# ECE 445

## Senior Design Laboratory

## Updated Project Proposal

---

# String/Drum Synthesizer

---


**Team #21**

Abhi Nayak (arnayak2@illinois.edu)

Joel Schurgin (joelbs3@illinois.edu)

**TA:** Manvi Jha


October 3, 2024

# Contents

# 1. Introduction

**Section Overview**

In this section, we will introduce the problem our project seeks to address, providing a clear problem statement supported by detailed examples to highlight its significance. We will then outline our proposed solution, offering a high-level overview of how it will be implemented. Additionally, a visual aid will illustrate the context of our solution, showing its interaction with relevant external systems. Finally, we will present a list of high-level requirements that our project must meet to effectively solve the problem.

## 1.1 Problem

A musical artist brings with them a sense of how to evoke certain emotions or create particular atmospheres, but may lack the technical knowledge to bring it to life. Sound design can be challenging as it involves crafting the unique timbres and textures that define a piece of music, which can be both an art and a science. The learning curve is steep, not only because it requires a grasp of complex technical tools but also because it demands a creative intuition that isn't easily taught. Musicians and producers must learn to navigate a vast array of software and hardware tools, each with its own set of knobs, sliders, and buttons. Understanding how to shape sounds means diving deep into the physics of sound waves, the principles of digital signal processing, and the subtleties of different synthesis methods. This process can feel overwhelming, especially for beginners who might feel lost amidst the jargon and the sheer number of options.

## 1.2 Solution

Our proposal is to build a synthesizer. A synthesizer is a machine that is capable of producing a range of sounds when given a simple key or noise pulse. The range of sounds are controlled by a set of parameters such as filter cutoffs, oscillator types, low frequency oscillators, etc. The proposed solution is two-fold. The first part of the project is to build emotive knobs that feature descriptive ranges such as soft to aggressive or thin to thick. These knobs would also control the internal parameters of the system to achieve the desired effects on the input sound. The other part is to create a system that records a sound and finds the best way to set the synthesizer's internal parameters to recreate it. The intended work flow of this device for a musical artist is to record a sound, tweak the knobs, and play that sound by pressing a key on a MIDI keyboard. It is up to the musician to record the sound into their own recording software for further manipulation.


Figure 1: Traditional Synthesizer

## 1.3 Visual Aid

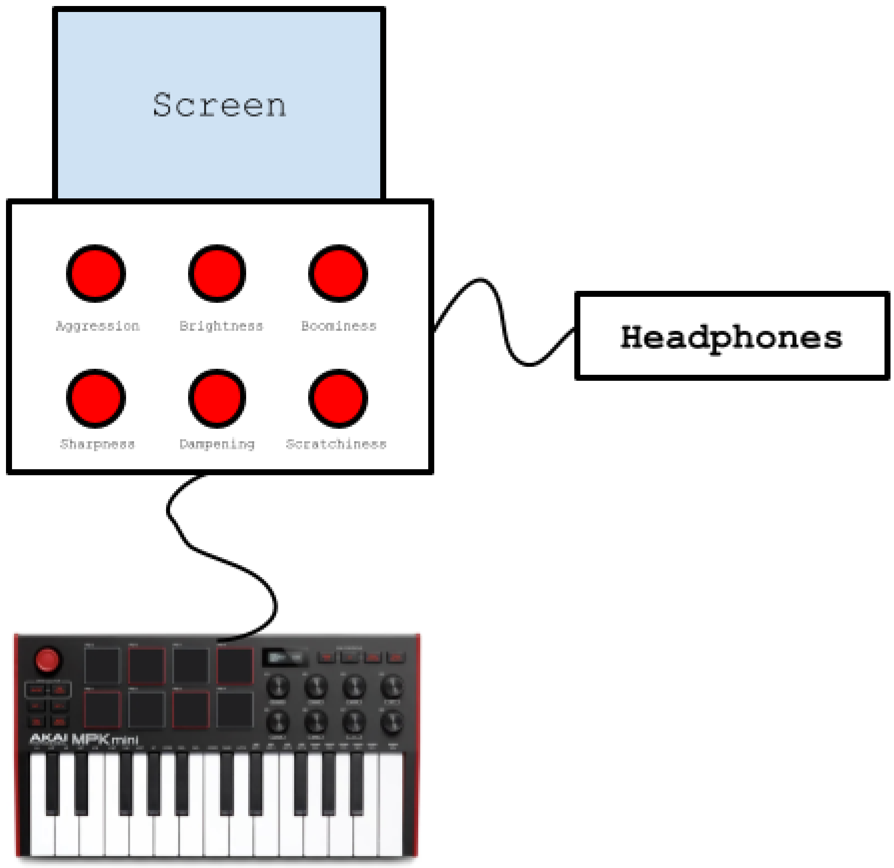


**Figure 2: High-Level Visual of our String/Drum Synthesizer**

## 1.4 High Level Requirements

1) Potentiometers are able to control synthesizer parameters with at least 8-bit resolution.
2) Notes are played at the pitch of the key pressed on the USB MIDI keyboard with latency faster than 10ms (3).
3) 114+/-1% ohm output impedance and sound can be heard with headphones (2).

# 2. Design

### Section Overview

In this section, we will provide a detailed breakdown of our design through several key components. First, we will provide a physical design of our project with labeled dimensions and included parts. Next, we will present a block diagram that divides the design into subsystems, labeling voltages and data connections to illustrate their relationships. Following this, we will offer a brief overview of each subsystem's function and explain how they interconnect. Each subsystem will be discussed in its own paragraph to clarify its role within the overall design. We will then delve into the specific requirements for each subsystem, detailing how each contributes to meeting the high-level project requirements and defining any critical interfaces and dependencies. Finally, we will conduct a tolerance analysis to identify any potential risks to the project's success, using a synthesizer simulation to demonstrate the feasibility of critical components.

## 2.1 Physical Design

Shown below is the physical design for our project, with labeled dimensions and components. Our PCB will be held inside of this enclosure.

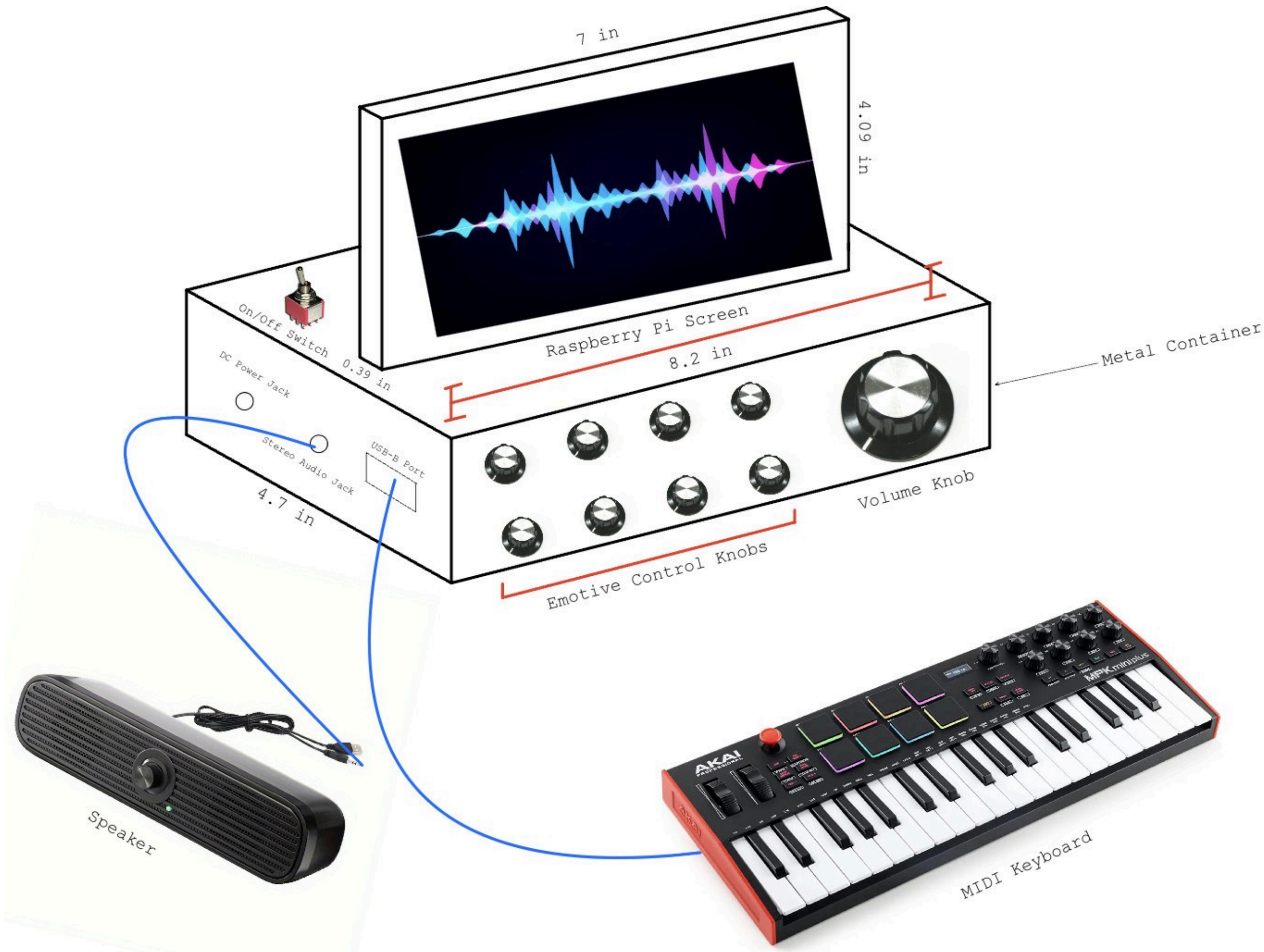


**Figure 3: Physical Synthesizer Design**

# 2.2 Block Diagram

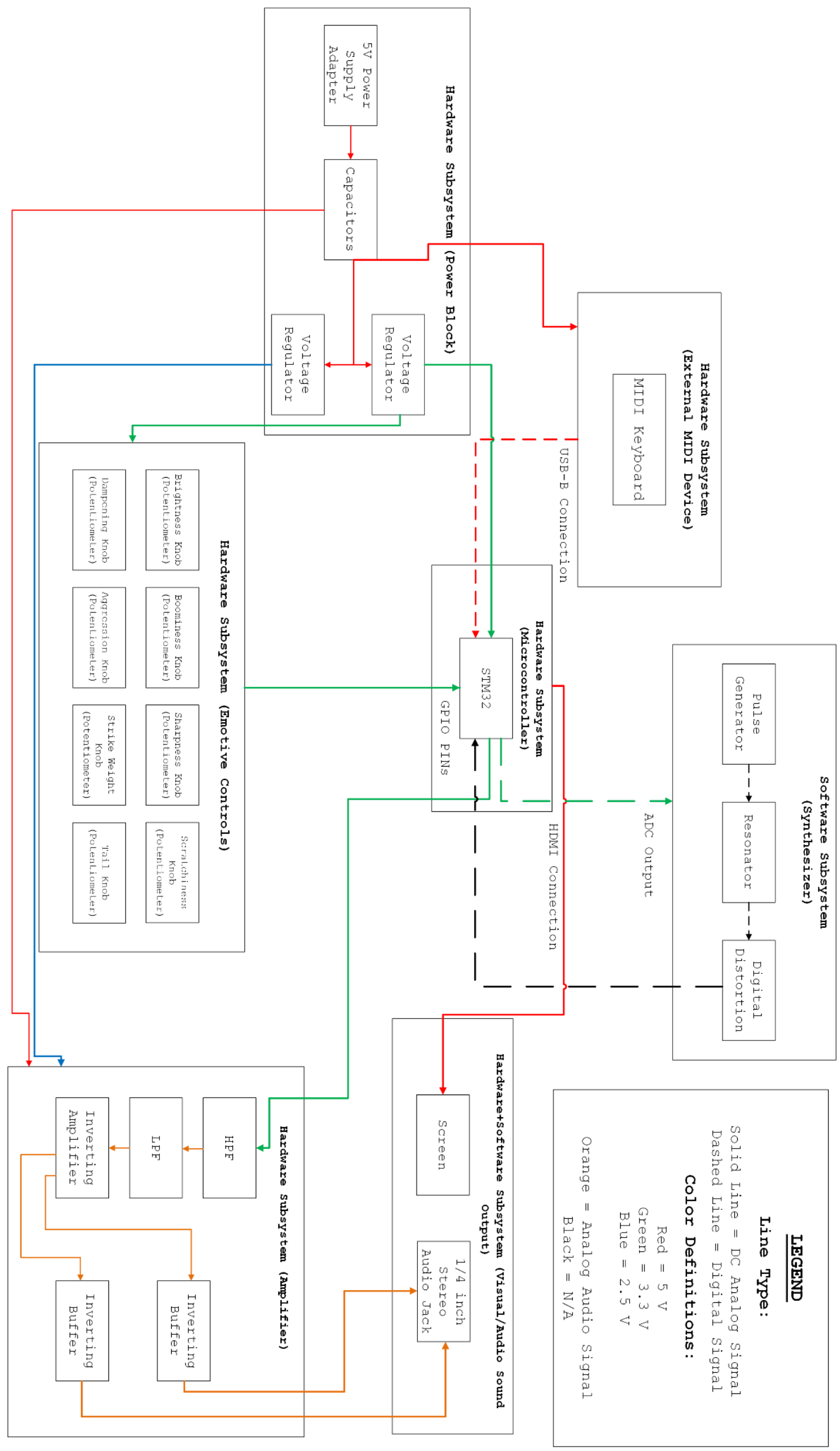


**Figure 4: High-Level Block Diagram**

## 2.3 Subsystem Overviews and Requirements

### 2.3.1 Power Block Subsystem, Hardware

#### Overview

The power block provides DC power to the other subsystems. It begins with an adapter that plugs into a wall outlet and provides 5V DC and at least 300 mA. This voltage is stabilized with parallel capacitors in order to achieve 5V +/- 0.5% DC with at least 300 mA to the amplifier, screen, and USB port continuously over time.

After the parallel capacitors, there will be an LM1117 voltage regulator set to 3.3V. This stage must be able to supply 3.3V +/- 0.5% DC at 150 mA to the STM32 processor continuously over time.

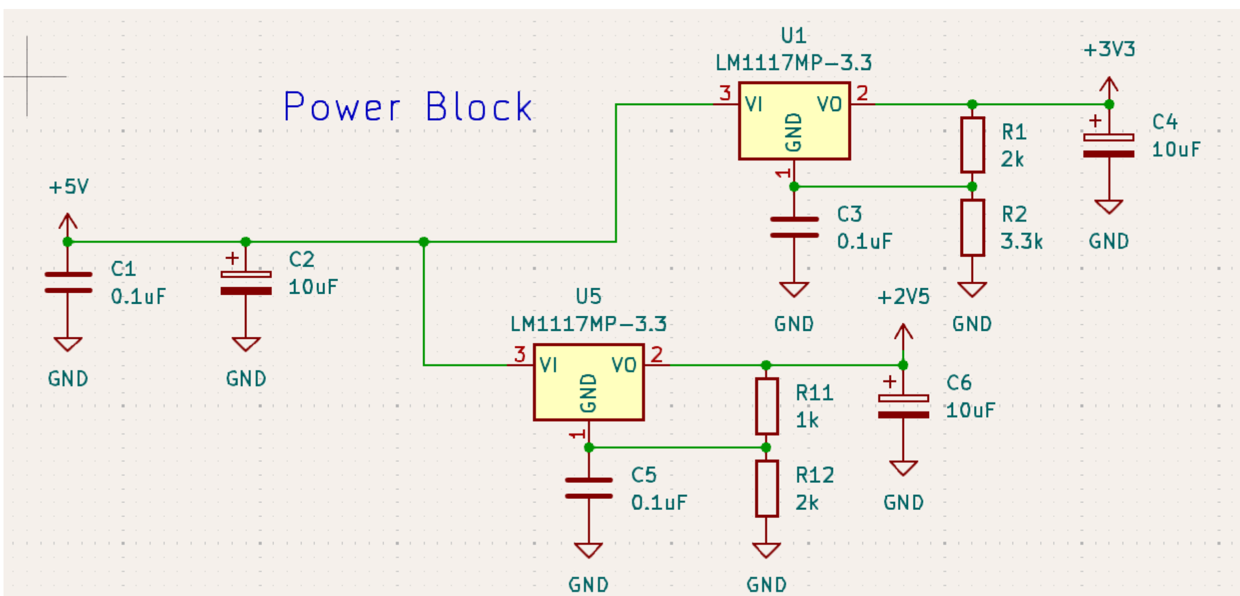


**Figure 5: Power Block Subsystem Circuit Schematic**

The above circuit diagram details our Power Block subsystem circuit, consisting of a 5V input coming from the 5V DC Power Supply adapter, a voltage stabilizing circuit, and finally, two voltage regulator circuits. In the voltage stabilizing circuit, there are two key components — an electrolytic capacitor and a ceramic capacitor. The electrolytic capacitor (C1) is used to stabilize the input voltage whereas the ceramic capacitor (C2) is being used to filter out high frequency noise. By placing these components in parallel, this subsystem as well as the others will be able to receive a steady 5V voltage supply. Next, we have the voltage regulator circuit, consisting of a LM1117MP-3.3 chip, a bypass capacitor, a ceramic capacitor, and two resistors in series. The bypass capacitor (C3) in this circuit is used to improve ripple rejection and the ceramic capacitor (C4) is used to stabilize the output voltage. The overall circuit functions as a voltage step down circuit, where $V0 = 1.25 (1+(R2/R1))$. Knowing this equation, we have selected specific values for R1 and R2 such that the output voltage will be 3.3V, which is the voltage required to power the STM32 microcontroller subsystem. For the same purpose, we have another voltage regulator circuit where we have selected specific values for R11 and R12 such that the output voltage will be 2.5V, which is the voltage required to power the Amplifier subsystem.

**Table 1: Power Block Subsystem – Requirements & Verification**

| Requirements | Verification |
|---|---|
| 1) Provide 5V +/- 0.5% from the adapter.<br>2) 5V output can operate with loads ranging from 0-300mA while maintaining requirement 1.<br>3) Provide 3.3V +/- 0.5% from the adapter.<br>4) 3.3V output can operate with loads ranging from 0-150mA while maintaining requirement 1. | 1) Measure the output voltage with an oscilloscope and ensure that the voltage remains within +/- 0.5% of 5V.<br>2)<br>a) Connect 5V output to a current meter and a variable resistor in series.<br>b) Change the load resistance, using the current meter to verify the current range and an oscilloscope to verify that requirement 1 remains satisfied.<br>3) Measure the output voltage with an oscilloscope and ensure that the voltage remains within +/- 0.5% of 3.3V.<br>4)<br>a) Connect 3.3V output to a current meter and a |

| | |
|---|---|
| | variable resistor in series.<br>b) Change the load resistance, using the current meter to verify the current range and an oscilloscope to verify that requirement 3 remains satisfied. |

## 2.3.2 Synthesizer Subsystem, Software

**Overview**

This subsystem will generate sounds using the Karplus Strong algorithm which features a pulse generator and an echo chamber (modeled with digital delay, feedback, and filtering). When a note is played on a USB MIDI keyboard, this component must be able to change the pitch based on which key is pressed. The algorithm's parameters must be controllable using potentiometers, but several parameters can be determined as a function of one knob.

We chose the Karplus Strong algorithm to reduce the computational load, but we are modifying certain aspects to allow for a greater range of sounds.

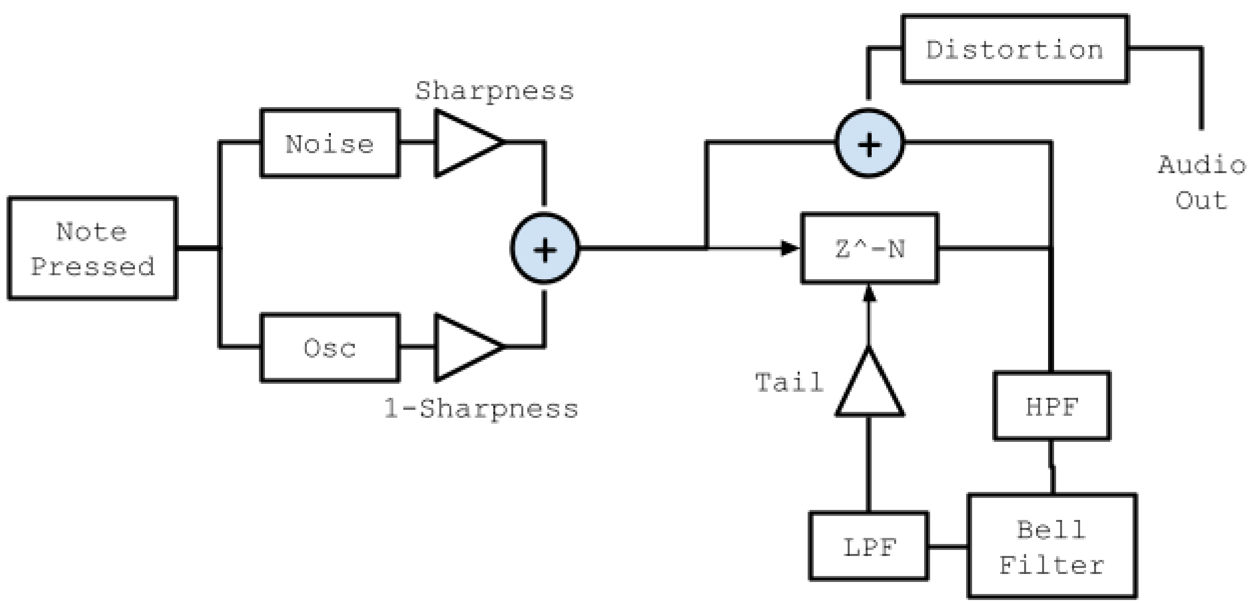

**Figure 6: Modified Karplus-Strong Block Diagram**

The pulse that we will use is a blend of noise and sine wave oscillator which are added together. In addition, there will be some distortion applied on the output. The echo chamber will use 3 IIR filters: highpass, lowpass, and a bell. The choice to use IIR filters is to lower the amount of computation.

The synthesizer's parameters will be controlled by the emotive knobs subsystem. The audio will be converted to a PCM wave using the STM32 processor's I2S system. The PCM wave will be sent to the amplifier subsystem.

**Table 2: Synthesizer Subsystem – Requirements & Verification**

| Requirements | Verification |
|---|---|
| 1) Read MIDI data from the USB MIDI keyboard when keys are pressed.<br>2) Audio buffer outputs sound.<br>3) Generates pulse at correct time within 10ms.<br>4) Resonator resonates at the correct pitch.<br>5) Verify that Digital Distortion distorts signal. | 1) Plug in the keyboard, press a key and print pitch on screen as an integer. Adjacent notes should differ by 1.<br>2) Write a sine wave to buffer and check output with an oscilloscope.<br>3) Record a video that shows the key pressed and records the sound of the pulse. Verify that they are within 10ms.<br>4) Connect output to oscilloscope and verify pitch is correct.<br>5)<br>a) Turn distortion off and press a key on the keyboard.<br>b) View the spectrum with an oscilloscope.<br>c) Repeat b with distortion on. Should see more high frequencies. |

## 2.3.3 Emotive Controls Subsystem, Hardware

The emotive knobs subsystem provides a way to control the synthesizer parameters using more descriptive names for the knobs. We made a demo of the synthesizer algorithm using a digital audio workstation in order to play with parameters and determine the names of the emotive knobs.

The physical emotive knobs are connected to potentiometers that range between 3.3V and ground. As the user selects values for each of the knobs, their respective voltages are instantaneously sent to the GPIO Pins of the STM32 Microcontroller. Then, a software program will rapidly retrieve these voltages one by one and pass them through one of the ADC's on the Microcontroller. These now digital values will populate a knob_value buffer, which will be used by the Synthesizer subsystem to make manipulations to the sound the user is building.

**Table 3: Emotive Controls Subsystem – Knob Parameters**

| **Emotive Knob Names** | **Corresponding Synth Parameters** | **Param Values: Knob Left** | **Param Values: Knob Center (if applicable)** | **Param Values: Knob Right** |
|---|---|---|---|---|
| Aggression | Distortion | Low gain | | High gain |
| Brightness | Low pass filter cutoff | 20 Hz | | 20 kHz |
| Boominess | High pass Cutoff | 20 kHz | Note pitch | 20 Hz |
| Dampening | Bell Q | High Q | | Low Q |
| Scratchiness | Length of pulse | 0 | 1/(Note pitch) | 4/(Note pitch) |
| Sharpness | Mix between pulse noise and | Fully oscillator | | Fully noise |

| | | | | |
|---|---|---|---|---|
| | oscillator | | | |
| Strike weight | Pulse oscillator pitch | 20 kHz | | 20 hz |
| Tail | Feedback | 0.0 | | 1.0 |

Notes:

- Note pitch is determined by USB keyboard input.
- Bell gain is fixed and should be set to attenuate the signal.
- Bell frequency will be fixed at 3.3 kHz.

**Table 4: Emotive Controls Subsystem – Requirements & Verification**

| Requirements | Verification |
|---|---|
| 1) Processor correctly reads the value of all potentiometers. | 1)<br>a) Turn one potentiometer and print the value on a computer screen/debugger.<br>b) Repeat with the other potentiometers. |

## 2.3.4 Amplifier Subsystem, Hardware

### Overview

The amplifier will modify the synthesizer’s signal provided by the Microcontroller Subsystem so it can be heard with headphones. It receives the audio from the I2S system on the STM32 processor in the form of a PWM wave which must be filtered and amplified and have correct output impedance of 114+/-1% Ohms.

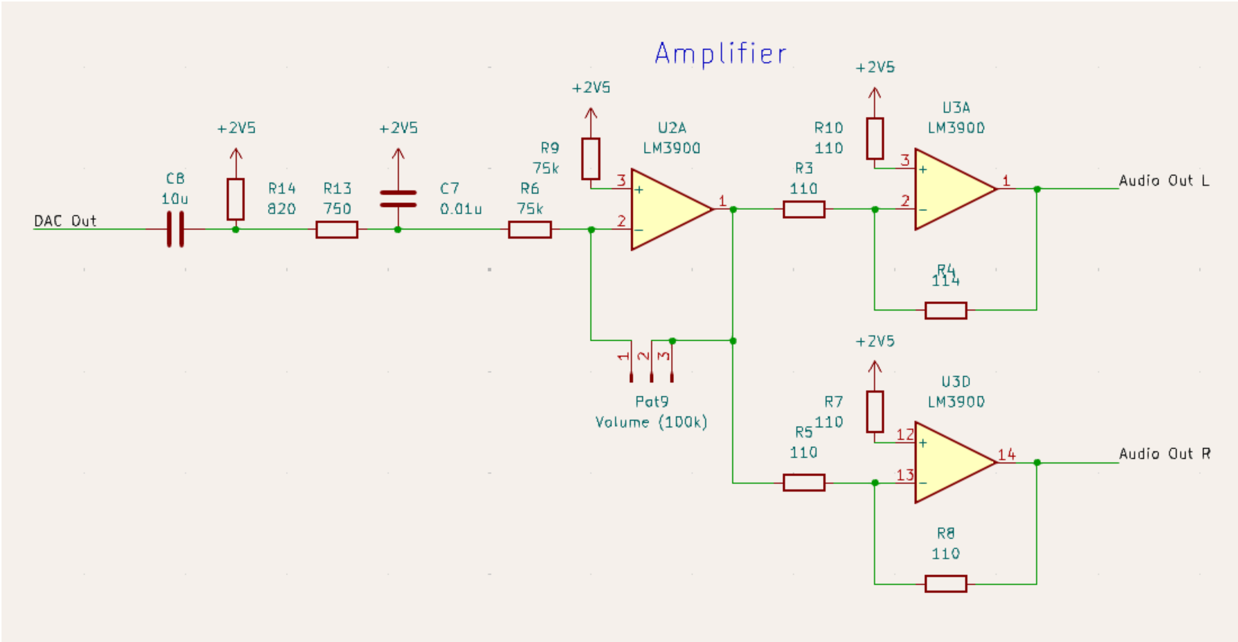


**Figure 7: Amplifier Subsystem Circuit Schematic**

From left to right, the amplifier circuit starts with high and low pass filters, which are amplified then split into left and right signals that are connected to the headphone jack. Since the two rightmost op-amps are in the inverting buffer configuration, their output impedances are equal to R4 and R8 (more detail in the Tolerance Analysis section).

**Table 5: Amplifier Subsystem – Requirements & Verification**

| Requirements | Verification |
|---|---|
| 1) Impedance is 114+/-1%<br>2) Low pass filter converts PCM wave to audio. | 1)<br>a) Attach a waveform generator to the output of the low pass filter.<br>b) Attach a multimeter to measure current and an oscilloscope to measure voltage.<br>c) Find impedance using Z=V/I<br>2)<br>a) Play note on synthesizer and attach oscilloscope to input |

| | |
|---|---|
| | and output of the low pass filter. Compare input and output to verify that the output is not a PCM wave. |

### 2.3.5 Visual/Audio Sound Output Subsystem, Hardware+Software

#### Overview

This subsystem should display a visual representation of the sound being created on the Raspberry Pi Screen (7 inch monitor external display and also output the audio of this sound using a ¼ stereo audio jack. The screen will be directly connected to the STM32 processor via HDMI. We will have a software program written on the processor that will retrieve the waveform of the signal produced by the Synthesizer Subsystem. This waveform output will then be displayed on the screen in real-time as manipulations are made to the sound. Simultaneously, the output of the Amplifier Subsystem will be connected to the audio jack in this subsystem, which will allow users to plug in headphones or a speaker to be able to hear the sound they are building.

The main purpose of this subsystem is to enhance the usability of the synthesizer by visually displaying the effects of each emotive knob, making it easier for users to identify how different settings affect the overall sound. This would bridge the gap between technical sound parameters and intuitive control.

**Table 6: Visual/Audio Sound Output Subsystem – Requirements & Verification**

| Requirements | Verification |
|---|---|
| 1) Display accurate sound waveform to the monitor screen.<br>2) Sound waveform updates in real-time on the display after an emotive knob is manipulated.<br>3) Can hear sound with headphones or a speaker. | 1)<br>a) Using the characteristics of the sound being created (such as its frequency and amplitude), generate an expected waveform using Scopy. This will be used as |

our base reference point.

b) Next, write code in our Synthesizer Subsystem that will take the synthesizer’s output and create a visualization of the sound.
c) Display this visualization on the screen via the HDMI connection to the STM32 microcontroller.
d) Cross examine the screen output with the base reference waveform and verify that its frequency and amplitude match exactly.

2)

a) Take a picture of the unmanipulated sound waveform that is currently being displayed on the screen and write down its frequency and amplitude.
b) One by one, turn each emotive control knob and verify that it is not only changing the waveform being displayed, but doing so in the expected manner for that particular knob. At the same time, check if the frequency and amplitude of the sound is changing as the knob is being turned.

3)

a) Plug in headphones or

| | |
|---|---|
| | a speaker into the audio jack.<br>b) Attach a waveform generator to the output of the low pass filter in the Amplifier subsystem.<br>c) Verify that the sound can be heard. |

## 2.4 Tolerance Analysis

### Meeting Amplifier Requirements

The STM32 processor's DAC outputs a PWM wave. On its own, the PWM wave is not ideal to connect directly into headphones for three reasons: magnitude, DC offset, and the waveform's sharp edges.

The DAC outputs a PWM wave with a magnitude of 1.65V since the processor is supplied with 3.3V. Since we have 5V to work with, we could scale the magnitude to 2.5V. We will leave 0.2V of headroom, so the maximum amplifier gain should be $2.2V/1.65V = 1.333\,V/V$. The amplifier gain is set by the volume knob ranging from 0 to 1.333.

To shift the wave properly, we will use a high pass filter with a 3dB cutoff a little below 20Hz, the low end of the hearing spectrum (4). 2.5V will be used as a virtual ground for the filter and remaining stages of the amplifier, which ensures that frequencies of 0Hz correspond to 2.5V.

The sharp edges correspond to high frequencies far beyond the hearing spectrum (>20kHz) (4), so we will filter them out using a lowpass filter with a 3dB cutoff that's a little higher than 20kHz.

The connection between the synthesizer and headphones can be modeled as two transmission lines with separate impedances ($Z_{Headphones}$ and $Z_{Synth}$). Ideally, all of the signal is transmitted ($\tau = 1$), and none of it is reflected ($\Gamma = 0$). In addition, $\Gamma < 0$ is also desirable for the transmission lines because the reflected wave destructively interferes with the incoming wave from the synthesizer. $\Gamma = -0.5$ is the best case scenario because the destructive interference is absolute. We want to avoid $\Gamma > 0$ because the reflected wave will cause unwanted resonance.

$$\frac{Z_{Headphones} - Z_{Synth}}{Z_{Headphones} + Z_{Synth}} \leq 0$$

$$Z_{Headphones} - Z_{Synth} \leq 0$$

$$Z_{Headphones} \leq Z_{Synth}$$

$$\frac{Z_{Headphones} - Z_{Synth}}{Z_{Headphones} + Z_{Synth}} \geq -0.5$$

$$Z_{Headphones} - Z_{Synth} \geq -0.5Z_{Headphones} - 0.5Z_{Synth}$$

$$3Z_{Headphones} \geq Z_{Synth}$$

We are using headphones with $Z_{Headphones} = 38\,\Omega$ (2), so $38\,\Omega \leq Z_{Synth} \leq 114\,\Omega$. $Z_{Synth} \rightarrow 114\,\Omega$ is better because we can support $Z_{Headphones} = 114\,\Omega$ as that would result in $\Gamma = 0$, but with $Z_{Headphones} = 38\,\Omega$, $\Gamma = -0.5$ and $\tau = 0.5$. We will use $Z_{Synth} = 114\,\Omega$ to support a wider range of headphones.

### Modified Karplus-Strong Proof-of-Concept

The most crucial subsystem is the synthesizer algorithm. Using an algorithm that sounds pleasing to the ear is vital to the efficacy of our project, so we created a demo of the modified Karplus-Strong algorithm using commercial music software. This demo also aided in designing the emotive knobs. We will write DSP code to implement the final project. To generate the pulse, we used a synthesizer called Vital. Below is a screenshot of the settings. On the left side, we implement a few of the emotive knobs, to test if they work.


**Figure 8: Digital Pulse Generator Software**

Next, we created a feedback loop to implement part of the resonator. The changing delay time affects the pitch of the synthesizer. Below is a screenshot of the delay we used. There are many knobs, but we set most of them to a state where they don't do anything. The only knob we care about for this is the delay time (which is set to 7ms in the screenshot).


**Figure 9: Simulated Delay Line**

Lastly, we filtered the delayed signal to create the resonances, which finishes the resonator design. This filtered signal is sent back to the delay loop which repeats until the signal is too quiet to hear.


**Figure 10: Delay Line Feedback Filter + Frequency Spectrum**

We have uploaded some sound samples to Google Drive to demonstrate the intended output of our project ([Link]).

# 3. Ethics and Safety

**Section Overview**

In this section, we will be assessing the ethical and safety issues relevant to our project. We will be considering both issues arising during the development stages and those which could arise from the accidental or intentional misuse of our device.

## 3.1 Ethical Concerns

Ethical Concern #1: Loud noises/ringing pitches can cause hearing damage (IEEE Sec 7.8, Rule 1).

<u>Resolution Steps</u>

In order to address this concern, we will always start with the volume down and will slowly raise the volume for testing. The final product should be free of unintended loud noises and ringing pitches and will therefore not have this concern.

Ethical Concern #2: Producing a sound that already exists could lead to copyright infringement (IEEE Sec 7.8, Rule 5).

Resolution Steps

In order to address this concern, we will instruct users that prior to publishing their sound, they must first submit their music to the United States Copyright Office. This serves as a formal acknowledgment of their work and provides legal protection against potential infringement claims.

## 3.2 Safety Concerns

Potential Safety Hazard #1: Electrical hazards including risk of shock or fire.

Note

We are working with low voltages, thus the chance of shock or fire is low. Although this is the case, we will still make sure that the device is always powered off during modifications to prevent any potential injuries. The final product will be housed in a metal casing and since the case will be connected to ground, there will be no risk of electrical shock.

Potential Safety Hazard #2: Risk of burning ourselves with a soldering iron.

Note

To minimize the risk of burns while using a soldering iron, we need to stay focused and avoid distractions to maintain a steady hand. Using proper equipment, such as a stand to safely rest the soldering iron, is essential. If necessary, we can also protect ourselves by wearing heat-resistant gloves or using tools like pliers to hold components, keeping our hands away from the hot tip. It's also key that we remain aware of where the soldering iron tip is at all times, especially when moving it around the work area. And lastly, we should make sure to always turn off the soldering iron when taking breaks or after completing our task to prevent accidents. By following these precautions, we can work safely and reduce the risk of burns.

Potential Safety Hazard #3: Risk of inhaling soldering fumes.

<u>Note</u>

We will ensure proper ventilation by turning on the fan while soldering to reduce the risk of inhaling fumes.

Potential Safety Hazard #4: Metal cases can have sharp edges.

<u>Note</u>

If any sharp edges are found, we will sand them down to eliminate the risk of injury and ensure the case is safe to handle

# 4. References


[1] “IEEE Code of Ethics.” *IEEE - Advancing Technology for Humanity*, June 2020, www.ieee.org/about/corporate/governance/p7-8.html.

[2] “Professional Monitor Headphones: ATH-M50X.” *Audio-Technica*, www.audio-technica.com/en-us/ath-m50x.

[3] Dengate, Paul. “When Milliseconds Matter: The Reality behind Latency in Networked AV.” *HARMAN Professional Solutions Insights*, 23 Sept. 2022, pro.harman.com/insights/harman-pro/when-milliseconds-matter-the-reality-behind-latency-in-networked-av/#:~:text=Most%20experts%20agree%20that%20visual,and%20the%20user’s%20visual%20acuity.

[4] Purves, Dale. “The Audible Spectrum.” *Neuroscience. 2nd Edition.*, U.S. National Library of Medicine, 1 Jan. 1970, www.ncbi.nlm.nih.gov/books/NBK10924/#:~:text=Humans%20can%20detect%20sounds%20in,to%2015%E2%80%9317%20kHz.