Guitar Pedal Synthesizer and Harmonizer

ECE 445 Final Report

Ishan Jain

TA: Feiyu Zhang

Team 28

12/08/2021

ABSTRACT

This report details the design and implementation of processes of a novel guitar pedal and synthesizer that dynamically recognizes notes played as a digital signal and overlays them with synth harmonies in order to conveniently create unique sounds within a guitar pedal chain. This report goes into specific details of each component of this device in a technical sense. It also outlines the high- and low-level requirements of each module as well as how these requirements were verified using lab tools and online research. Each process is carefully discussed, and an estimated cost and labor value is determined regarding development of this device. Many important figures, tables, and schematics are also included as they were essential to conceptually understanding the material. A succinct conclusion discussing results follows and two appendices detailing overall schematics and cost can be found at the end of the paper.

Table of Contents

INTRODUCTION

Many guitarists these days, especially in the metal/indie scene, like to double/harmonize their guitar parts with synthesizer parts overlayed on top. This is usually done note-for-note, and usually wastes a lot of effort as the guitarist must 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/layer 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 different notes and/or chords. This would make it easier for the musician to achieve the specific sound/note that they desire.

The main component of our device is seen as a DSP enabled microprocessor which is the MK20DX256VLH7 on the Teensy 3.2 microchip. This enabled us to develop a multifunctional guitar pedal that allows the guitarist to create a more unique sound by adding electronically synthesized harmonies to an analog guitar note. Various knobs on the pedal allow the guitarist to choose specific harmonies and waveforms to generate the desired sound from the speaker. This product is useful and 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 cutting down on the amount of stage gear they need when performing live.

The specific subsystems we implemented include the power subsystem, the input/output subsystem, the ADC/DAC subsystem, the DSP subsystem, and the Synthesizer subsystem. Beginning with the power subsystem, the main purpose is to supply power to all the distinct parts of the system, as well as to ensure that voltage conversions take place depending on the voltage requirements of different chips. This also includes a voltage regulator which ensures too much voltage will not damage the Teensy 3.2 since this microchip can handle a maximum of 6V. The ADC/DAC subsystem was implemented in software instead of on the audio shield as initially planned because this allows us to account for 16-bit depth rather than 12. This system is responsible for converting the analog guitar signal to a digital signal which will allow us to combine harmonies. The DAC then converts this combined signal back to analog output once the harmonies have been combined. The input-output subsystem is meant to ensure that the input guitar signal and output signal are consistent with other guitar effects pedals on the market. It will also make sure that throughout the signal chain from ADC to DSP processor to DAC, signal levels are adjusted according to the requirements of each sub-part. The DSP subsystem is meant to recognize the note that the guitarist has played after Analog to Digital Conversion. This is done using binary search on an array in a 440 Hz system. The synth subsystem had the main purpose of allowing the guitarist to choose specific harmonies or waveforms they would like to overlay. The later sections of this report will detail each of these functional components more thoroughly.

We were able to achieve the high-level requirements we had set and succeeded in making a functional device that allows the guitarist more conveniently output a customized sound as a component of the pedal chain. From a technical standpoint, we were able to complete our high-level requirements by ensuring there was less than 10 milliseconds of latency factor (measured 5). Also, we were able to complete our other requirements by obtaining 16-bit resolution due to how we implemented the DAC subsystem in the software portion instead of using the audio shield. This will be later discussed in the design details section of the report.

BLOCK DIAGRAM

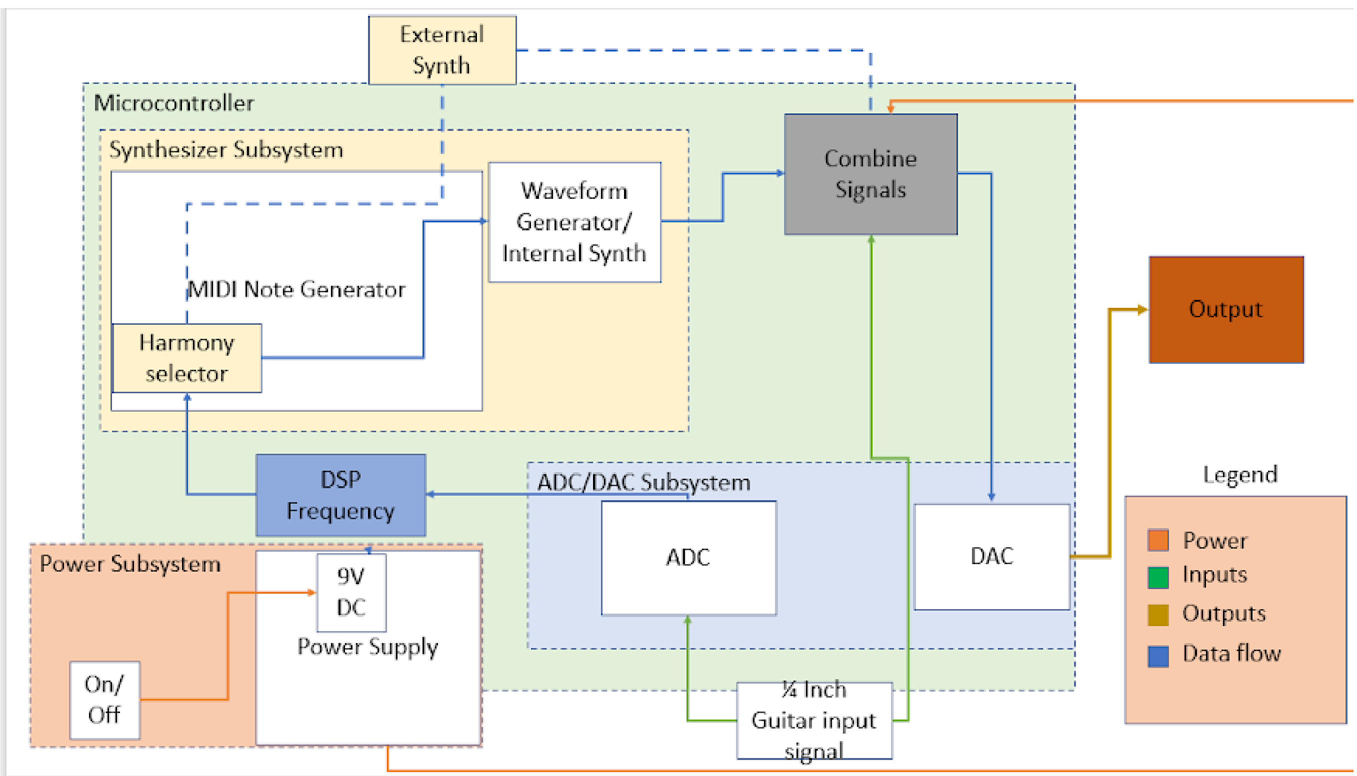


Figure 1: A block diagram for each of the subsystems

Figure 1 depicts each subsystem of the device as well as how they interconnect which shows the direction of signals regarding inputs, outputs, and flow of data. As seen in the bottom left of the diagram, the Power subsystem uses a 9V DC power supply and supplies power to all components of the device shown by orange connections. The. ¼ inch Guitar input signal represents the analog guitar input signal and first connects to ADC converter so that the signal may be recognized by DSP software algorithm. The signal then travels to the harmony selector and waveform generator. Once the harmony/waveform is known, combinational logic in the software combines this with the initial analog signal and outputs that to an external speaker.

DESIGN

Design Procedure:

Throughout the semester, there were many times where a change of direction was either needed for base functionality or preferable due to higher performance, efficiency, or general convenience from the perspective of the guitarist. One instance of a design change early in the process was deciding to go with a 9V power source rather than a battery or portable power source. We realized this was more compatible with most guitar chains and electric guitars often stay plugged in to avoid the risk of a battery losing charge during a performance. For this reason, we decided that a battery was not the right choice for our specific design. The Input Buffer was meant to attenuate the input signal by a specific factor of 3. This was the correct decision made because the ADC module on board the Teensy needed a peak-to-peak headroom of 9V corresponding to the initial power source. The main decision making that was needed regarding the DSP module was choosing the correct algorithm to efficiently detect and search a table for frequency of the signal. Our initial idea was to use a hash table and let the alphabetic notes and their fundamental frequencies be the respective keys and values. However, upon doing more research, the YIN algorithm was the best choice, and its capabilities of efficient binary search was a more promising alternative. The details of this algorithm will be explained in later sections. The decision making involved in the combinational logic was also altered through experimentation and research throughout the semester. This module was originally planned to be a hardware component in the form of an op-amp adder circuit. However, we saw that this had adverse effects on latency issues as the signals would often not arrive at the same time. For this reason, we decided that software implementation of this module would increase efficiency of signal processing and would therefore reduce latency as it did

Figure 2: A physical showing of the exterior of the device depicting harmony/waveform selectors

Another primary alteration we made in our design involved the DAC subsystem. This module was originally planned to be implemented on board the Teensy Audio Shield. However, after completing base functionality using this part, we decided to make the project more challenging by implementing the DAC ourselves instead of simply using the on-board converter on the audio shield. In hindsight, this may not have been as beneficial as we thought because we saw that the audio shield provided higher sound quality. Yet, our reasoning still held because we were able to process in 16 bits with software rather than 12 which allowed for higher and more compatible resolution. We also made changes to the physical design, depicted in Figure 2, by means of removing the rotary potentiometer with a button to bypass the signal and use the signal without the synth module. We made this decision to improve convenience for the guitarist.

Design Details:

Power Module:

This module (schematic depicted in Figure 4) was designed to properly disperse power throughout the circuit and each of its components in a way that only the specified voltage is received by certain subsystems. For this to be executed, the components we included in our implementation of this module included the 9V DC power source, LM7805 Voltage Regulator IC, and the secondary regulator on board the Teensy microchip. The LM7805 (depicted in Figure 3) had the primary purpose of regulating the input voltage of 9V and making it step down. to 5V instead. This was an essential component to this module because the Teensy 3.2 Microcontroller can only operate properly under a maximum voltage of 6V [4]. Another minor component of the power subsystem is the on-board voltage regulator which regulates the voltage from 5V to 3.3V to ensure not too much heat is dissipated. We were able to make sure there was not too much heat by doing a simple calculation:

Current of Teensy Operation (60 mA) * Voltage Drop Across Circuit (4V) = 0.25 W heat:

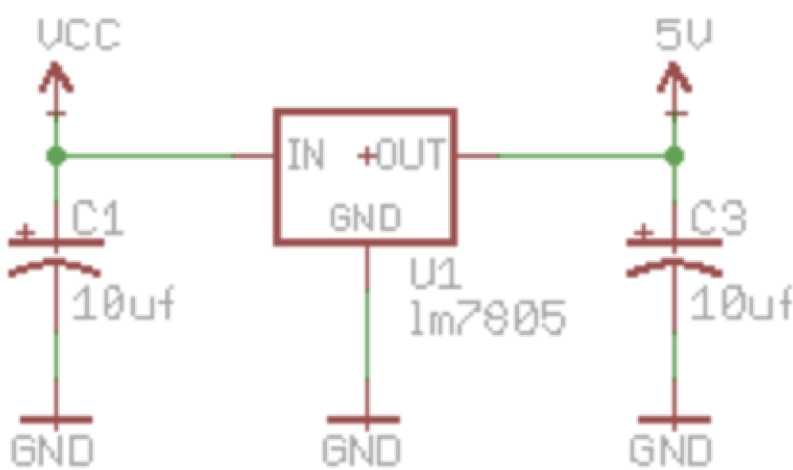


Figure 3: Circuit for Voltage Regulation

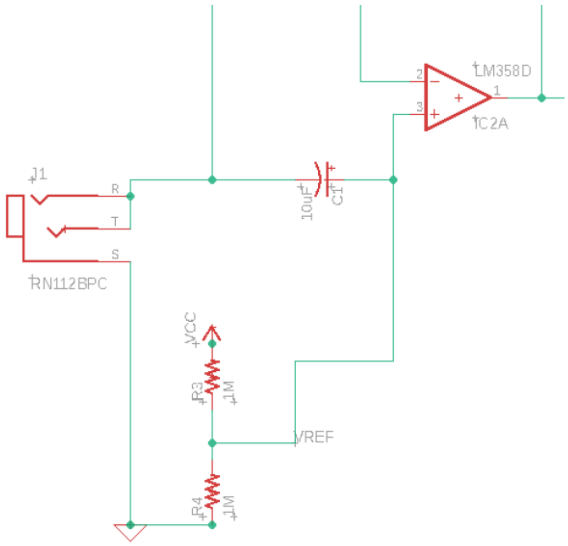
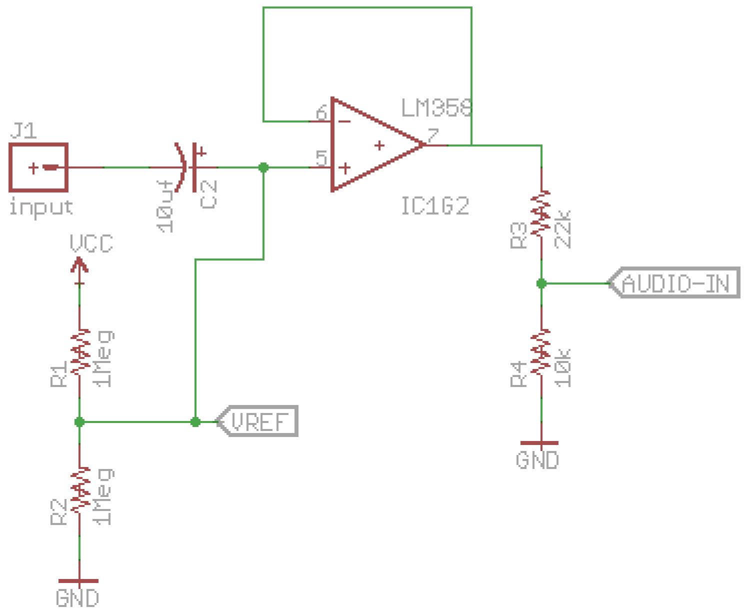


Figure 4: Circuit Schematic for Power Module

Input Module:

The input module is a subsystem that primarily deals with an input buffer which modifies and processes the analog input signal from the guitar before it is ready for ADC conversion on the Teensy microcontroller. The main purpose of this module is to ensure that input and output signals to the device are consistent with what is found on the guitar pedal market because it is essential that this product works on all electric guitars with similar input and output signals. This subsystem also ensures that throughout the signal chain, ADC to DSP to DAC, the signal is adjusted as it needs to be in order for there to be consistency in timing as well as peak voltage for each signal. The specific means by which the input buffer achieves this is by reducing the input signal voltage before the ADC by a factor of three. This is a crucial step in the signal chain because the input signal stage is powered directly by the 9V power source, and it has the same voltage regarding peak-to-peak headroom. Since the Teensy requires a headroom of 3V peak-to-peak, we needed to use a voltage divider circuit (schematic shown in figure 5). This circuit was successful in creating our input buffer to adjust the signal before ADC module was called upon.

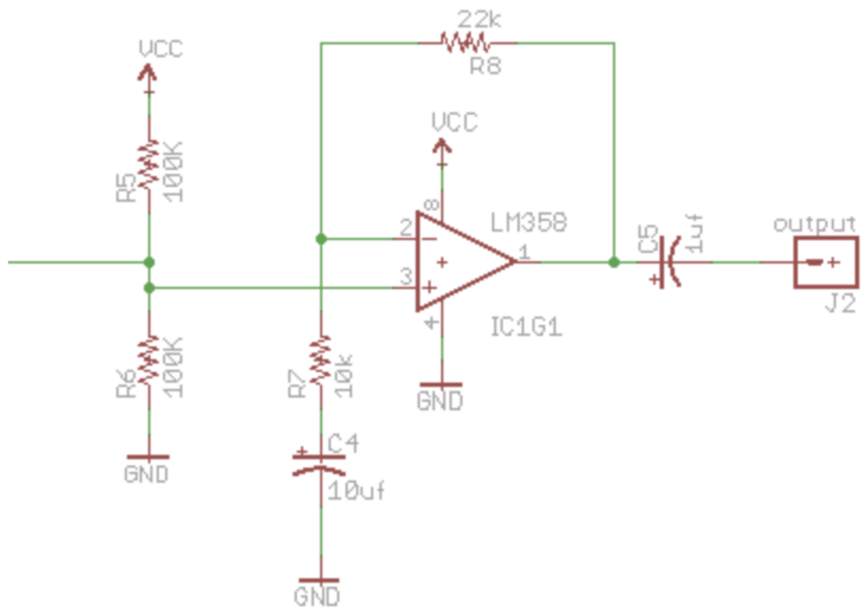
Figure 5: Input Buffer Circuit Schematic



Output/Combinational Module:

The output buffer handles the output signal from the DAC and does the reverse of the input buffer functionality. Since the voltage of the signal at this stage of the circuit is measured at 3V regarding peak-to-peak headroom, this value needs to be multiplied by a factor of three in order to amplify the signal so that it is more compatible with peak-to-peak headroom of the pedal chain which is 9V [3]. This buffer also has additional functionality it is responsible for in regard to mixing the output of the synth notes with the guitar signal analog output. We achieved this by using a simple gain circuit across two resistors (R7 and R8) depicted in the output buffer circuit below by Figure 6.

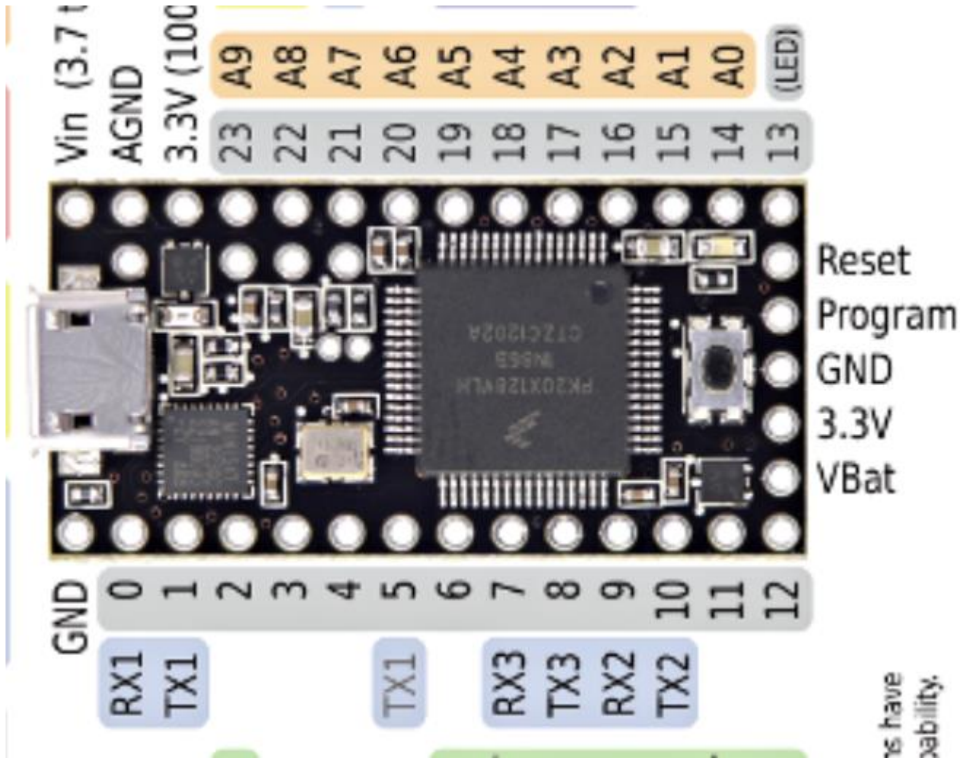
Figure 6: Circuit Schematic of the Output Buffer



ADC Module:

This module's primary purpose in this design is to intake the processed input signal and convert the analog signal into a digital signal which is essential to then be able to recognize the frequency and select a harmony to overlay. Initially, we had planned to implement both of these on the Teensy Audio Shield and have all ADC/DAC operations done on this shield itself. However, we decided that implementing these in a different manner would increase both the efficiency of the product as well as the complexity of the design. For ADC, we use the 16-bit CD quality analog input which is already implemented on board the Teensy 3.2 microcontroller. This ADC conversion along with the attenuation by factor three by the input buffer properly processes the raw input signal so it can be fed into the DSP analyzer. The input pin for 16bit ADC on the Teensy 3.2 is labeled by A2 depicted in Figure 7 below:

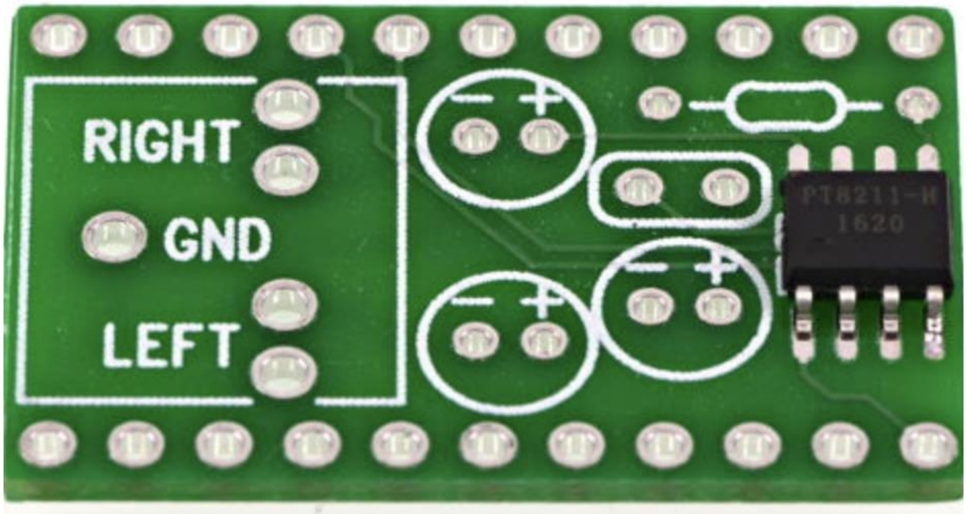
Figure 7: Teensy 3.2 where A2 carries out 16-bit ADC at 44.1 kHz [4]



DAC Module:

This module has the main purpose of converting the signal from digital back to analog once the addition of the corresponding harmony for that note has been combined with the raw guitar input. Although we had initially planned to use the Teensy Audio Shield for 16-bit output, we decided to increase the complexity of our design by implementing our own. In order to implement our own DAC, we built a circuit using a PT8211 DAC chip (pictured below in Figure 8) which also contains the 16-bit CD quality that we required to enhance the resolution of the output signal in the same way we did for ADC [7]. This process combined with the functionality of the output buffer allowed us to effectively convert the digital signal to analog with the proper amplified effect. This ensured that a 9V peak-to-peak headroom was achieved to ensure compatibility with all guitar pedal chains.

Figure 8: PT8211 DAC chip on board for Analog Output at 44.1 kHz



DSP Frequency Analysis Module:

This subsystem is meant to recognize the note that has been played once the signal has been converted from analog to digital form. The primary way this is done is by mapping the fundamental frequency of the note to a 12-tone temperament in a 440 Hz system. This entire module was done in software in the form of a search algorithm known as the YIN algorithm. The YIN algorithm specializes in real-time fundamental frequency estimation. This was the perfect algorithm for us to use because our note recognition algorithm had to be efficient and done in real time as well. This algorithm uses two main formulas (depicted in Figure 9) to calculate and estimate the fundamental frequency of the input signal. The algorithm will then output the frequency depicting the highest amplitude. To then recognize the actual note, we used a sorted array filled with forty-nine distinct values that correspond to the frets on a guitar. The algorithm then uses binary search on this array to locate the closest frequency value according to a twelve-temperament system (depicted in Table 1). Once the note has been recognized, the next step is to calculate the specific harmonies needed from the synth and this is done by using the interval multipliers and doing a calculation with the recognized note and multiplying them both. The final step is to generate the waveform.

Figure 9: YIN ALG Formulas Used for Calculation/Estimation of Fundamental Frequencies[9]

$$r'_t(\tau) = \sum_{j=t+1}^{t+W-\tau} x_j x_{j+\tau}.$$

$$r''_t(\tau) = \begin{cases} r_t(\tau)(1-\tau/\tau_{\max}) & \text{if } \tau \leqslant \tau_{\max}, \\ 0, & \text{otherwise.} \end{cases}$$

Synth Module:

This subsystem is meant to allow the user to select a unique harmony they would like to overlay with the input guitar signal. Although we never got to combine the MIDI note generator as originally planned due to late shipment of parts, we were able to see success in this module as well using wiring in the physical design to allow for knobs and dials to select waveforms and harmonies at the guitarist's convenience.

REQUIREMENTS AND VERIFICATIONS

Power Subsystem:

Requirements:

Since the guitar pedal is ideally meant to be as easy to plug in, we wanted 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]. To fall in line with these compatibility requirements, we set a requirement that the voltage conversion from 9V to 5V should be accurate within ±5%. The temperature of the voltage regulator (LM7508) should not exceed 40 degrees C. This was the main downfall of our project during the final demo as the power system caused the device to overheat, more specifically the Teensy 3.2 itself. We believe this was caused by a short circuit when soldering two ports together and not from any prior reasoning. We have good reason to believe this to be true because our calculation of heat dissipation outlined in the design section of the power was quantified correctly. According to that calculation, no more than 0.25 Watts of heat should have been dissipated in a proper circuit without a short.

Verifications:

We verified compatibility with standard guitar pedal power supplies by simply adding our guitar pedal to an existing pedal chain and ensuring that it ran with full functionality. We verified that the voltage conversion occurred as expected by using a multimeter to read the voltage from the DC power supply as well as the voltage from the LM7805 chip. We used an electronic thermometer to ensure that the temperature of the LM7805 did not exceed 40 degrees C. We used this same tool to measure as well when the device overheated, and we took a measurement of upwards of 60 degrees C.

Input/Output Buffer Subsystem:

Requirements:

The requirements for these subsystems were fairly straightforward and very similar to one another. We require that these modules should be able to divide and multiply respectively the peak-to-peak voltage of an input signal by a factor of three. It should have been able to do so for signals between 2V and 15V peak-to-peak voltage and we also required that the input impedance was greater than 100kOhms.

Verifications:

We verified these requirements by attaching an oscilloscope to the input and output signals, calculating the peak-to-peak voltage, doing the same for the signal after the input buffer and checking that the ratio is within ±5% of the scalar factor of three to make sure that the buffered signal was the same as the required input of the Teensy or the output of the guitar pedal chain. We chose a ±5% deviance from this as because we did not want any value to distant from 3.15V to prevent distortion of the signal. The input impedance was also measured using an oscilloscope, by using the formula Z = V/I for input signals.

ADC/DAC Subsystems:

Requirements:

The requirements for these subsystems are also very similar to one another. For both systems, we required output to be less than or equal to 1Vrms. This is because 1Vrms is slightly higher than the highest guitar output voltages. For the ADC module, we required that it should be able to convert in the frequent range of 50-3000Hz with maximum ±5% signal distortion. We chose 5% to be the bound because even with this level of noise distortion, the fundamental frequency can still be properly identified. For the DAC module, we used a tighter boundary of 1% because these are the frequencies that the user will hear, and they need to be more accurate than those required for note recognition.

Verifications:

Since the first requirement is the same for both of these modules, they also share the same verification protocol to ensure that it is met. To verify the output voltage for both of these systems, we used the serial monitor and rms voltage function in the Teensy Audio Library to calculate the rms voltage of the output. This confirmed functionality in the way that the measured RMS voltage was always under 1V. For the ADC conversion, we verified that the signal should was received by the Teensy Audio board by running the serial monitor and ensuring there was zero input from the ADC. We measured signal frequency after input buffer using an oscilloscope and the note2freq serial monitor was used to measure the signal frequency from the ADC. We verified functionality of ADC by ensuring these values were within the bound of 5%. The same verification protocols were carried out for the DAC subsystem except for a tighter bound of 1% difference.

DSP Subsystem:

Requirements:

The main requirements for this subsystem were fairly simple in that it had be able to resolve a frequency to the nearest 'correct' note within 5 milliseconds and tie break to the higher note in the case of equidistant values from the frequency.

Verifications:

This was verified by attaching an oscilloscope to the input signal from the guitar and the output from the frequency analyzer. We then used the plot of the waveforms and calculated the time difference between the starts of the two signals. We were able to verify this was working by recognizing that the correct note was being detected within a 5ms time span.

Synth Subsystem:

Requirements:

The main requirement for this module is that latency should be under 5 milliseconds. This is highly important because a high latency effect would cause distortion and manipulation of the signals in a way that would affect how the user hears the generated sound. We also required that we are able to calculate at least the following harmonies: unison, fifth, major third, minor third, major 7th, minor 7^{th} to allow for a wide variety of harmonies for the guitarist to choose from.

Verifications:

This module functionality was verified in a similar fashion to the DSP module by checking the time difference between the two start stages of each signal and verifying that they were less than 10ms apart. We were able to achieve this since the peaks lineup with less than 5ms of latency factor. We also verified the output notes by checking them against chord charts to verify correct harmonies.

COSTS

Desired Salary = $87,637/year for 261 workdays at 8 hours per day = $44.58/hour

Total Labor Cost = $44.58 * 56 hours * 2.5 = $6241.20

****Cost Table by Part and Manufacturer found in Appendix ii****

CONCLUSION

Accomplishments and Future Plans:

Overall, we were able to successfully develop this device in which we achieved great sound output and were able to see the success of each of our subsystem implementations. We were also able to successfully complete our high-level requirements in that we were able to 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. We stayed within the bounds of error tolerance of +/- 5%. Both the ADC/DAC modules were able to sample at a rate of at least 44.1 kHz and at least 16-bit resolution. We certainly made the overall music production and performance significantly more convenient for the guitarist. Finally, our total end-to-end latency was under 10 milliseconds. In the future, we hope to extend the functionality of our project by allowing the user to select a specific scale on top of the harmony and waveform that is already implemented. Also, we could improve on this project by including harmonies that are diatonic so that they would reach a new level of accuracy no matter what octave the harmony falls in.

Uncertainties and Challenges:

The biggest challenge we had was finding a PCB that would work with the changes we made to our design. Many times, we had to reorder a PCB due to problems with the layering or because the design would not get approved. Another big challenge we had during the final demo was the power subsystem overheating and causing the Teensy microcontroller to malfunction. This uncertainty still remains but a quantitative explanation we have is that a short occurred during the soldering process and this caused the device to overheat.

Our Project and the IEEE Code of Ethics:

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]. We considered 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. There is also concern that because we are using circuitry our design should be 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. 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.

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] “The YIN Algorithm Documentation” 2006

http://mroy.chez-alice.fr/yin/index.html

[10] "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

[11] 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

[12] Mottola, "Table of Musical Notes and Their Frequencies and Wavelengths" 2020,

https://www.liutaiomottola.com/formulae/freqtab.htm

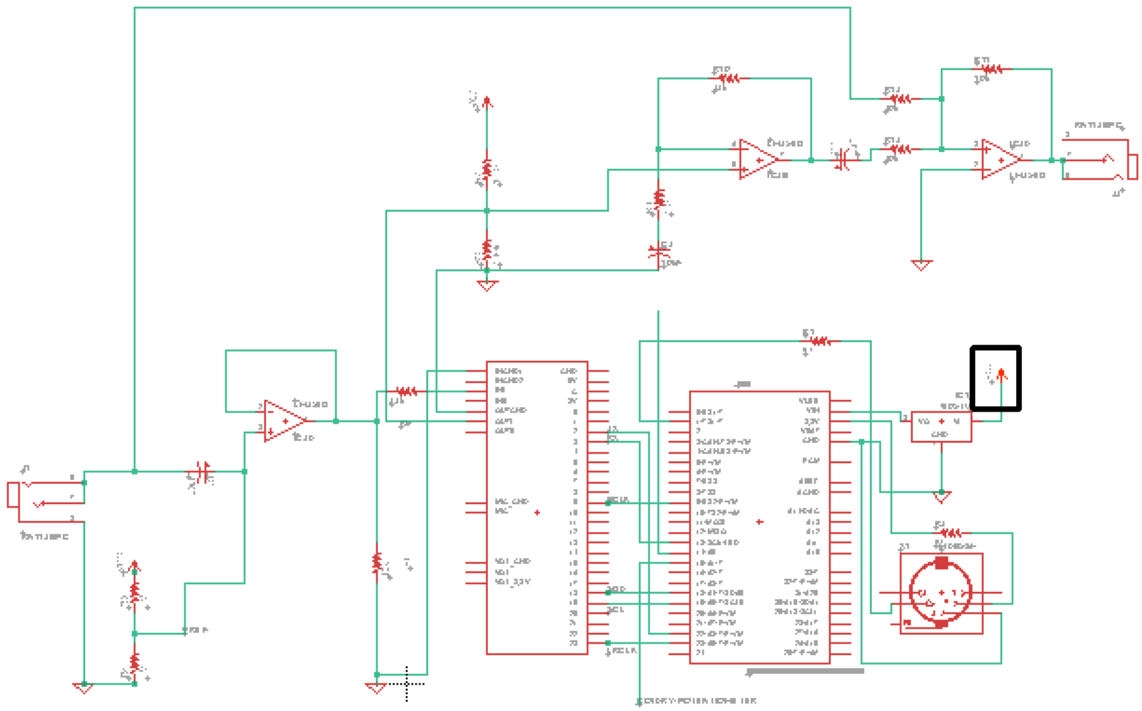
[13]” Tom’s Guitar Projects”, "Electric Guitar Output Voltage Levels" 2014,

https://tomsguitarprojects.blogspot.com/2014/12/electric-guitar-output-voltage-levels.html

[14] David Herres, "Waveforms in oscilloscopes and elsewhere" 2017,

https://www.testandmeasurementtips.com/waveforms-oscilloscopes-elsewhere/

Appendix I: Circuit Schematic Diagram for Guitar Pedal



Appendix II: Cost Table

Part	Part Number From Manufacturer	Manufacturer	Quantity	Cost/ unit
Teensy 3.2	DEV-13736	PJRC	1	19.80
Teensy Audio Board	SGTL5000	PJRC	1	17.95
Black Knob	COM-09998	SparkFun Electronics	2	4.75
Rotary Potentiometer	COM-09939	SparkFun Electronics	1	4.75
Hook-Up Wire- Stranded, 22 AWG	PRT-11375	SparkFun Electronics	1	20.95
Hook-Up Wire- Solid Core ,22 AWG	PRT-11367	SparkFun Electronics	1	19.95
DIP Sockets Solder Tail	PRT-07937	SparkFun Electronics	1	0.50
Electrolytic Decoupling Capacitor	COM-00523	SparkFun Electronics	5	0.45
Resistor Kit	COM-10969	SparkFun Electronics	1	7.95
Break Away Headers	PRT-00116	SparkFun Electronics	1	1.50