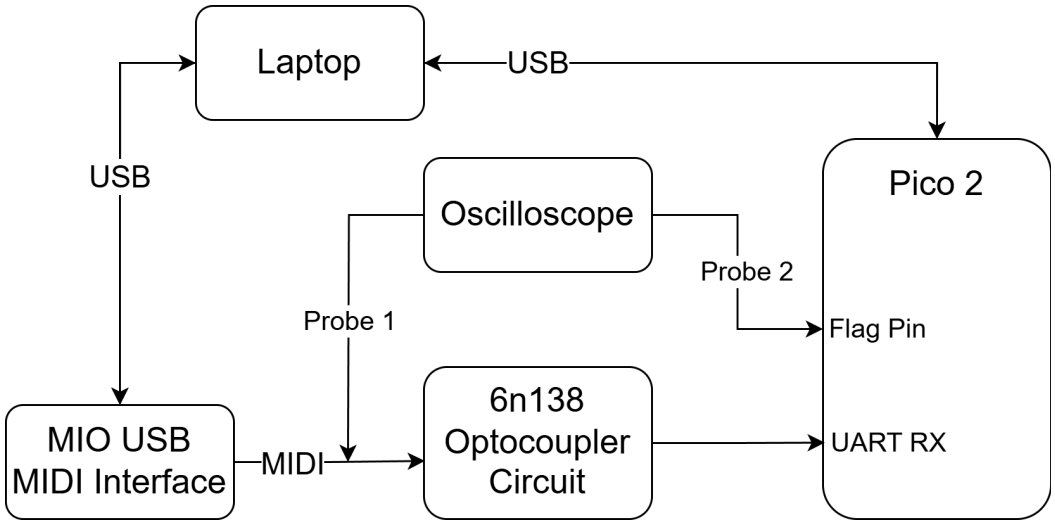


**Fig. 6.9: MIDI Input test setup**

**Procedure**:

- A MIDI keyboard was used to send Note On and Note Off messages to the CHORDS MIDI input.
- The oscilloscope was used to measure the time delta between the rising edge of the MIDI input signal and the rising edge of the RP2350's flag pin.
- Multiple trials were run to confirm consistency and latency performance.

**Results**: The system correctly received and interpreted MIDI messages, and the RP2350 microcontroller responded with a valid flag signal. The measured latency from signal entry to message interpretation was 4.798 ms, which is well within the acceptable range for MIDI handling in real-time audio systems. For context, MIDI itself introduces inherent delays of approximately 1 ms per three-byte message at its standard baud rate, meaning additional latency under 5 ms is typical for embedded decoding [5], [13]. No signal degradation or transmission errors were observed, confirming correct optocoupler function and electrical compatibility.

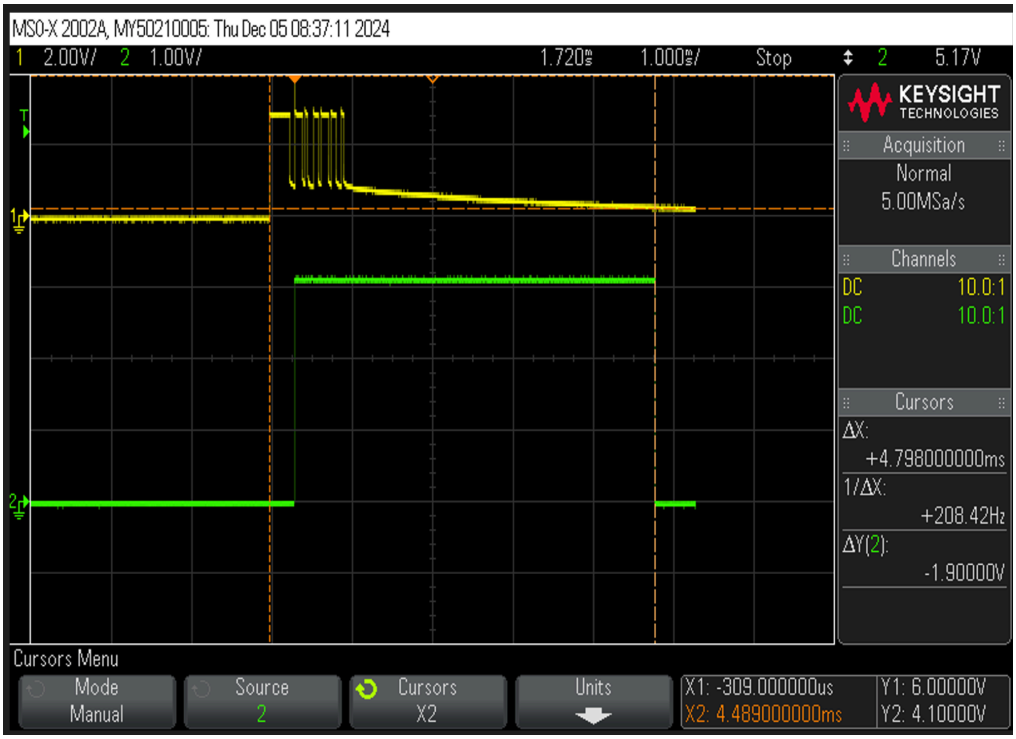


**Fig. 6.10: MIDI Input test oscilloscope output measurement**

**Conclusion**: The MIDI input module functioned as intended, successfully decoding serial MIDI messages with low latency and high signal integrity. The measured 4.798 ms latency

satisfies real-time audio responsiveness criteria [15] and establishes a reliable input path for use with other CHORDS modules. Future module-level latency testing will follow a similar oscilloscope-based method to ensure synchronization across the system.

### 7. FT 3: Harmonic Distortion in Audio Output Module:

**Objective**: To evaluate the CHORDS system's ability to produce a low-distortion sine wave by measuring Total Harmonic Distortion (THD) across the oscillator and I²S output modules. The goal was to achieve a THD of less than 1%, corresponding to a harmonic content at least 40 dB below the fundamental, which is commonly used as a threshold for acceptable waveform purity in synthesizer design [13].

**Set-up:**

- MIDI input data was used to trigger a 440 Hz sine wave from the CHORDS oscillator module.
- The output was routed through the I²S DAC module to a 3.5 mm audio output jack.
- The system was powered using a lithium-ion battery pack to avoid introducing switching noise from bench power supplies.
- The analog signal was measured using the Analog Discovery 2 data acquisition device.
- THD was analyzed in the WaveForms Spectrum Analyzer, capturing harmonic energy up to the 10th harmonic for accuracy.

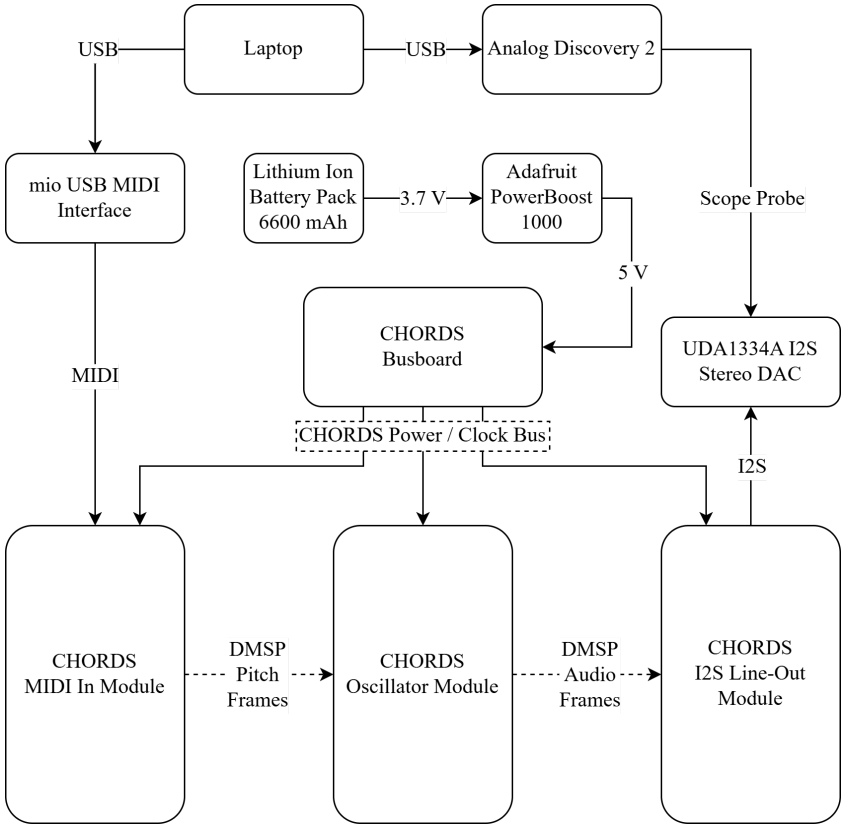


**Fig. 6.11: Harmonic Distortion in Audio Output Module test**

**Procedure**:

1. Initialize CHORDS and load a single-note MIDI signal to generate a 440 Hz sine wave.
2. Ensure that the oscillator and I²S modules are synchronized via the DMSP and global clock.
3. Route the analog audio output to the Analog Discovery 2 input and launch the spectrum analyzer.
4. Measure the amplitude of the fundamental frequency and each harmonic.
5. Compute THD using the standard formula:

$$\text{THD} = \frac{\sqrt{V_2^2 + V_3^2 + V_4^2 + \cdots}}{V_1}$$

**Results**: The CHORDS system produced a sine wave with a measured **THD of 5.93%**, which exceeds the original design goal. To contextualize this performance, the same test method was applied to a variety of commercial synthesizers and pure wave generators:

CHORDS

THD = 5.93%

| | root | 1st | 2nd | 3rd | 4th | 5th |
|---|---|---|---|---|---|---|
| Freq | 440 Hz | 880 Hz | 1,320 Hz | 1,760 Hz | 2,200 Hz | 2,660 Hz |
| dBV | -16.72 | -65.13 | -45.18 | -61.49 | -44.88 | -59.18 |
| Volts | 0.146 | 0.001 | 0.006 | 0.001 | 0.006 | 0.001 |

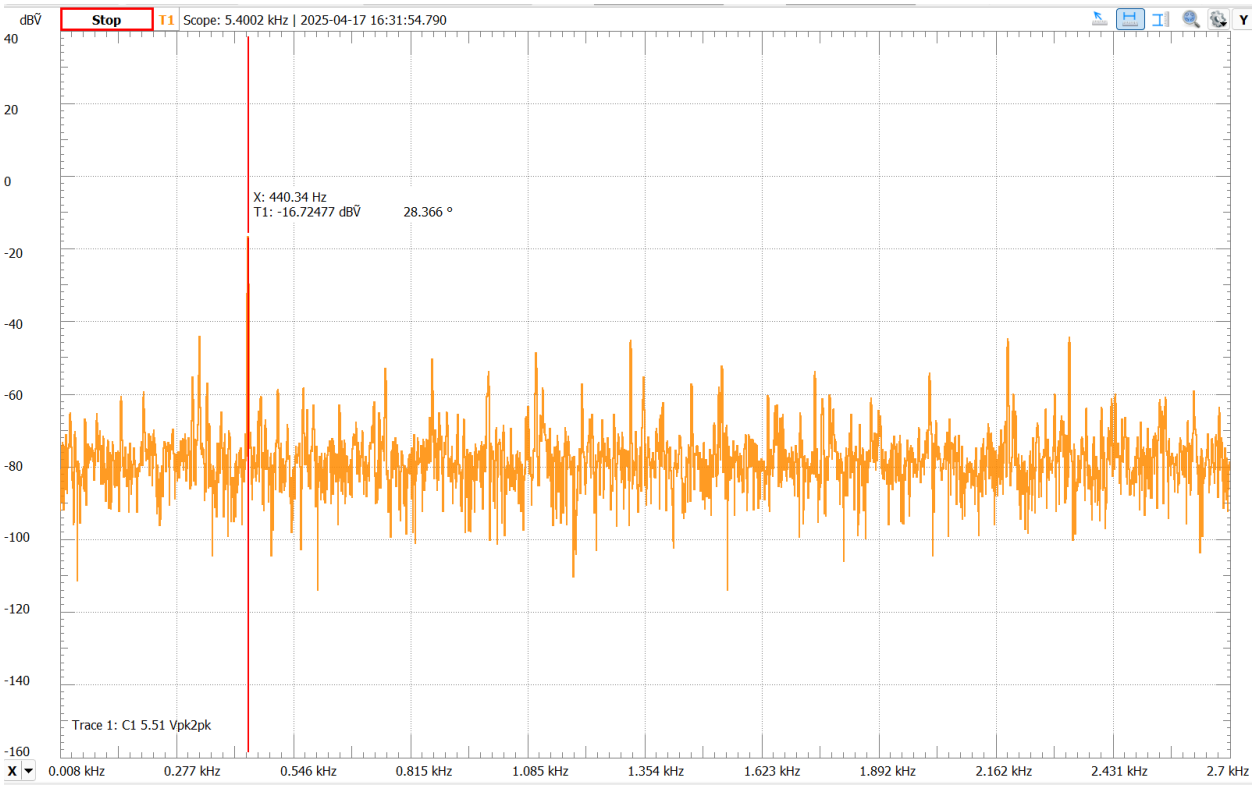


**fig. 6.12: CHORDS THD measurement**

Korg Arp 2600

THD = 5.14%

| | root | 1st | 2nd | 3rd | 4th | 5th |
|---|---|---|---|---|---|---|
| Freq | 440 Hz | 880 Hz | 1,320 Hz | 1,760 Hz | 2,200 Hz | 2,660 Hz |
| dBV | -1.61 | -27.87 | -40.61 | -39.39 | -48.40 | -54.81 |
| Volts | 0.831 | 0.04 | 0.009 | 0.011 | 0.004 | 0.002 |

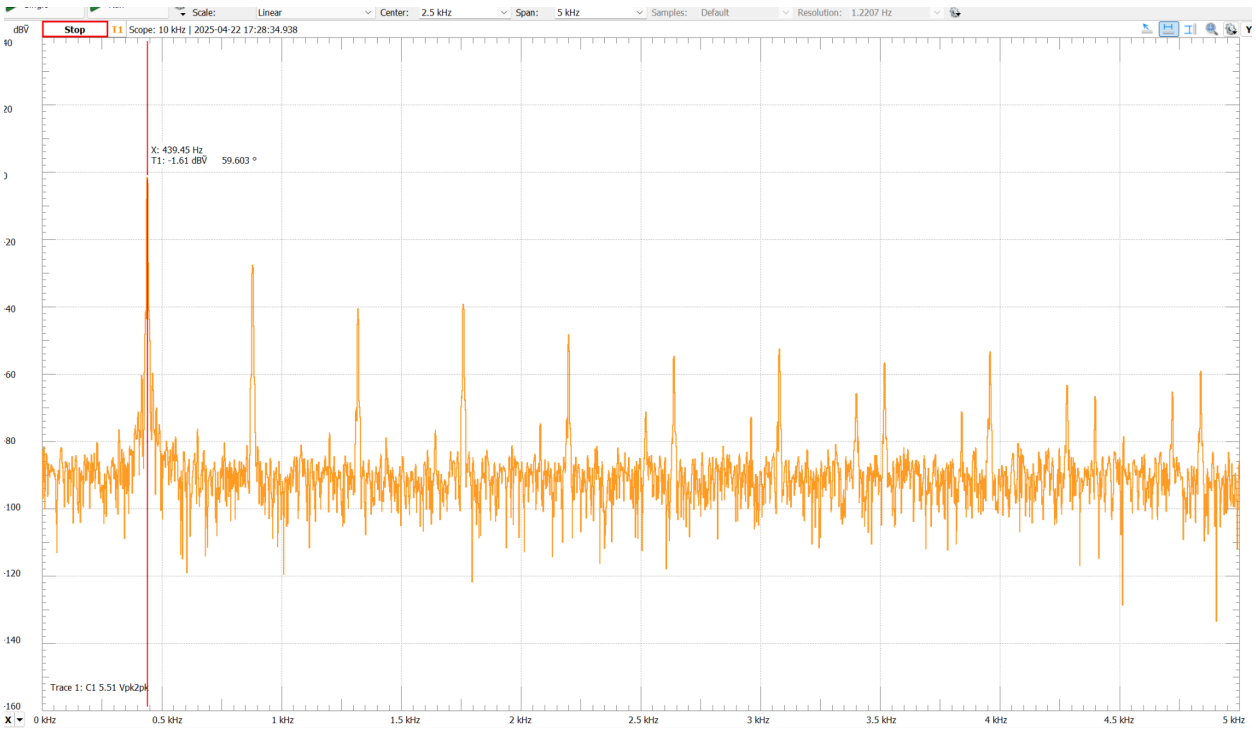


**fig. 6.13: KORG Arp 2600 THD measurement**

Dreadbox Erebus

THD = 22.12 %

| | root | 1st | 2nd | 3rd | 4th | 5th |
|---|---|---|---|---|---|---|
| Freq | 440 Hz | 880 Hz | 1,320 Hz | 1,760 Hz | 2,200 Hz | 2,660 Hz |
| dBV | 7.37 | -13.39 | -6.76 | -20.98 | -26.49 | -34.76 |
| Volts | 2.336 | 0.214 | 0.459 | 0.089 | 0.047 | 0.018 |

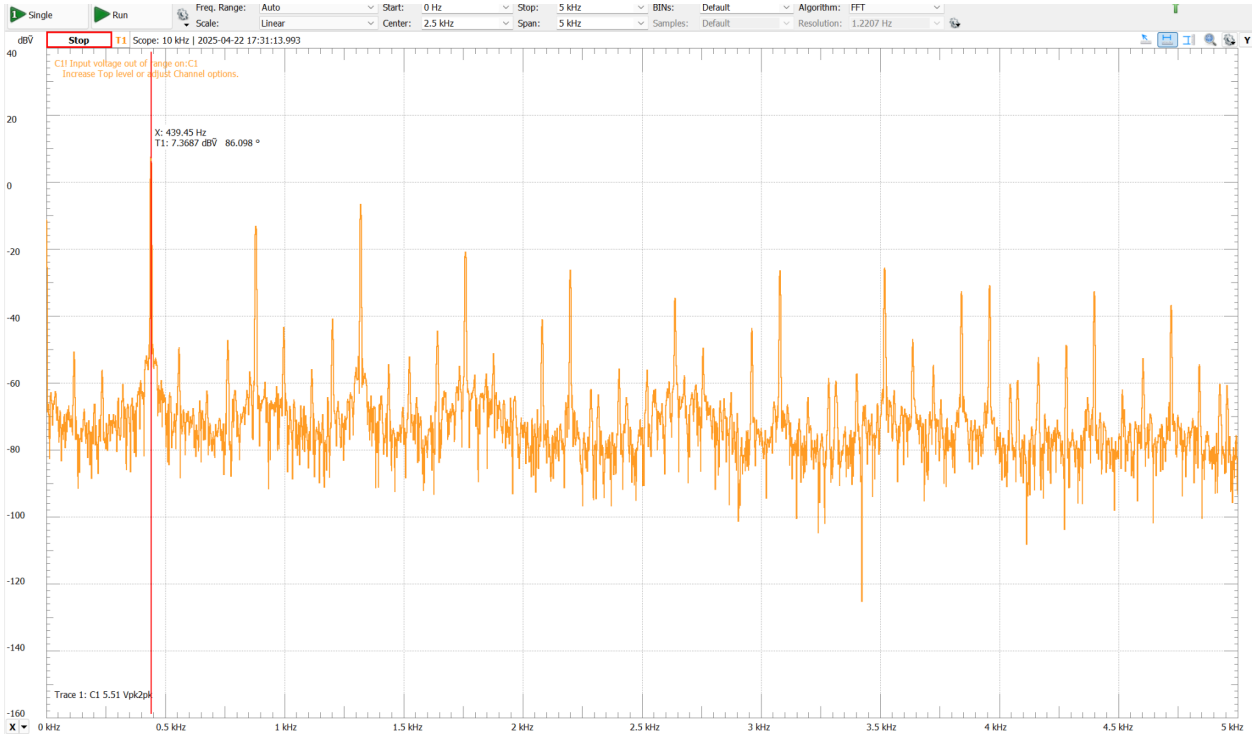


**fig. 6.14: Dreadbox Erebus THD measurement**

<u>Pure Sine</u>

THD = 0.60%

| | root | 1st | 2nd | 3rd | 4th | 5th |
|---|---|---|---|---|---|---|
| Freq | 440 Hz | 880 Hz | 1,320 Hz | 1,760 Hz | 2,200 Hz | 2,660 Hz |
| dBV | -7.8 | -74.73 | -58.83 | -61.49 | -56.42 | -67.76 |
| Volts | 0.407 | 0.000 | 0.001 | 0.001 | 0.002 | 0.000 |


**fig. 6.15: Pure Sine THD measurement**

Arturia MiniBrute Reference

THD = 2.23%

| | root | 1st | 2nd | 3rd | 4th | 5th |
|---|---|---|---|---|---|---|
| Freq | 440 Hz | 880 Hz | 1,326 Hz | 1,778 Hz | 2,200 Hz | 2,664 Hz |
| dBV | -4.86 | -40.15 | -46.17 | -58.11 | -44.65 | -70.85 |
| Volts | 0.571 | 0.010 | 0.005 | 0.001 | 0.006 | 0.000 |

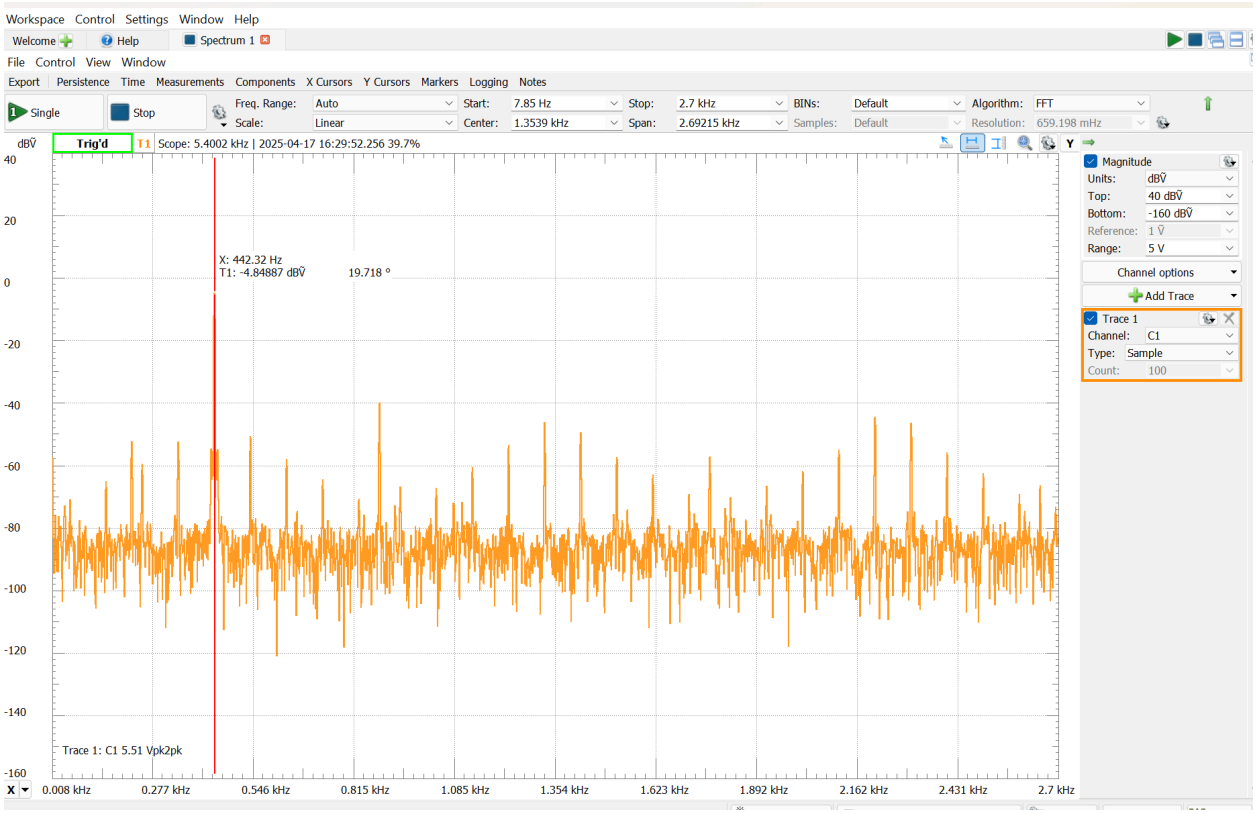


**fig. 6.16: Arturia Minibrute THD measurement**

Interestingly, even the pure sine wave generator produced a THD of 0.60%, suggesting a high noise floor or limitations in the Analog Discovery 2 hardware or software resolution. The CHORDS THD measurement, though above ideal, falls within the range of consumer-grade synthesizers, particularly analog or hybrid systems where distortion can be musically desirable [2], [12].

**Conclusion**: Although the CHORDS system did not meet the <1% THD target, the results remain consistent with many commercial synthesizers currently in use. The relatively high distortion observed across all devices, including a pure wave source, suggests measurement limitations inherent to the Analog Discovery 2 hardware and environmental noise. Additionally, the use of a low-cost I²S DAC module likely contributed to a higher noise floor and non-linearities in the output stage. Future improvements could involve higher-resolution DACs or external audio interfaces to minimize distortion and improve signal fidelity [6], [13].

### 8. FT 4: Filter Module:

**Objective:** To verify the functionality of the CHORDS filter module, specifically its ability to operate as a voltage-controlled low-pass filter (VCF) with a variable cutoff frequency ranging from 20 kHz to 0 Hz. The cutoff frequency is dynamically adjusted using an analog potentiometer interfaced with the RP2350 microcontroller. This allows the user to shape the frequency spectrum of the signal in real-time, a core feature in subtractive synthesis [2].

**Set-up:** The test configuration included:

- The oscillator module produces a stable sine wave signal at a fixed input frequency.
- The filter module, connected via DMSP to synchronize with the global control structure.
- A potentiometer attached to the filter module to adjust the cutoff frequency and Q (resonance).
- Oscilloscope probes connected at:
    - The input to the filter module
    - The output of the filter module

The global clock was initialized to 5 MHz and distributed to all modules via the CHORDS Busboard. Communication between the microcontrollers was verified using standard DMSP frame headers to ensure time alignment and synchronized data interpretation [4].

**Procedure:**

- Initialize the global DMSP clock and verify signal synchronization across modules.
- Load the DMSP protocol onto the RP2350 microcontroller in the filter module.
- Use the signal generator to produce a 440 Hz sine wave as the test input.
- Slowly sweep the potentiometer and observe the resulting frequency response at the filter output on the oscilloscope.
- Repeat the process with varying signal frequencies to confirm consistent filter behavior across the spectrum.

**Results**: The filter module successfully attenuated frequencies above the dynamically set cutoff frequency, responding smoothly to potentiometer adjustments. The observed cutoff frequency range spanned from ~20 kHz down to below 20 Hz, effectively reducing high-frequency content while passing low-frequency components. As the potentiometer was

turned, the filter's Q point (resonance) increased, producing a noticeable peak at the cutoff frequency, a characteristic behavior of second-order low-pass filters [14].

Oscilloscope measurements showed smooth transitions in frequency response with minimal digital stepping or aliasing, confirming adequate resolution in the microcontroller's DAC reading of the potentiometer. These results are consistent with standard digitally controlled analog filter architectures [13].

**Conclusion**: The CHORDS filter module meets its design objectives by providing user-controlled low-pass filtering from 0 Hz to 20 kHz with real-time resonance shaping. The analog potentiometer interface offers intuitive control for musicians to sculpt harmonic content and timbre, enabling dynamic tonal shaping and textural modulation during performance or recording. This test confirms the filter's integration with the CHORDS system and its suitability for subtractive synthesis applications [2].

### 9. FT 5: I2S Module:

**Objective:** The goal of this test is to verify the functionality of the I²S audio output module by confirming its ability to convert digital audio data from the Oscillator module into an audible analog signal. This includes ensuring proper transmission over the I²S protocol, as well as confirming that the output signal is free from distortion, jitter, or aliasing artifacts when monitored on an oscilloscope.

**Set-up:** The test circuit included the following components:

- An I²S-compatible DAC connected to the RP2350 microcontroller's I²S peripheral,
- A signal source (Oscillator module) generating a stable waveform at 440 Hz,
- A headphone output jack mounted on the module PCB for analog monitoring,
- An oscilloscope with probes connected to both the digital input and the analog output of the I²S module,
- A 3.3V regulated power supply used to power the digital and analog components.

The modules were synchronized via the CHORDS global clock, operating at 5 MHz, and

communication between them was established through the DMSP protocol. The I²S interface was configured for standard 24-bit stereo audio at 44.1 kHz sample rate, a widely accepted format in consumer and professional digital audio systems [6].

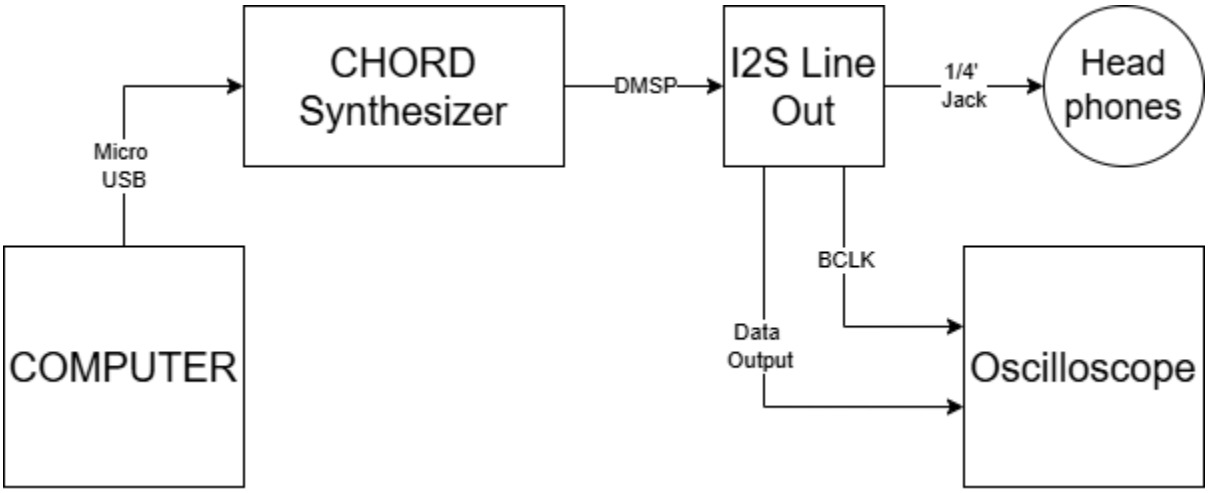


**fig. 6.17: I2S Module Test setup**

**Procedure:**

1. Power the oscillator and I²S modules using a 3.3V supply.
2. Verify the global DMSP clock is synchronized across all connected modules.
3. Load the DMSP frame structure onto the microcontroller to enable proper addressing and data parsing.
4. Begin waveform generation from the oscillator (e.g., sine wave at 440 Hz).
5. Monitor the analog output via headphones and verify audible playback.
6. Simultaneously observe the waveform on the oscilloscope for fidelity, jitter, and distortion analysis.

**Results**: The I²S module successfully converted digital audio data from the oscillator into an audible analog signal. The waveform observed on the oscilloscope matched the expected profile of the oscillator's sine wave output. No visible phase jitter, unexpected noise, or

waveform distortion was observed, indicating proper clock recovery and stable serial transmission, essential for high-fidelity digital audio output [6], [13].

The audible output was clean and continuous when monitored through headphones, confirming the correct function of the DAC and the analog driver stage. These results also validate the I²S timing configuration and the compatibility between the RP2350's I²S peripheral and the selected DAC.

**Conclusion**: This test confirms that the I²S audio output module performs as intended, converting digitally synthesized waveforms into analog signals with high accuracy and no observable degradation. With this module operational, subsequent tests can begin on audio quality, including measurements of signal-to-noise ratio (SNR), total harmonic distortion (THD), and frequency response. The I²S protocol has proven reliable in this application, consistent with its widespread use in embedded audio systems for low-jitter and low-latency playback [6].

## 7. Challenges and Risks

Challenges

1. **Noise:** CHORDS is intended to provide clean digital waveforms for the Musician
    a. Factors: Signal being sent to separated modules, cheap components
    b. Plan: Testing Signal-to-Noise Ratio, applying filters to remove unwanted noise
    c. Outcome: SNR = -43.28dBV. Harmonics were lost in the noise floor raising total harmonic distortion (THD)

2. **Clock Distribution:** All of the modules need to be synchronized
    a. Factors: Improper clock frequency, inconsistent distribution of DMSP
    b. Plan: Test Global Clock output individually and collectively
    c. Outcome: Global Clock was synced ever 32 bits with no measurable drift after 2 hours

3. **Module Latency:** Latency is a major concern for all digital audio systems
    a. Factors: Each CHORDS module will introduce latency into the patch
    b. Plan: Testing latency and improving connections between modules can minimize latency
    c. Outcome: Our DMSP helped sync the modules with an average system latency of 13.02ms

Risks

1. **Latency between input and output:** Rating = 15; Consequence = 3, Likelihood = 5

   a. Factors: Too many modules and Inefficient implementation of DMSP or DSP algorithms
   b. Contingency Plan: Optimize each module to reduce latency added to patch, Ensure Modules meet ER 4.
   c. Outcome: Latency was kept minimal at an average of 13.02ms threw the entire system

2. **Signal Integrity of DMSP and Global CLK Bus:** Rating = 15; Consequence = 5, Likelihood = 3
   a. Factors: cable type and length, distances between modules or latency causing synchronization issues
   b. Contingency Plan: consistent hardware implementation for bus and DMSP connections
   c. Outcome: DMSP was a success, and signal integrity was maintained

3. **Running out of RAM or ARM Core overload:** Rating = 10; Consequence = 5, Likelihood = 2
   a. Factors: Buffer size, DSP Algorithm implementation,
   b. Contingency Plan: Reduce buffer size and optimize code, or reduce module feature
   c. Outcome: Did not encounter overloading issues

# 8. Ethics of the Engineering Profession and Our Project

Ethical design in engineering focuses on creating products that prioritize safety, sustainability, accessibility, and societal well-being while minimizing harm to people and the environment. For CHORDS, ethical design is central to its development. By providing an affordable and versatile and reprogrammable tool for musicians, we aim to democratize access to hardware modular synthesizers. The ethical use of technologies such as microcontrollers and MIDI inputs ensures that our product is both reliable and non-invasive, adhering to principles of safe and responsible innovation.

Sustainability is a key consideration in our project. While the synthesizer is built using electronic components, which can have environmental implications, the modular design promotes longevity and reduces electronic waste. If a module fails, it can be repaired or replaced individually, avoiding the need to discard the entire system. Additionally, the use of recyclable materials for the casing and the encouragement of responsible disposal practices further mitigate environmental impact. However, the project does not currently incorporate biodegradable materials, a potential area for future improvement.

Accessibility is another important aspect of our ethical design. The interface, including knobs, sliders, and buttons, is designed to be intuitive, ensuring ease of use for musicians of varying skill levels. This is also why we must clearly define the DMSP as it is important for any user, regardless of musical skills, to understand and utilize CHORDS effectively. Furthermore, incorporating visual feedback mechanisms helps users with hearing impairments engage with the product.

Finally, potential harms have been carefully analyzed. The synthesizer operates at low power levels, eliminating risks associated with high-power RF signals or other hazardous emissions. By addressing these ethical considerations, our project aligns with the engineering profession's commitment to creating technologies that improve lives while respecting societal and environmental boundaries.

# 9. Appendix

### Budget/Parts List

Our project is based around creating a low cost synthesizer. We found the PICO-2 RP2350 was the ideal microcontroller for price and capability. The bulk of our budget is related to that microcontroller as most of the modules we intend to build will be through it. As well there are some other components such as IC (6N138) which is used in an optocoupler circuit for the MIDI controller input module. The Audio output is run through an I2S board ensuring the proper sample rate. The MIDI controller is something most musicians would have, it is added to the budget due to the fact it is necessary for controlling the synthesizer, but not necessarily for production.

**Table IV: Synthesizer bill of materials**

| Part/ Quantity | Price | Description | Link | Test | ER# |
|---|---|---|---|---|---|
| PICO-2 (RP-2350) | $70 (x14) | Board used for Global Clock, DMSP, and Modules | https://www.adafruit.com/product/6006 | ST.1<br>FT.1<br>FT.2 | ER.1<br>ER.2<br>ER.3<br>ER.5 |
| I2S output Board | ~$10 | Audio output module | https://www.adafruit.com/product/3678 | ST.1 | ER.1<br>ER.2<br>ER.3 |
| MIDI Controller | ~$100 (Provided) | MIDI keyboard that will trigger input data | https://www.amazon.com/Nektar-SE49-49-Key-Controller-Keyboard/dp/B01MF9EJPG?th=1 | ST.1 | ER.4 |
| IC (6N138) | $0.96 | MIDI Input module | https://www.digikey.com/en/products/detail/lite-on-inc./6N138/1969179 | ST.1<br>FT.1 | ER.4 |
| PCB | $39.20 (x25) | Module architecture | https://cart.jlcpcb.com/quote?spm=Jlcpcb.Homepage.1006 | | |
| ¼’ Jack | $0.60(x14) | Custom busboard PCB | | | |
| Potentiometers | $1.20 (x50) | Front Panel Parameter | Potentiometers | | |
| 2-1825910-7 (Button) | $5.48 (x50) | Front Panel Parameter | https://www.digikey.com/en/products/detail/te-connectivity-alcoswitch-switches/2-1825910-7/1632528 | | |
| Switch | $10.21 (x35) | Power Control | https://www.digikey.com/en/products/detail/same-sky-formerly-cui-devices/SL | | |

| | | | | | |
|---|---|---|---|---|---|
| | | | W-864574-5A-RA-N-D/24399231 | | |
| MIDI Socket | $1.33 | Socket to plug in MIDI controller | https://www.amazon.com/dp/B0978STN2X?ref=ppx_yo2ov_dt_b_fed_asin_title&th=1 | | |
| IC (CD4053BE) | $10.94 (x25) | PCB Design | https://www.digikey.com/en/products/detail/texas-instruments/CD4053BE/67309 | | |
| 1N519 Diode | $4.87 (x35) | PCB Design | https://www.digikey.com/en/products/detail/stmicroelectronics/1N5819/1037326 | | |
| Header Pins | $7.99 | PCB Design | Header Pins | | |
| Ribbon Cable | $6.99 | PCB Connection | Ribbon Cable | | |
| Bus Board | $13.00 (x10) | DMSP output board | https://cart.jlcpcb.com/quote?spm=Jlcpcb.Homepage.1006 | | |
| Lithium Ion Battery (3.7V) | $24.50 | Battery Power | https://www.adafruit.com/product/353 | | |
| Battery Charger | $19.95 | Battery Power | https://www.adafruit.com/product/2465 | | |
| 3D printing | $9 | For Case, knobs, and sliders | https://library.sonoma.edu/create/makerspace | | |
| M3 Heat Inserts | $7.39 | Heat inserts for Screwing Modules to Case | https://www.amazon.com/dp/B01DBOBRHQ?ref=ppx_yo2ov_dt_b_fed_asin_title | | |
| M3 x 10mm Screws | $8.76 | Securing Modules to Case | https://www.amazon.com/dp/B08H2HSPQD?ref=ppx_yo2ov_dt_b_fed_asin_title | | |
| ⅛” Acrylic sheet | $8.25 | Clear backplate for the synthesizer | https://www.amazon.com/dp/B0BBQ8B41F?ref=fed_asin_title | | |
| Total Cost | | | | | $331.35 |

## Project Schedule

The schedule for this project revolves around Digital Modular Synth Protocol (DMSP) and Global Clock control. These two aspects will be the first thing we start on and continue to edit and improve as we add to the project. Once we begin development on the DMSP and Global Clock we intend to work on the multiple modules within the synthesizer. These can be developed simultaneously since there are no new design parameters being created for this project. Our oscillators, filters, envelopes, and other modules will be based on existing designs. These steps in the process should be the fastest. After the new modules are created, our next goal is to connect these modules together. This stage will revolve around debugging code and troubleshooting hardware issues. This stage is where we would improve our clock, DMSP, and sound output quality. Once all the modules are fully functional and we are able to connect the system together, we intend to design an aesthetic case and modules to make the system appealing to customers.

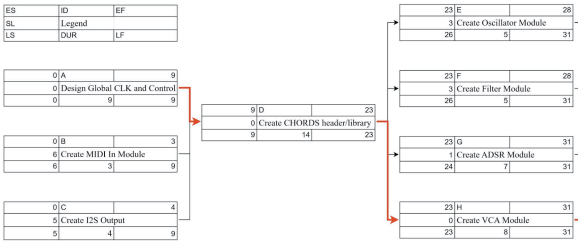


**Figure 7.1: Winter Gantt Chart**

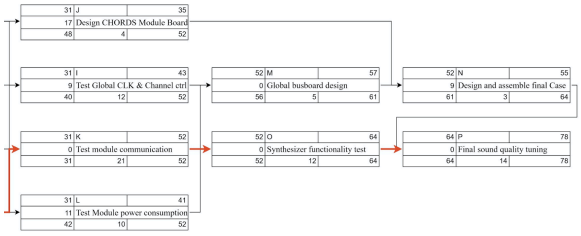


**Figure 7.2: Spring Gantt Chart**

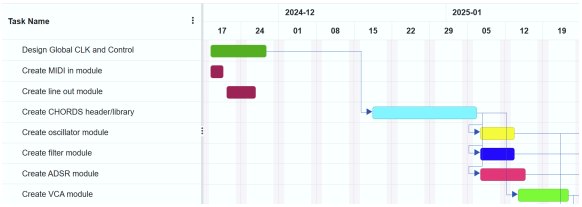


**Figure 7.3: Winter Schedule**

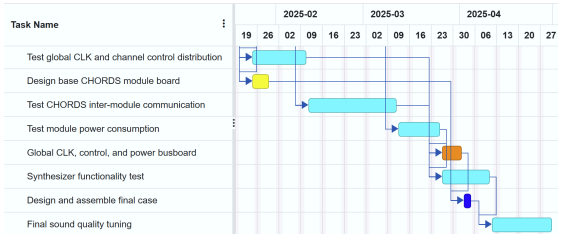


**Figure 7.4: Spring Schedule**

## 10. References

[1] P. D. Manning, *Electronic and Computer Music*. Oxford University Press, 2013.

[2] C. Roads, *The Computer Music Tutorial*. MIT Press, 1996.

[3] B. Hutchins, “Polyphony and voice allocation in modular synthesizers,” *J. Audio Eng. Soc.*, vol. 61, no. 7/8, pp. 507–519, 2013.

[4] GigaDevice Semiconductor Inc., “GD32F103xx Datasheet (RP2350 Equivalent),” [Online]. Available: https://www.gigadevice.com/datasheet/gd32f103xx-datasheet/

[5] A. W. Moore, *MIDI for the Professional*. A-R Editions, 2010.

[6] Texas Instruments, “Understanding the I2S Audio Interface,” Application Report SLAA449B, Jan. 2010. [Online]. Available: https://www.ti.com/lit/an/slaa449b/slaa449b.pdf

[7] VCV Rack, “Open-source virtual modular synthesizer,” VCV, [Online]. Available: https://vcvrack.co. [Accessed: May 19, 2025].

[8] Behringer, “Neutron - Paraphonic Analog and Semi-Modular Synthesizer,” [Online]. Available: https://www.behringer.com/product.html?modelCode=0718-AAO. [Accessed: May 19, 2025].

[9] Doepfer Musikelektronik, “A-111-4 Quad VCO,” [Online]. Available: https://www2.doepfer.eu/en/item/a111-4. [Accessed: May 19, 2025].

[10] Erica Synths, “Black Sequencer,” [Online]. Available: https://www.ericasynths.lv/shop/eurorack-modules/by-series/black-series/black-sequencer/. [Accessed: May 19, 2025].

[11] Teenage Engineering, “OP-Z multimedia synthesizer and sequencer,” [Online]. Available: https://teenage.engineering/store/400/. [Accessed: May 19, 2025].

[12] AudioTechnology, “Mixing Synth Pop Synths,” AudioTechnology Magazine, [Online]. Available: https://www.audiotechnology.com/tutorials/mixing-synth-pop-synths. [Accessed: May 19, 2025].

[13] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, 4th ed. Pearson, 2006.

[14] A. V. Oppenheim and A. S. Willsky, *Signals and Systems*, 2nd ed. Prentice Hall, 1997.

[15] R. F. Moore, "The Role of Latency in Musical Performance," *Computer Music Journal*, vol. 25, no. 1, pp. 34–41, 2001.