Guitar Synthesizer and Harmonizer

ECE 445 Design Document

Madhav Khirwar, Danielle Lange,
and Ishan Jain
TA: Feiyu Zhang
Team 28
10/28/2021

Table of Contents

1 Introduction

1.1 Problem and Solution Overview

A lot of guitarists these days, especially in the metal/indie scene, like to double/harmonize their guitar parts with synthesizer parts overlayed on top [1]. This is usually done note-for-note, and usually wastes a lot of effort as the guitarist has to re-record the synth parts over the guitar tracks. Even if the synth is not played but merely programmed in through MIDI, there is a lot of wasted effort that is directly proportional to how complex/fast the guitar parts are. This product is meant to act as a device (guitar pedal) that adds musical context/layering on the notes played on a guitar. The guitarist will benefit from this product in that he/she can create more specialized sounds by blending together different notes and/or chords. This would make it easier for the musician to achieve the specific sound/note that they desire.

Our product is marketable because the cost will save a lot of time for the artist when they are tracking guitar parts and doubling them with synths, as well as cut down on the amount of stage gear they need when performing live. Our approach is to use a DSP-capable microprocessor (such as the MK20DX256VLH7 on the Teensy 3.2) to implement a novel kind of guitar pedal – one that will add electronically synthesized harmonies to an analog guitar note played by the guitarist. This will be in the form factor of a normal pedal, and the knobs will give the guitarist the option to select what kinds of harmonies they want to overlay (Major 5th, Major triad, 7th, etc).

1.2 Visual Aid

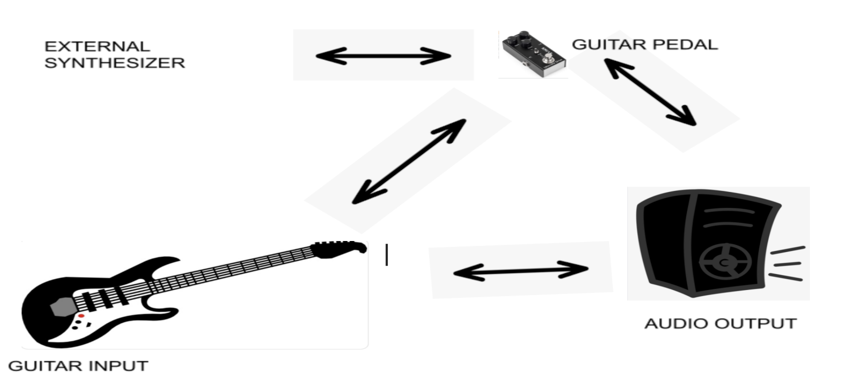

Figure 1: Depiction of how the signal chain would be used by the guitarist.

We envision the chassis being of dimensions equal to or less than (20 cm, 15cm, 8cm), and having 2 female ¼ inch connectors, one MIDI out, and one 9V DC power input. We want the dimensions to be such that it fits on a standard guitar pedal board and is compatible with the rest of the guitarist's signal chain. In terms of controls, we want the guitarist to be able to specify the harmony they would like to generate as well as the type of waveform (from square, sawtooth, or sinusoidal). The electric guitar's output will be connected to the input female ¼ inch connector on the pedalboard. The guitarist can use the harmony selector knob on the pedal to select what harmonies to generate as MIDI notes, and the waveform selector knob to select what waveform to generate. If the guitarist wants to use their own external synth with a more elaborate/customized waveform generator or signal chain, they can route the MIDI notes generated to the external synth. The output of the pedal would then be connected to either an amplifier or the rest of the guitarist's pedal chain. The guitar signal itself will bypass the entire circuit and be mixed and balanced with the pedal output. This is because we don't want there to be any tone-shaping of the guitar signal itself, we only need the guitar signal to calculate the harmony notes for the synth module.

1.3 High-Level Requirements

- Can resolve a frequency in the range 50-1500 Hz to the closest note in 12-tone equal temperament tuning system with A at 440 Hz. Error tolerance of +/- 5%.
- ADC/DAC should be at a sample rate of at least 44.1 kHz and at least 16-bit resolution [7]. This is because CD quality audio has 16-bit resolution and is sampled at 44.1 kHz.
- Makes overall music production and performance significantly more convenient for the guitarist.
- Total end-to-end latency should be under 10 milliseconds. For guitar, this is when guitarists can 'feel' the performance getting sluggish even though listeners cannot tell the difference [3].

2 Design

2.1 Block Diagram

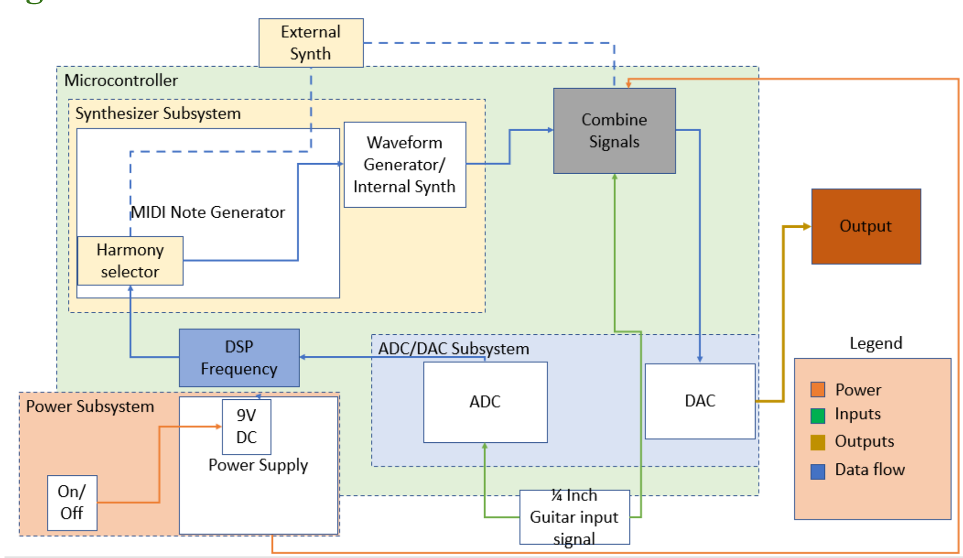

Figure 2: A block diagram for each of the subsystems, with arrows to depict the sequence in which each subsystem would process the guitar signal.

2.2 Physical Design

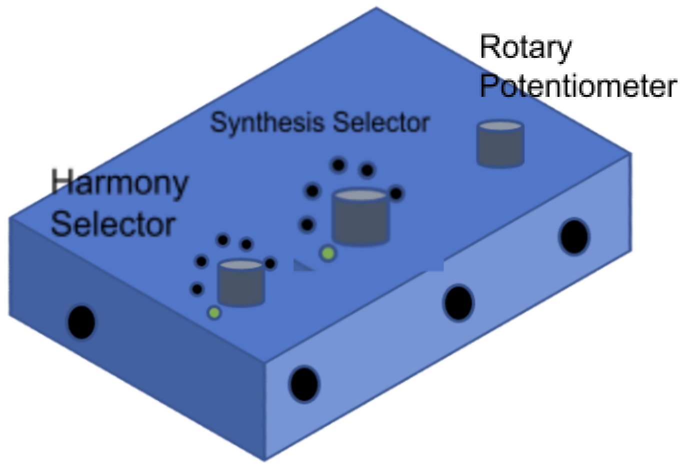

Figure 3: A depiction of what the final product would look like on the outside.

From left to right on the pedal diagram above, there is the power input (9V DC), the ¼ inch guitar input, a ¼ inch audio output, and a MIDI output for an external synthesizer. On the top, there are two knobs from left to right that respectively allow the guitarist to use a single hand to select the harmony and the waveform they wish to generate, and a push button that bypasses the signal if the guitarist wishes to play without the synthesizer.

2.3 Block Descriptions

2.3.1 Input/Output Subsystem

This subsystem is meant to ensure that the input guitar signal and output signal are consistent with other guitar effects pedals on the market, as well as ensure that throughout the signal chain from ADC to DSP processor to DAC, signal levels are adjusted according to the requirements of each sub-part.

Input Buffer

The input signal to the ADC needs to be attenuated by a factor of 3, since this stage is powered directly by the 9V power supply, it has roughly 9V peak-to-peak headroom. To match the headroom of the Teensy line input which is 3V peak-to-peak, we will use a divider circuit using two resistors (labelled R3 and R4 in the diagram). The circuit diagram is as follows:

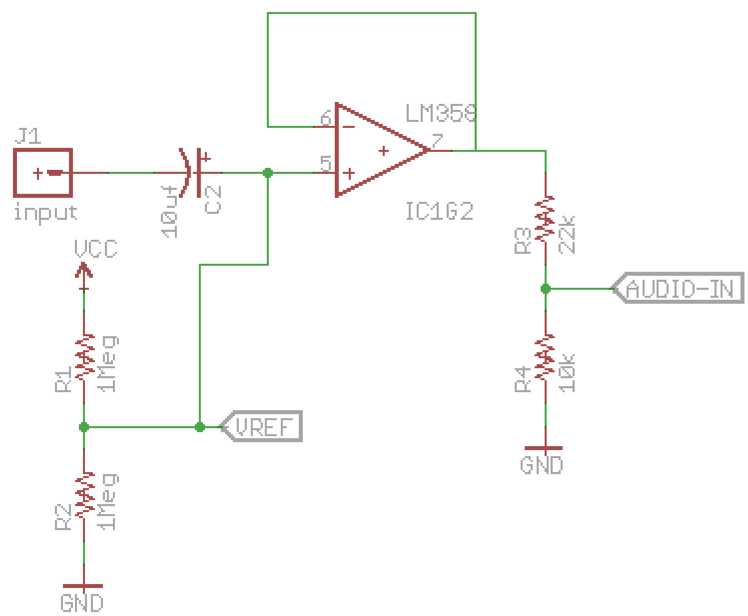

Figure 4: Input Buffer circuit, as described in a tutorial to make a reverb pedal with Teensy [10]

Requirements	Verification
• This subsystem should be able to divide the peak-to-peak voltage of an input signal by approximately 3. It should be able to do so for signals between 2V and 15V peak-to-peak voltage. • The input impedance is greater than 100kOhms to prevent signal attenuation	• This can be verified by attaching an oscilloscope to the input signal, calculating the peak-to-peak voltage electronically, doing the same for the signal after the input buffer and checking that the ratio is within ±5% of 3 to ensure that the buffered signal matches the headroom of the Teensy line input. We chose a ±5% deviance from this as the Teensy's max headroom is 3.3V and accounting for the pick attack and articulation of the

	played note, we want ensure that the signal doesn't occupy more than 3.15V to prevent signal clipping and distortion. ● The input impedance can be measured using a multimeter or oscilloscope, by using the formula $Z_{in} = V_{in}/I_{in}$

Analog to Digital Converter

This subsystem manages converting the signal from analog to digital which is critical to being able to harmonize the signals. Both ADC and DAC will be implemented on the Teensy Audio Shield because the DAC onboard the MK20DX256VLH7 on the Teensy board is only capable of a bit depth of 12 instead of 16.

Requirements	Verification
● Input is equal to or less than $1V_{rms}$. This is because 1Vrms is a little more than the highest guitar output voltages[11]. ● The digital signal is communicated to the microcontroller. ● Should be able to convert in the frequent range of 50-1500Hz with maximum ±5% signal distortion. This is because even with ±5% signal distortion, the fundamental frequency will be close enough to a 'correct' note that it can be resolved to it.	● Use the serial monitor and rms voltage function in the Teensy Audio Library to calculate the rms voltage of the input. It should be less than 1V, since 1V is only rarely reached even with high-gain guitar pickups and strong pickstrokes. ● The digitized signal should be received by the Teensy Audio board and this can be verified by running the serial monitor and ensuring there is input from the ADC. ● An oscilloscope can be used to measure the signal frequency after the input buffer, and the internal note2freq serial monitor can be used to measure the signal frequency from the ADC. They should be within ±5% of each other.

Digital to Analog Converter

This subsystem manages converting the signal from digital to analog after the addition of the harmony notes. Since we are aiming for the highest harmony note to be an octave (double the frequency of the input), the constraints and requirements of the DAC are higher. Luckily, the Teensy Audio Shield can manage this easily with a 16 bit output.

Requirements	Verification
• Output is equal to or less than $1V_{rms}$. This is because $1V_{rms}$ is a little more than the highest guitar output voltages. • Should be able to convert in the frequent range of 50-3000Hz with maximum ±1% signal distortion. This is a tighter boundary, as these are the frequencies that the user will actually hear and they need to be a lot more accurate than the ones required for note computation.	• Use the serial monitor and rms voltage function in the Teensy Audio Library to calculate the rms voltage of the output. It should be well under 1V, so that it is not very difficult to balance with the original guitar signal. • An oscilloscope can be used to measure the signal frequency after the output buffer, and the internal note2freq serial monitor can be used to measure the signal frequency from before D-A conversion. They should be within ±1% of each other.

Output Buffer/Mixer

The output signal from the DAC needs to be amplified by a factor of 3, since the standard peak-to-peak headroom for guitar pedals is 9V. This stage also needs to mix the output of the synthesized notes with the raw output of the guitar signal, so that neither of the two overpowers the other to the listener. This will be achieved by a gain circuit consisting of two resistors labelled R7 and R8.

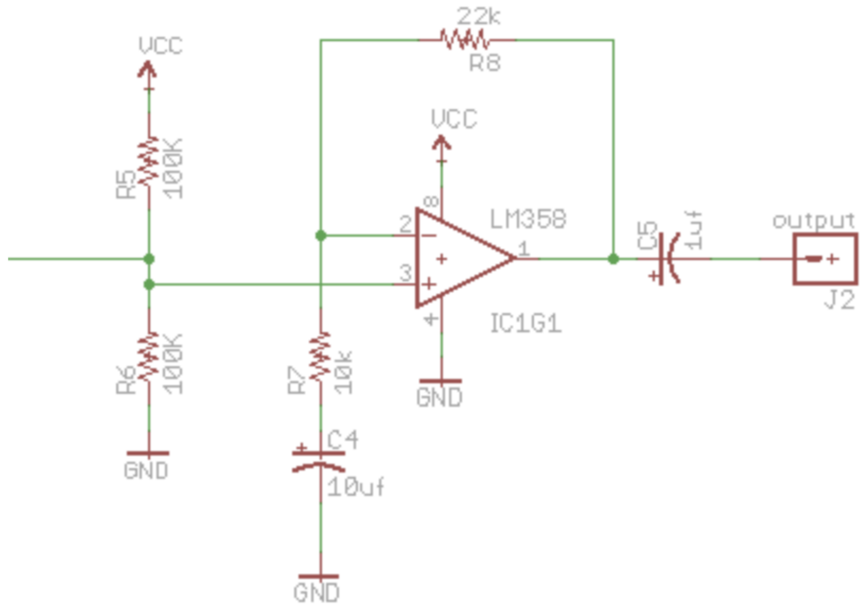

Figure 5: Output Buffer circuit, as described in a tutorial to make a reverb pedal with Teensy [10]

Requirements	**Verification**
• This subsystem should be able to multiply the peak-to-peak voltage headroom of an input signal by approximately 3. It should be able to do so for signals between 0.5V and 5V peak-to-peak voltage. • The output impedance is greater than 100kOhms to prevent signal attenuation • The output from the DAC and the raw guitar signal should be balanced in rms Voltage.	• This can be verified by attaching an oscilloscope to the output signal from the DAC, calculating the peak-to-peak voltage electronically, doing the same for the signal after the output buffer and checking that the ratio is within ±5% of 3. • The output impedance can be measured using a multimeter or oscilloscope, by using the formula $Z_{out} = V_{out}/I_{out}$ • This can be verified by attaching an oscilloscope to the output signal from the DAC, calculating the rms voltage electronically, doing the same for the signal from the guitar and checking that the ratio is within ±5% of 1.

2.3.2 DSP Frequency Analyzer Subsystem

This subsystem is meant to recognize the note that the guitarist has played after Analog to Digital Conversion. This will be done by mapping the frequency of the played sound to the closest note in the 12-tone equal temperament, A at 440 Hz system. The following are the frequencies of the 12 notes on the guitar in the lowest frequencies[11]:

Note	Frequency in Hz
E	82.407
F	87.307
F# / Gb	92.499
G	97.999
G# / Ab	103.826
A	110
A# / Bb	116.541
B	123.471
C	130.813
C# / Db	138.591
D	146.832
D# / Eb	155.563

Table 1: The frequency of the lowest 12 notes on the guitar

The algorithm used to calculate the played note will check which of these notes divides the input frequency with the smallest remainder (to calculate octaves), and will assign that note according to the table. This will be implemented on the main Teensy board, and the Audio library provides functionality to recognize the note frequency.

Requirements	Verification
• Should be able to resolve a frequency to the nearest 'correct' note within 5 milliseconds. • Tie break to the higher note.	• This can be verified by attaching an oscilloscope to the input signal from the guitar as well as the output from the frequency analyser, and from the plot of the waveforms calculating the time difference between the starts of the two signals. • This can be ensured by using the waveform generator to input a frequency exactly halfway between two notes, and ensuring that the output note is the higher one.

2.3.2 Synthesizer Subsystem

This subsystem purpose is to select the notes and the type of waveform that will be combined with the original signal and output at the end. Since we want the guitarist to have the option to use an external synthesizer to play their own custom waveforms, we will have a MIDI output that sends note information via a MIDI output.

Note Generator:

The Midi Node Generator will receive the note from the tuner and choose what MIDI notes to harmonize with the note input from the tuner. This will only happen each time the note played by the guitarist changes, to avoid a situation where tremolo picking or sustained notes causes a 'stuttering' effect as the same harmony is continuously generated. This subsystem will also generate a MIDI output that can be sent directly to an external synth.

Requirements	Verification
• Latency should be under 5 milliseconds. • Should be able to calculate at least the following harmonies: unison, fifth, major third, minor third, major 7th, minor 7th. Advanced functionality may include picking a key and generating the diatonic chords.	• This can be verified by attaching an oscilloscope to the input signal from the guitar as well as the output from the frequency analyser, and from the plot of the waveforms calculating the time difference between the starts of the two signals. • The outputted notes can be manually checked against chord charts to ensure

• Should allow the guitarist to quickly select the desired harmony setting	that the correct harmonies are being generated. • This can be done by checking if a guitarist can use only one hand to change the settings, and if it can be done within ~2 seconds.

MIDI Output:

The external synthesizer is optional based on the user. If the user wants the effects of our internal synthesizer this subsystem will not exist.

Requirements	Verification
• The MIDI notes should be able to drive any MIDI-controlled device and should externally appear as a standard MIDI controller.	• To ensure that the output is MIDI standard compatible, one can connect a laptop and use multiple digital instruments in a DAW, and compare the generated notes with the expected notes using a software keyboard.

Waveform Generator/Internal Synthesizer:

The purpose of the internal synthesizer is to decide what timbre the notes from the MIDI Note Generator will have. It serves to display the waveform and has a selector deciding whether it will change the waveform of the note to be a square wave, sinusoidal.

The chip used for this synthesizer is the MK20DX256VLH7 which will be prototyped on the Teensy 3.2 board. This chip is chosen because the Teensy Audio Library has extensive support for Digital Signal Processing Applications in real time.

Requirements	Verification
• The user should be able to select between square, sinusoidal, and sawtooth waveforms. Advanced functionality would include mixtures of these waveforms. • The waveforms should be generated from the calculated notes, and	• The waveform shape can be verified on an oscilloscope. • The note values can be verified by reading the serial monitor output of the synthesizer subsystem on the Teensy microcontroller itself.

shouldn't include the 'root' note of the chord	

2.3.3 Power Subsystem

This subsystem's purpose is to supply power to all the different parts of the system, as well as to ensure that voltage conversions take place depending on the voltage requirements of different chips. Since the power supply needs to be compatible with the rest of the guitarist's pedalboard, it will take a 9V DC input. However, the Teensy microcontroller operates on a maximum of 6V, so we will use a LM7805 Voltage Regulator IC to convert from 9V to 5V [5]. Since the Teensy operates on about 60mA of current, and the voltage drop is 4V, this subsystem will dissipate about 0.25 Watts as heat - which should not make it appreciably hotter than room temperature and should still be quite safe to the circuit. To further ensure thermal safety, we will attach this chip to a heat sink that will dissipate the power across the chassis of the entire pedal.

The Teensy itself has on-board voltage regulators that will step 5V down to 3.3V for DSP operations.

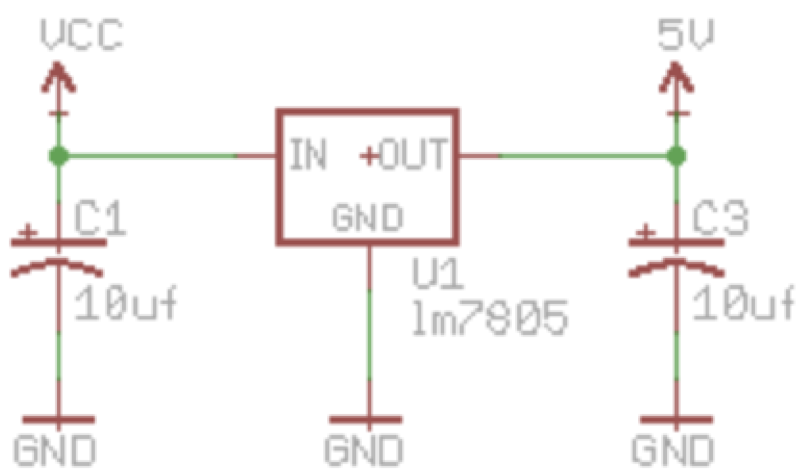

Figure 6: Voltage regulator circuit, as described in a tutorial to make a pedal with Teensy [10]

Requirements	Verification
• Since the guitar pedal is ideally meant to be as easy to plug in and use for the player as possible, we want it to be compatible with the power supply of the rest of the pedal chain. Guitarists often have a single 9V DC adapter and simply attach multiple 5.5mm * 2.1mm connectors to them for each pedal in a 'daisy chain' [4]. Be powered by an external 9V DC power supply rated under 500mA with a 5.5*2.1mm barrel connector with negative polarity in the center [7]. • The voltage conversion from 9V to 5V should be accurate within ±5%. • The temperature of the LM7805 should not exceed 40 degrees C, to ensure that none of the other parts are negatively affected and that the pedal itself doesn't become too hot.	• We will verify compatibility with standard guitar pedal power supplies by simply adding our guitar pedal to an existing pedal chain and ensuring that it is running with full functionality. • We can ensure that the voltage conversion is as desired by using a multimeter to read the voltage from the DC power supply as well as the voltage from the LM7805 chip. • We can use an electronic thermometer to ensure that the temperature of the LM7805 doesn't exceed 40 degrees C.

2.4 Tolerance Analysis

An important aspect of our project is considering the combining of the original guitar signal with the harmonized signal. The reasons for concern are that the signals must be balanced in amplitudes and the harmonized signal must not contain much noise. At the point of combining the audio signals our output is finally created but making sure the audio quality is optimal is an important part of our product from the perspective of a user.

To combine the signals we would use an op-amp summing amplifier built on the PCB.

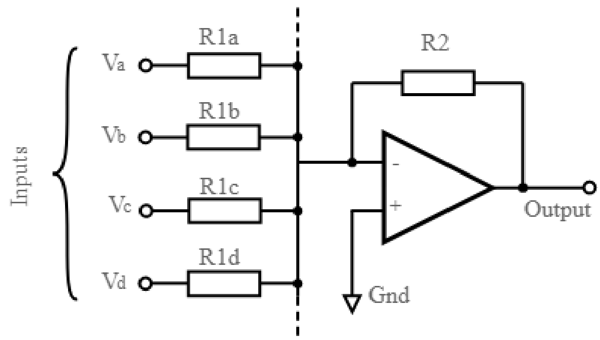

Figure 7: Summing op-amp [9]

The values of the resistor values will give different values of gain in the output. In our project Va is the output voltage of the original raw guitar signal and Vb is the voltage from the harmonized/synthesized signal. Ideally we want the Vout (gain) to be equal to the input from the guitar, as well as have a balance of both the guitar signal as well as the synthesized output. The formulas for a summing amplifier are as follows [9]:

The output Voltage value that Va creates is calculated by:

$$\text{Vout}=-\ Va(\frac{R2}{Ra})$$

The output Voltage value that Vb creates is calculated by:

$$\text{Vout}=-\ Vb(\frac{R2}{Rb})$$

In order to have a balanced final signal, we will have to experimentally determine the values of R1, R2, and Rb because even if the rms voltages of the synth output and the guitar signal are equal, the waveform shapes are different and accounting for the variance in voltage from the guitar will have to be done experimentally. Specifically, the output voltage of the guitar signal will vary according to how hard the note is picked/how long the note is sustained, whereas the output of the synth will be a continuous waveform that will have the same amplitude throughout.

For some of the duration of the note, the guitar signal will be slightly louder while for the rest, the synth signal will be slightly louder. This in and of itself is not a bug, since in the context of music production, there are notes that are emphasized at the beginning by the pluck of a guitar string and after a while the synthesizer sound overpowers the guitar sound. Finding a resistor combination that balances both signals in a desirable way will need trial-and-error, since the waveforms of a single note played on a synth vs an electric guitar are different, as evidenced by the signal decay of a single picked guitar note compared to the constant sine wave output from a synthesizer:

Figure 8: Guitar signal decay from a single pick stroke, as the player plays an open low E string [12]

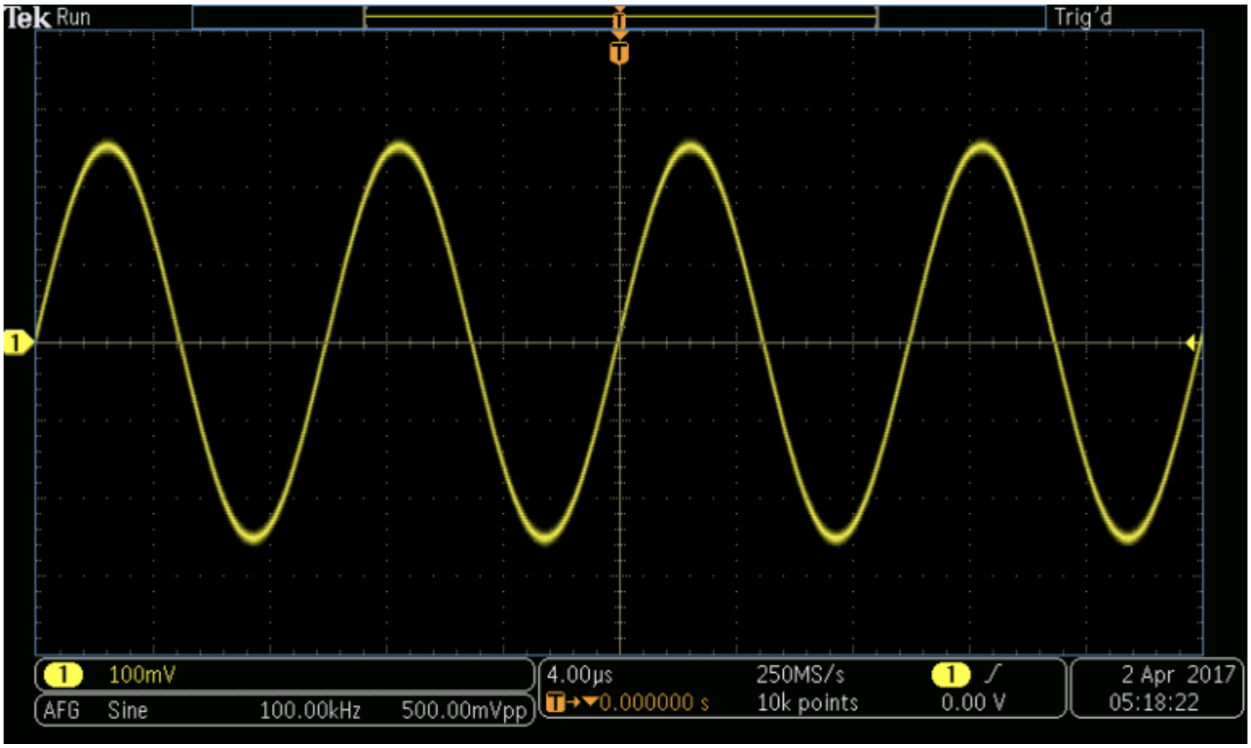

Figure 9: Sine wave signal as produced by a synthesizer [13]

3 Cost and Schedule

3.1 Cost Analysis

The expected cost per person assumes an average of the reported Computer and Electrical Engineering graduating salaries from 2019-2020. They are sequentially 110,978 and 76,129 making the average 87,637 a year [0].

$$\frac{87{,}637\$}{year} \times \frac{year}{261\ work\ days} \times \frac{work\ day}{8\ hours} = 41.97\$/hour$$

Estimates for total hours assume the time it will take to build/debug once the planning and designing is complete.

$$\frac{41.97\$}{hour} \times \frac{20\ hours}{person} = \frac{839.8\ \$}{person}$$

$$Total\ labor\ cost = 839.8\$ \times 3people = 2518.2\$$$

Part	Part Number From Manufacturer	Manufacturer	Quantity	Cost/ unit	Description [7]

Teensy 3.2	DEV-13736	PJRC	1	19.80	MCU 32-Bit embedded evaluation board. Core processor of ARM@ Cortex-M4.
Teensy Audio Board	SGTL5000	PJRC	1	17.95	Tool to add 16 bit and with a 44.1kHz sampling rate. Attaches to the Teensy 3.2. Include incorporated synth waveforms, effects. Supports stereo line-level output [1].
Black Knob	COM-09998	SparkFun Electronics	2	4.75	Knob connects to a 0.25" shaft with a set screw.
Rotary Potentiometer	COM-09939	SparkFun Electronics	1	4.75	Adjustable potentiometer which turns will change resistance. Varies voltage with input. ¼" mounting parameter with a 10k linear taper.
Hook-Up Wire- Stranded, 22 AWG	PRT-11375	SparkFun Electronics	1	20.95	Assortments of colored wires.
Hook-Up Wire- Solid Core ,22 AWG	PRT-11367	SparkFun Electronics	1	19.95	Assortments of colored wires.
	COM-00107	SparkFun Electronics	1	0.95	L7805 voltage regulator that has a 5V output. Maximum current of 1.5A. Used for voltage regulating.
DIP Sockets Solder Tail	PRT-07937	SparkFun Electronics	1	0.50	300mil wide 8-pin socket.
Electrolytic Decoupling	COM-00523	SparkFun Electronics	5	0.45	Radial electrolytic capacitors. Maximum

Capacitor					voltage of 25V, the capacitor can hold a charge of 25V×10μF
Resistor Kit	COM-10969	SparkFun Electronics	1	7.95	Comes with 20 different resistor values and 25 of each kind.
Break Away Headers	PRT-00116	SparkFun Electronics	1	1.50	40 pins that are breakable for use with the PCB

Assuming all products are used(all wires, all capacitors) the total cost of parts is 101.30$. The total cost of everything including labor time fo all three group members is:

$101.30\$ + 629.55\$ = 730.85\$$

3.2 Schedule

	Tasks		
week	Danielle	Madhav	Ishan
Sept 27th-Oct 3rd	-Research the PCB layout -Practice speaking about the tolerance section to end of design document -order Teensy Audio Board	-Research the PCB layout -Practice speaking about the block descriptions	-Research the PCB layout -Practice speaking about the intro to the physical design section of the design document
Oct 4rd-10th	-Research the PCB layout about teensy microcontroller/audio board -Kicad the PCB layout	-Research the PCB layout about midi note generator -Kicad the PCB layout	-Research the PCB layout about combining signals
Oct 11th-17th	-Get guitar signal through the ADC	-Get teensy to recognize closest note(tuner)	-Antialiasing on the guitar signal
Oct 18th-24th	-Get different harmony options on midi	-Get note to be recognized on midi get check its validity	-Get knob to select different harmonies on midi
Oct 25th-31st	-Get signal recognized and check its validity	-Set up the different waveform options	-Make knob select different waveforms of the singal
Nov 1st-7th	-Perform antialiasing of signals	-Get guitar signal through the DAC	-Combine signal from synth with original signal

	-Start plan and research for debugging issues	-Connect output to speaker and test results	
Nov 8th-14th	-Prepare for demo of tuner -Work on Mock Presentation on tuner -Debug issues of project	-Prepare for demo of midi note generator -Work on Mock Presentation on midi -Debug issues of project	-Prepare for demo of synthesizer -Work on Mock Presentation of synthesizer -Debug issues of project
Nov 15th-21th	-Work on Final paper beginning and parts personally worked on -Debug issues of project	-Work on Final paper middle part and parts personally worked on -Debug issues of project	-Work on Final paper and parts personally worked on -Debug issues of project
Nov 22nd-28th	-Break	-Break	-Break
Nov 29th-Dec 5th	-Work on Final paper beginning and parts personally worked on	-Work on Final paper middle section and parts personally worked on	-Work on Final paper beginning section and parts personally worked on
Dec 6th- 13th	-Revise Madhavs final paper parts	-Revise Ishans final paper parts	-Revise Danielles final paper parts

4 Discussion of Ethics and Safety

As the IEEE code of ethics says we must “hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development”[5]. In order to do this we must consider the fact that someone can be unaware of the risks of the system and make the audio output louder than safe for people's ears. Anything 80 db and above can cause hearing damage. To comply with an ethical standard our product will include a warning with the range of dB that can cause hearing damage.

It is also a concern that because we are using circuitry that our design is rain proof. Water can cause short circuiting which is a fire safety hazard. To do this we must make sure that our circuit design is sealed enough to not let any water in. We would also need to ensure that all voltages and currents are appropriately grounded so as to not make the strings of guitar live wires. This can be extremely dangerous for the player and adhering to design standards will help prevent this.

The risk of electrical shock if mishandled also would be a safety hazard to children. From the AMC ethic code 1.2 to avoid harm our team wants to avoid any possible risk to anyone's safety [2]. Our team plans to put a safety warning on the product to keep away from small children to avoid a hazard like this from occurring.

5 Citations

[0]"Annual Reports | Illini Success - Illinois." *University of Illinois at Urbana Champaign.* 2021, https://illinisuccess.illinois.edu/annual-reports/
[1]"Audio Adaptor Board for Teensy 3.0 – 3.6." 2021, www.amazon.com/Audio-Adaptor-Board-Teensy-3-0/dp/B013V0ZK1E
[2] "ACM Code of Ethics and Professional Conduct." *The Association for Computing Machinery*, 2021, www.acm.org/code-of-ethics
[3] A. Swanson, "Latency and Its Effect on Performers." Church Production Magazine, 19 June 2017, https://www.churchproduction.com/education/latency-and-its-affect-on-performers/
[4] B. Neunber, "Pedal Power Basics." *Neunaber Audio*, 2021, https://neunaber.net/blogs/neunaber-audio-blog/13849473-pedal-power-basics
[5] "IEEE Code of Ethics." 2021, https://www.ieee.org/about/corporate/governance/p7-8.html
[6] Nicholas, "Doubling Guitar with Synthesizers - zZounds Music Blog." *zZounds Music Blog*, 2018, https://blog.zzounds.com/2017/12/20/doubling-guitar-tracks-with-synthesizers/
[7]"SparkFun Electronics." 30 Sept. 2021, www.sparkfun.com/products
[8] Tom, "Electric Guitar Output Voltage Levels." 2021, http://tomsguitarprojects.blogspot.com/2014/12/electric-guitar-output-voltage-levels.html
[9] "Op Amp Summing Amplifier: Virtual Earth Mixer » Electronics Notes." 2021, www.electronics-notes.com/articles/analogue_circuits/operational-amplifier-op-amp/virtual-earth-mixer-summing-amplifier.php
[10]Byron, "Proto Pedal Example: Programmable Digital Pedal" 2021, https://learn.sparkfun.com/tutorials/proto-pedal-example-programmable-digital-pedal/all#assembly-part-2-teensy-and-controls
[11]Mottola, "Table of Musical Notes and Their Frequencies and Wavelengths" 2020, https://www.liutaiomottola.com/formulae/freqtab.htm
[12]"Tom's Guitar Projects", "Electric Guitar Output Voltage Levels" 2014, https://tomsguitarprojects.blogspot.com/2014/12/electric-guitar-output-voltage-levels.html
[13]David Herres, "Waveforms in oscilloscopes and elsewhere" 2017, https://www.testandmeasurementtips.com/waveforms-oscilloscopes-elsewhere/