ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

Self-Adjusting Volume Pedal

Team #34

NOAH DUVAL
(nbduval2@illinois.edu)
CHRIS JURCZEWSKI
(cmj7@illinois.edu)
NORBERT LAZARZ
(nlazarz2@illinois.edu)

TA: Shanthini Praveena
Purushothaman, Nithin

May 1, 2024

Abstract

This document is a report covering Team 34's senior design project during the Spring 2024 term of ECE 445. It explains the need for our product as well as the proposed solution to be implemented. Exhibited is a complete breakdown of all modular aspects of our product, from our initial design and block diagram to the prospective functions and concepts necessary for a successful project. We explain the explicit details of our schematic and printed circuit board, as well as the digital and analog components in every subsystem, in great depth. We model future improvements alongside our gathered data and verifications.

Contents

1 Introduction

1.1 Problem

Electric guitars are versatile instruments with a tremendous amount of options for what tone. Generally, there are two ways to majorly affect a guitar's tone, those being the EQA (i.e. Treble, Mids, and Bass) on the amplifier, and having pedals that have their own specialized effects built in. Pedals give an almost infinite set of combinations for sound alteration like delay, distortion, phasors, reverb, and more. Finding a tone that matches a certain playing style can be very difficult and time consuming. Therefore, after finding a specific tone, recreating it at any volume level is the next step. Guitarists encounter a problem when they only having two options of volume adjustment on a typical guitar/amp setup. The first involves using a volume knob on the guitar, where there is often a very noticeable change in tone especially in the lower range. The other option, the volume of the amplifier, incurs a similar problem, however the effect is more noticeable. On the amplifier, the increase in volume is not linear and behaves exponentially, saturating in the middle to upper ranges. On top of fluctuation in tone, these options are imprecise and impractical when attempting to maintain a specific tone. During performances especially, guitarists want to be able to slightly adjustment how loud the guitar is compared to the rest of the performers, without changing the preset tone.

1.2 Solution

To allow for a more precise and consistent way to adjust a guitar's volume without needing to further adjust the tone of a guitar, this project would consist of a self-adjusting volume pedal that could calculate the gain necessary to keep volume constant for a guitarist for all locations in a given setting. To achieve this goal, there would be three components of the whole system, one for capturing volume data, one for processing it, and the other responsible for proper amplification. The first is an attachment on the bottom of the guitar's body consisting of a microphone to collect audio data from an amplifier. This would interact with the second component of a microcontroller by sending data to it. Flashed onto the MCU will be a program to continuously receive and modify the audio signals for incrementing or decrementing a digital potentiometer. The final component of the volume pedal is a circuit responsible for amplifying the analog signal with operational amplifiers and various capacitors, resistors, and the aforementioned digital potentiometer. All of these systems would work together to make it so a guitarist has the ability to set a volume at a given location and hear that same volume no matter where they played in a room. This would allow for the increase or decrease in volume, without effecting the tone, by simply walking towards or away from the amplifier.

2 Design

2.1 Block Diagram

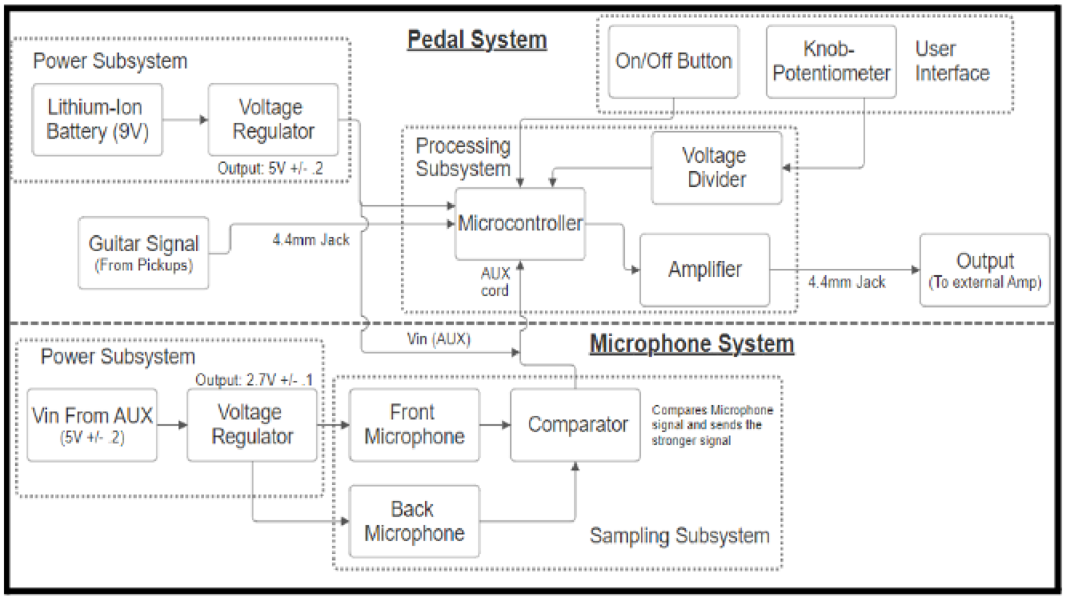

Figure 1: Block Diagram

The way that we originally set out do this can be seen in our block diagram. The plan was to have two microphones on the guitar with one on the front and back. That signal would be passed through a comparator allowing the stronger and or closer facing signal to be sent to the micro controller. Also sent to the micro controller was a voltage value from a volume knob operated by the user. Inside the code was to analyze the two signals to gain a desired decibel value that the volume should be set to. As well as, filter out frequencies outside of the range of a guitar (80-1500Hz) from the microphone signal in order to read a decibel value to understand the current volume. Once both of those values are found the system is then able to know if it should turn the volume up or down. All while trying to achieve three main goals. To keep the volume within five percent of the desired value, to not change the tone of the sound, and to avoid frequencies outside of 80-1500Hz to have an effect.

As the project progressed there was much for us to learn and understand. While the core plan did not change there was a few changes made to the way we implemented certain portions of the project. The original plan was to do the filters and a majority of the amplification digitally. However, as time went on we slowly started to switch to analog implementation. With all of us being Electrical Engineering it was something that we are all more familiar with and better understand than the digital approach. Outside of that the design remained as originally planned.

2.2 Design Details

2.3 Design Procedure

When deciding on how to design this project there were 3 major Subsystems that needed to be implemented for the project to work properly. These three were the digital processing, analog processing, and power subsystems.

- The Digital processing subsystem was responsible for taking input from microphones on the guitar and filtering it to get a signal that can be useful for determining volume levels. This filtered signal would be used to set the digital potentiometer in the analog processing subsystem.
- The Analog processing subsystem is responsible for amplifying the original guitar signal and adjusting its amplitude via digital potentiometer to get the desired volume output.
- The Power subsystem is responsible for providing the necessary power to all components of the system, ensuring smooth operation.

2.4 Circuit Design

The overall circuit consists of multiple parts connected to one another. Due to the nature of this project and it constantly evolving the final pcb design was unable to keep up. Thus, in the final version the pcb multiple test points were added to allow access to key signals. This made it possible to add additional circuits such as the analog filter. Also, giving the team freedom to skip or adjust circuits that were no longer need.

2.4.1 Schematic

In the final pcb schematic there are six main sections. The micro controller, debug header, guitar adjustment, user input, on/off switch, and the I/O of the audio. Everything on the pcb is powered by a 9V lithium ion battery that runs through a 3.3V linear voltage regulator. Along with the micro controller is the needed circuits found in the data sheet to prevent decoupling and allow for boot loading as well as resetting the MCU. A debug header was also added to allow for JTAG to be used for debugging and programming proposes.

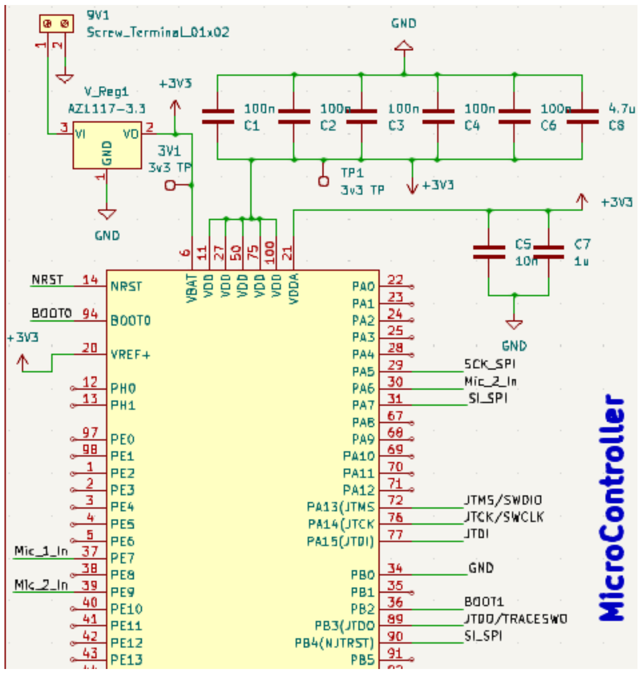
Figure 2: Micro Controller Schematic Part 1

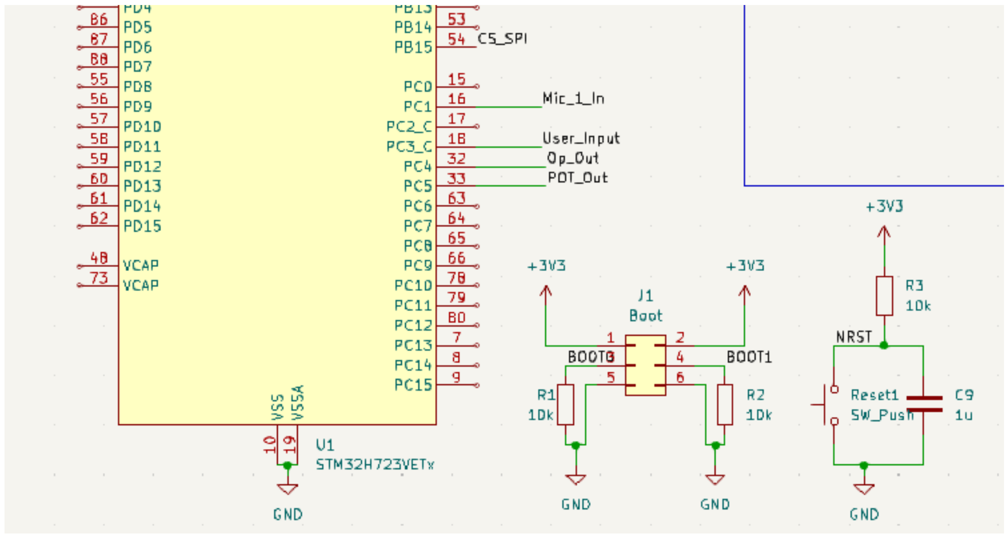
Figure 3: Micro Controller Schematic Part 2

The guitar adjustment consists of a digital potentiometer that is tied to the mcu through Serial Peripheral Interface connections. The potentiometer consists of eight pins three of which are used for SPI protocol. There is a pin for chip select, the clock, and one for a data line. The data line is shared for the MISO and MOSI lines as it contains an internal circuit that is able to switch between signals going out and coming in. It is able to operate in SPI modes ”00” and ”11”. In this design ”00” was used as it is the most common as the data is sent on the falling edge of the clock. At this time the current plan was to utilize the internal op amp on the mcu. However, later it was deemed simpler to do it physically which is explained in a later section.

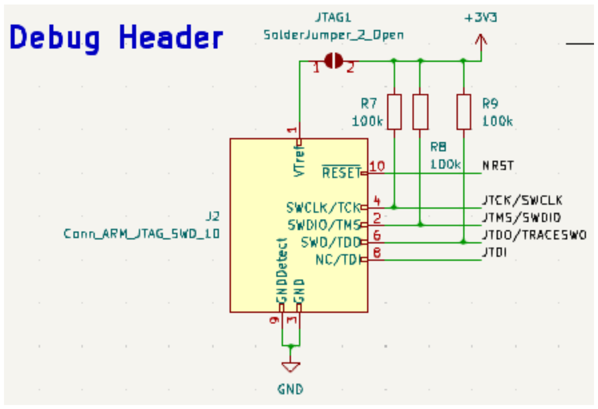

Figure 4: Debug Header Schematic

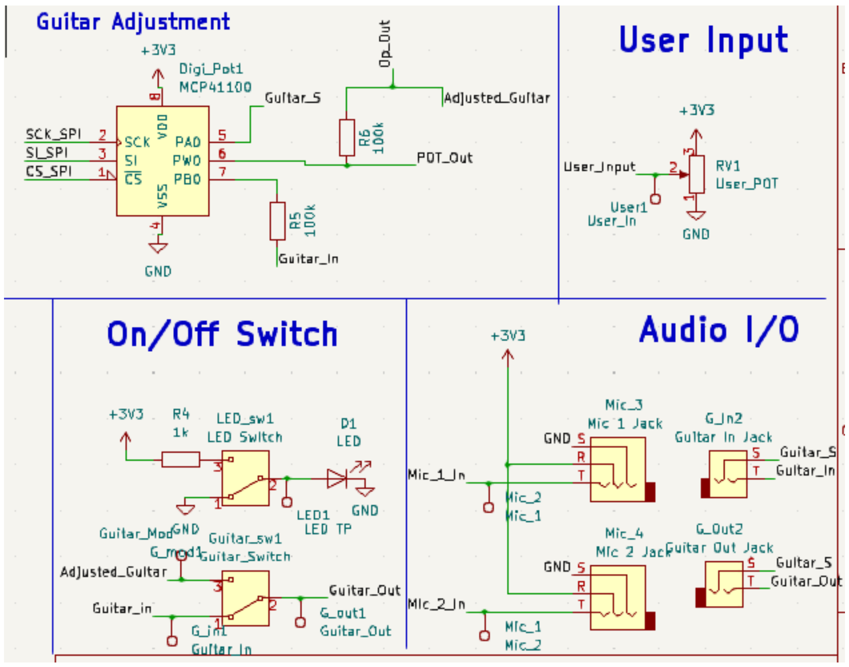

Figure 5: Guitar Adjustment and User Interface Schematic

The user input circuit is a simple 100k Ohm potentiometer that sends a variable voltage value to the mcu to be read and converted to a decibel value. Along with that goes the on/off portion. Originally the plan was to use switches. However, it was decided it would better to use a push button containing three single pole double throw switches. With one changing the guitar output from the adjusted signal to the original another to control the led and leading the third left unused. Lastly, is the in and out of the audio signals. The microphones consist of a TRS connection with the tip being the signal, the ring being power for the microphones, and the sleeve for ground. For the guitar the connection consists of a ST connection with the tip contain the signal and the sleeve for ground which is routed from one to the other.

2.4.2 PCB Design

The pcb is designed to be placed in the casing with a majority of the connections made using external wires. The rest is made up of surface mount components. The sizing of the tracks used changed by the signal being sent through. For power 20 mil tracks are used and 10 mil tracks for communication signals. With the grounds are connected using copper zones inside allowing for a reduced number of tracks used. As seen in the Figure 7 below, the 100 pin QPF micro controller is placed in the center with all the necessary components surrounding it. The external connections are also displayed with some being outgoing wires and other male headers allowing for female connector to be attached. This allows for the pcb to have flexibility in terms of how the team decided to place pcb inside of the casing. On the right hand side of the pcb are the previously mentioned test points which was used access to the inside of the circuit in order to add and bypass circuits.

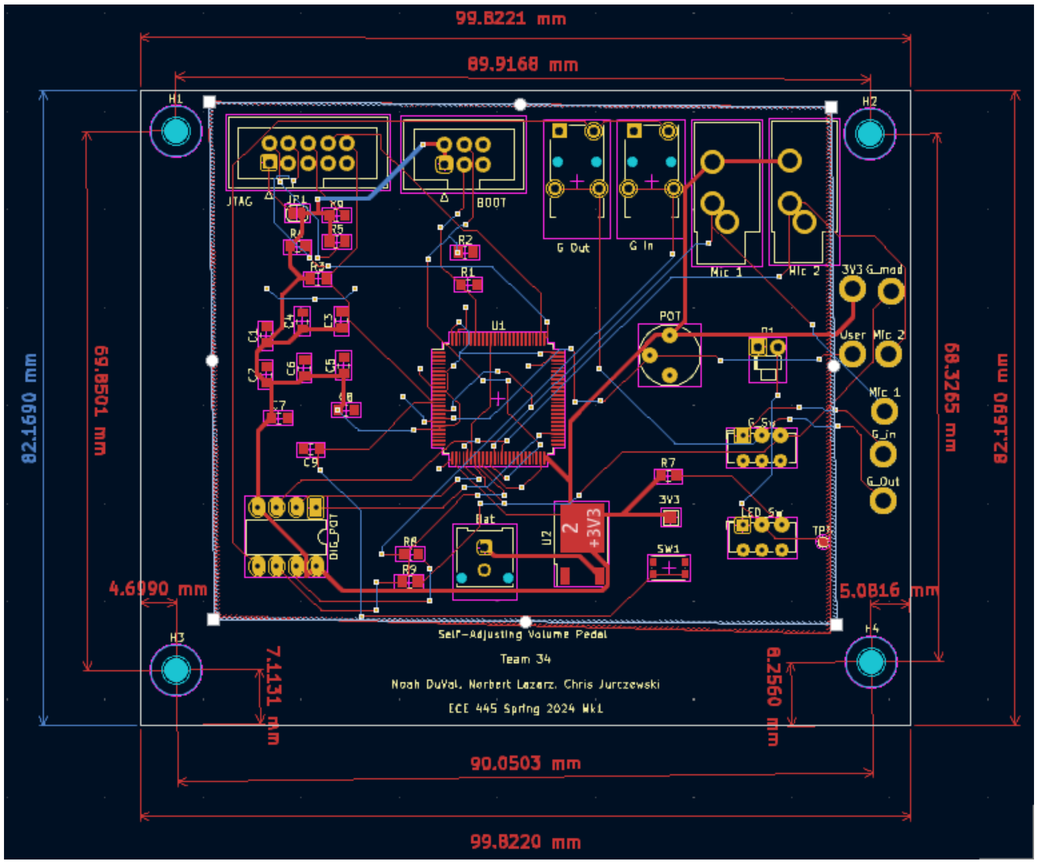

Figure 6: PCB

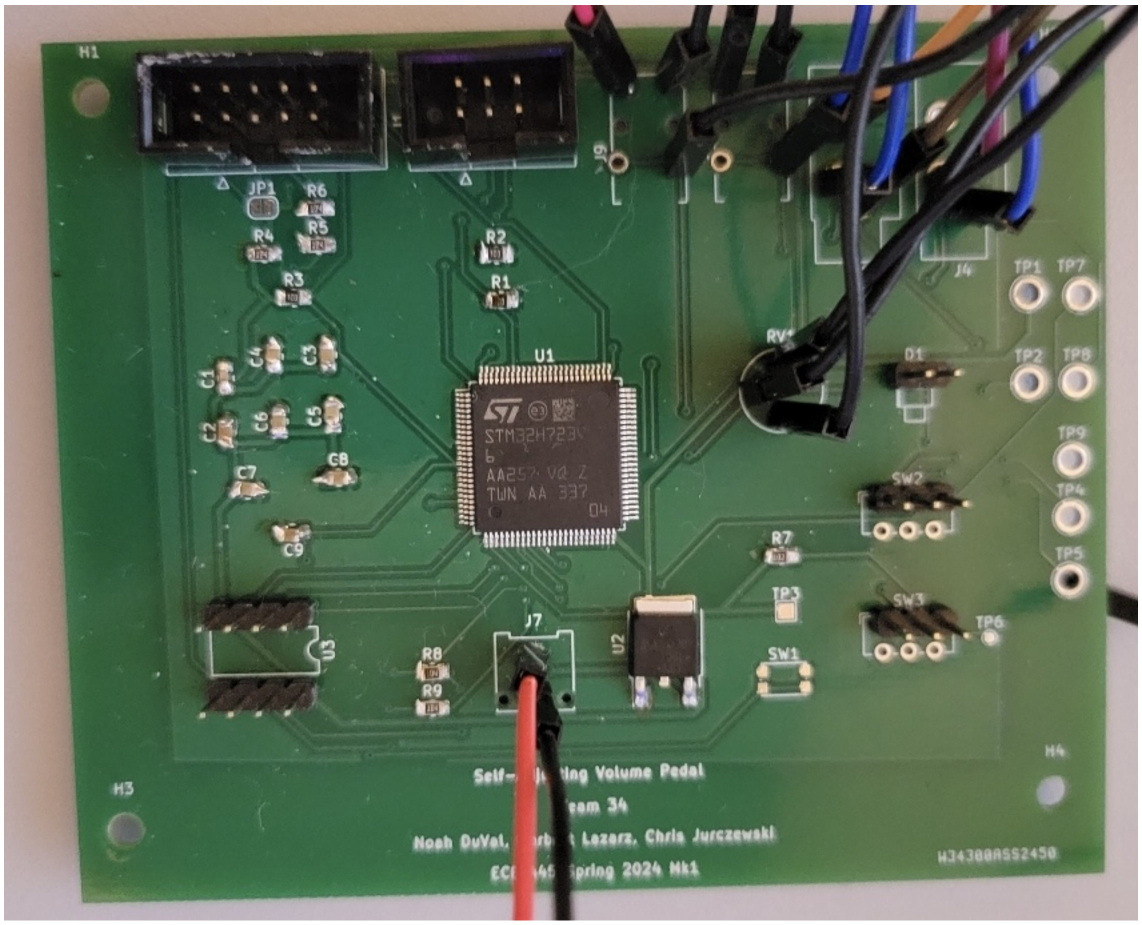

Figure 7: Physical PCB

2.5 Filter Design

2.5.1 Analog Filter

The analog filter design consisted of several low pass filters as well as a standard variable amplifier using a potentiometer and op-amp, as well as an amplitude adjuster using a digital potentiometer. When the initial guitar signal gets received it goes through a low pass filter with a 3dB cutoff around 80 kHz. Prior to amplification a smaller capacitor was used to not attenuate the amplitude of the signal since it is very low when initially coming from the guitar. Then the signal is amplified using the op-amp which has an adjustable gain from 1 to 2 times the original signal.

As shown in Figure 8, the amplifier is followed by another low pass filter with a 3dB cutoff of around 8 kHz to remove any noise created by the amplifier. This is then sent to the digital potentiometer where the guitar signal can range between 0 and 1 depending on what the control of the digi-pot is set to by the microcontroller. Figure 9 is an analog equivalent circuit followed by another low pass filter.

The numbers 5, 6, and 7 correlate to the pin assignments on the digi-pot. This works

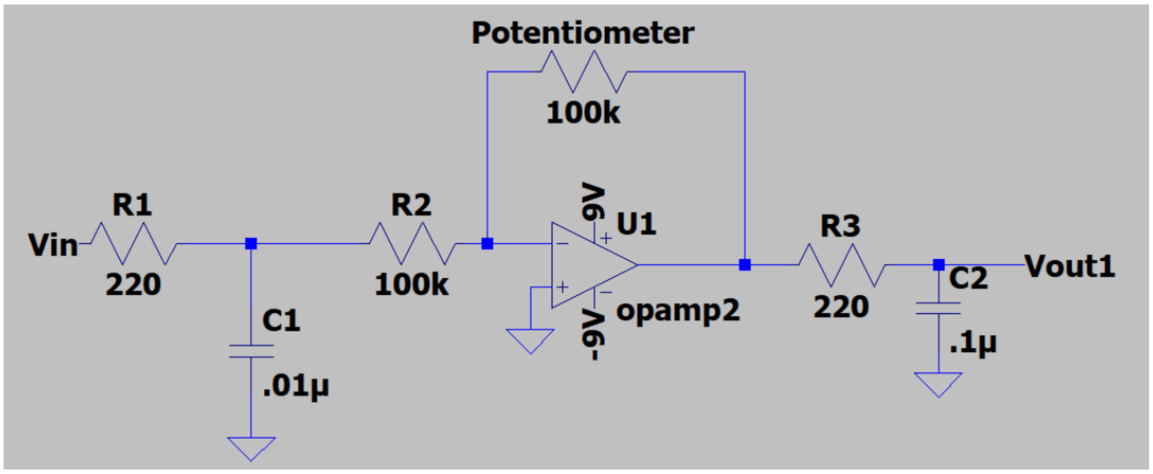

Figure 8: Amplifier Circuit

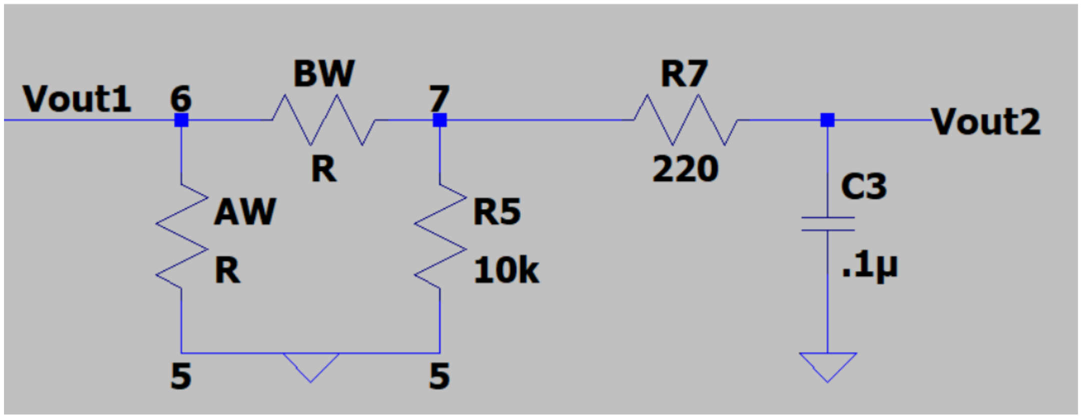

Figure 9: Digital Potentiometer Circuit

similarly to a volume knob that uses an analog potentiometer, however, this design allows the circuit to be adjusted using a program to control the digi-pot value, which allows for automatic adjustment.

2.5.2 Digital Filters

One of the filters we had hoped to implement in our final product was a band-pass filter that would attenuate frequencies picked up by the microphone outside of the specified range of 80-1500 Hz, as shown in Figure 10. This range was chosen as it is the typical frequency range outputted by a guitar, so this filter would attempt to eliminate outside noise from the system. This filter would be placed as the first step when sampling ADC data. The audio signal from the microphone would be collected through an ADC channel on the MCU, where it can then be modified by this filter.

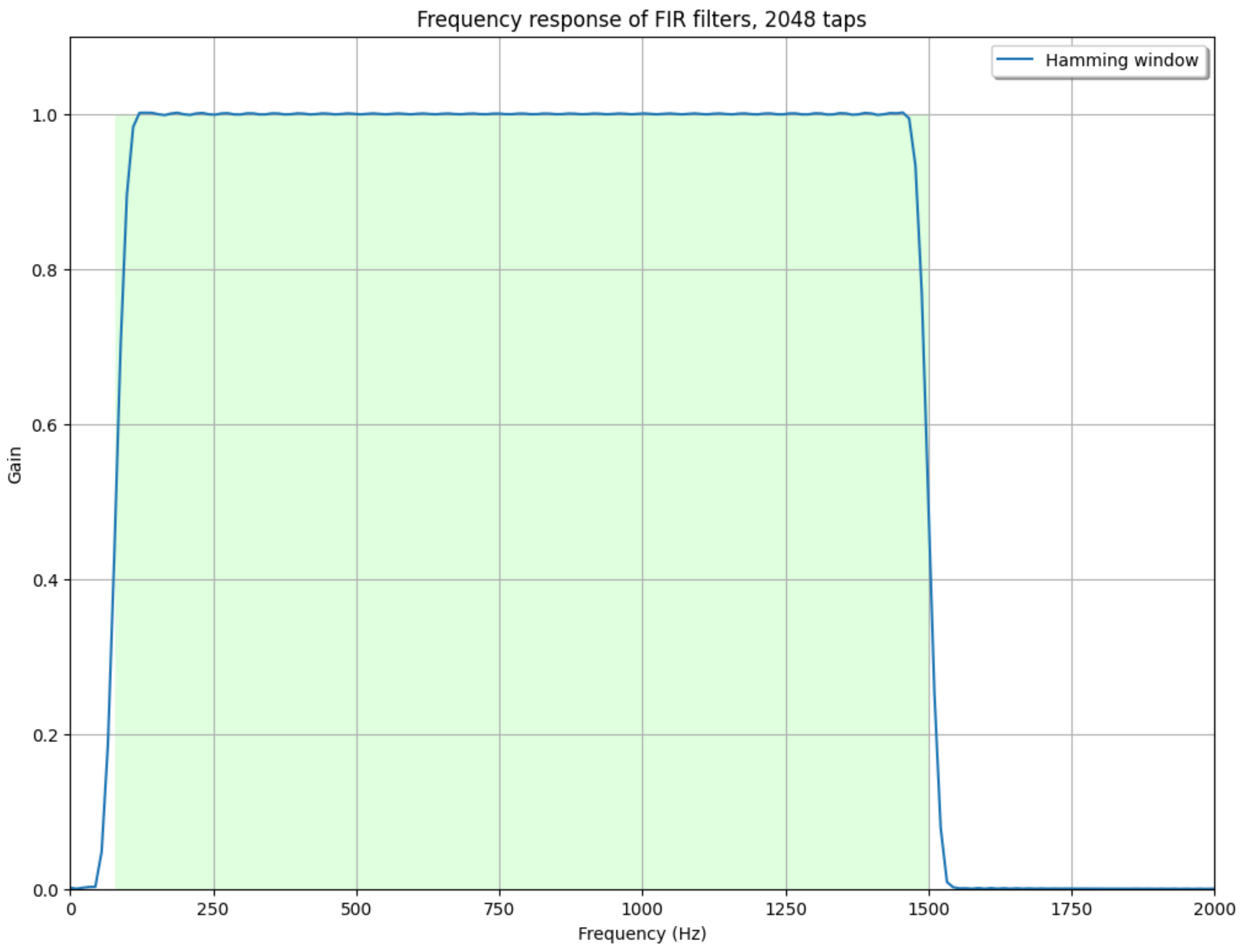

Figure 10: Digital Band-Pass Filter

The other digital filter we had hoped to implement is an envelope filter. The envelope of a signal outlines the variations in amplitude of a signal over time. The envelope can be pictured as a line drawn above the wave, following the peaks and troughs but not capturing the rapid oscillations within each cycle, like in Figure 11. After a signal would make it through both filters, it would be ready to help calculate the necessary control signal for the digital potentiometer.

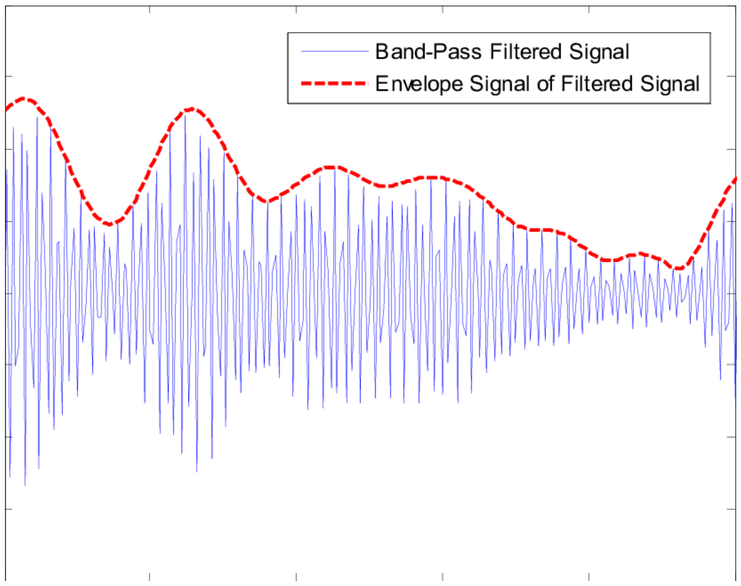

Figure 11: Envelope of a Signal

2.6 Microcontroller Implementation

2.6.1 Choice of Microcontroller

The microcontroller is the heart of the processing subsystem in this project, and is responsible for all of the digital components. Without a working microcontroller, this project was not able to be self-adjusting, but able to be implemented with further work applied. To begin, our group researched MCUs that are most advantageous for audio signal processing. The STM32 is a 32-bit microcontroller integrated circuit produced by STMicroelectronics. Our group chose to include the STM32H723VET6 on our printed circuit board. While debugging and testing our product, we were using the STM32F103RB that is integrated on a Nucleo development board. The differences between these two microcontrollers is laid out in Table 1. The H7 series of microcontrollers, which we have on our PCB, is a much more powerful unit compared to the F1 series. The H7 can sample ADC signals much faster and with higher resolution, which would have been helpful for getting meaningful data for the digital potentiometer control. Our group utilized an Integrated Development Environment, STM32CubeIDE, to program the development board. We were able to simulate an F103RB microcontroller with an ADC channel and SPI control, the only necessary pieces we needed. The software automatically populated clock configurations and other pin assignments, so we only needed to implement a few functions. One function set the value of the digital potentiometer through SPI control.

If our team had gotten a working set up for the MCU, all we would have needed to do was transfer the flash program from the F103RB on the development board to the PCB MCU

Table 1: Comparison of MCUs

MCU	H7 Series	F1 Series
Core [Hz]	600	72
Flash Memory [Mbytes]	1	0.128
ADC frequency [MHz]	50	14
ADC Resolution	16-bit	12-bit

through the use of JTAG. The pins assignments on both boards were similar enough, and we could have changed them before transfer if needed.

2.6.2 ADC Sampling

The ADC channel was the only input necessary for the instance of our microcontroller. This channel interacts with the audio signals delivered by the microphones by way of an AUX cable. In the STM32F103RB, the frequency of the ADC channels are 14 MHz. By use of equation 1, we can calculate the sampling frequency of each 12-bit signal with the addition of two other factors: (1) Sample Hold Time and (2) ADC Conversion Time.

$$f_{sample} = f_{ADC}/(SHT + ACT + 1) \tag{1}$$

The Sample Hold Time, or SHT for short, is the amount of clock cycles necessary for the ADC channel to acquire a stable sample voltage. During the sampling phase of this process, a switch inside the ADC connects the analog input signal to a capacitor, Csh. This capacitor charges to the voltage level present at the analog input at that specific instant. During the hold phase, the switch disconnects the analog input from Csh. The capacitor then holds the sampled voltage while the ADC performs the conversion from analog voltage to digital value. In STM32CubeIDE, this parameter was set to 1.5 cycles base and that is what we kept it at.

The other parameter is the ADC Conversion Time, or ACT for short. This is rather simple as it takes n number of cycles for n number of bits. In our case, our ACT for each signal was 12 cycles.

To compute the total sampling frequency, we plug in the values for each parameter into equation 1, getting a value of 966 kHz. This means it takes just over a microsecond for the MCU to receive each ADC signal.

2.6.3 Digital Potentiometer

The sole component to be modified by our MCU was the 10 kΩ digital potentiometer. This piece of the project would interface the digital processing with the analog amplification applications. The potentiometer would be programmed through SPI, where we could change the control value and thus resistance across the terminals of the component. There is a digital wiper in the component that can control out of the 128 internal resistors, how many would be powered as a given t=moment in time. The internal step resistance, Rs, is 78.125 Ω. To calculate the resistance across the potentiometer at a given control input, n, is shown in equation 2. When n is maxed out at a digital value of 127, the potentiometer is at full resistance.

$$R_{pot} = R_s n \tag{2}$$

Through the use of a digital potentiometer, we could have been able to modify the output amplification circuit by way of the ADC data. Data sampled would first go through both implemented filters. A relationship between decibel volume and change in envelope voltage would be used to increment or decrement the control signal of the digital potentiometer. On the analog side, this would then lead to more or less of the original guitar input to the pedal making it through the circuit and out to the amplifier/speaker.

In totality, the MCU was responsible for communication between the all other subsystems. It worked with the sampling subsystem by taking in 12-bit ADC signals and filtering them. It worked with the processing subsystem by control of a digital potentiometer. The power subsystem interacted with the MCU, which would have powered the microphone as well.

3 Verification

3.1 Power Subsystem

For this product the power requirements are fairly low. The system in entirety powered by a 9V lithium ion battery that can be disconnected and recharged using a micro-usb chord. The 9V is then put through an AZ1117 3.3V linear voltage regulator. This 3.3V is used to power micro controller as recommended by the data sheet. As is the two microphones and the LED.

3.2 User Interface Subsystem

The user interface consists of a push-button and a knob potentiometer, which comes standard on most guitar pedals. The push-button is responsible for turning the system on and off which entails switching between the guitar signal being altered or just going through a wire to the external amplifier. There is also an LED circuit that lights up when the system is on and turns off when the system is turned off. An example of the push-button configuration can be seen below:

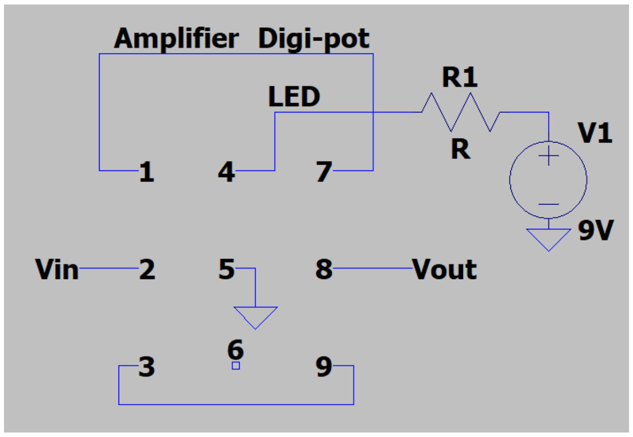

Figure 12: Push-button Circuit

The push-button has 9 pins which are separated by column creating the equivalent of 3 switch options. When the button is pressed the device switches from the top two rows and the bottom two rows. In the figure this is used to connect the V_{in} which is the guitar signal to V_{out}. The button then switches between the guitar signal going through the amplifier and digi-pot circuits when it is using the top two rows (On), and shorting V_{in} and V_{out} which is done when using the bottom two rows (Off). This also correlates to the LED, as when the system is off the node 4 becomes an open circuit causing the LED not to light up.

The knob potentiometer is available to the user and has the purpose of setting the amplification in the analog processing subsystem as well as setting the baseline voltage that the micro-controller will use to set the desired dB level.

3.3 Processing Subsystem

With our initial design of this project, the processing subsystem was responsible for filtering the received signals in real time while modifying them properly for amplification. After we had decided to implement an analog amplification circuit through the use of a digital potentiometer, this changed. The processing subsystem would then only be required to receive audio signals in real time, attenuate unwanted frequencies, and control a digital potentiometer value. These were all done though the use of a microcontroller and a flashed program. The MCU would take use of the two digital filters, the band-pass and envelope, to gather proper audio data. Then, we would calculate the necessary resistance of the digital potentiometer to maintain a gain equal to where the envelope of the siganl was going. This would have been perfected though tests of a working system.

With the development board and a simple virtual workbench, we gathered data on the microphones audio signals. We found that the microphone outputted a mean voltage of 1.654 volts when powered at 3.3 volts. The amplitude of the signal when played at drastically different decibel volumes, around 50-100 dBs, was on the order of only a few millivolts. The peak to peak voltages we were reading were difficult to replicate reliably and tabulate concrete data for the digital potentiometer control. This was one of the main factors that restricted us from creating a practical self-adjusting volume pedal, so were were unable to verify any of the results we set out to with this processing subsystem.

3.4 Sampling Subsystem

Our initial design of this project involved more components for this subsystem than we ended up with. Originally, we wanted to implement a dual microphone system on either side of the guitar, since each microphone would unidirectional in picking up sound. A comparator would be used in the MCU to choose the louder of the two signals for better accuracy in level of sound at the guitarist's location. Later on, we were unable to get a working MCU which rendered the sampling subsystem useless.

The microphones we chose to implement into this design were MAX4466 Electret Microphones. These are capable of delivering a 200kHz gain bandwidth, so they easily would work in our application of them. When initially testing the devices on a virtual workbench, we found out that the supplied voltage to the device did not affect the output voltage peak to peak, only the mean voltage. We chose to use a 3.3 volt input as this is what would be coming out of our 3.3 volt regulator on the PCB.

4 Costs

4.1 Labor Cost

The average Electrical Engineer salary for our skill sets is around $50 per hour. The length of this project is around that of eight weeks with the workload being around 12 hours per week. After accounting for overhead the total labor cost comes out to 50 * 2.5 * 12 * 8 = $12,000 per group member leading to a total labor cost of $36,000 for the whole team.

The machine shop also had some cost with the casing included in the parts list below. Due to the relatively simple machining of the project the cost would be estimated at another 3 hours adding another $150 of cost.

4.2 Part Cost

Parts List			
Description	Manufacturer	Quantity	Price
STM32H723VET6 Microcontroller	ST	1	$10.81
Push-Button Switch	Vicasky	1	$8.59
AZ1117-3.3 Fixed Voltage Regulator	Diodes Incorporated	1	$0.37
MAX4466 Electret Microphone	MAXIM	2	$12.52
Decibel Meter	TopTes	1	$29.99
Rechargeable 9V Lithium Ion Battery	EBL	2	$13.99
10nF Capacitor SMD 0805	KEMET	1	$0.18
1uF Capacitor SMD 0805	Samsung Electro-Mechanics	2	$0.20
100nF Capacitor SMD 0805	Quantic UTC	5	$1.00
4.7uF Capacitor SMD 0805	Samsung Electro-Mechanics	1	$0.10
Red LED	SunLED	1	$0.28
IDC-Header 2x03	Sullins	1	$0.36
IDC-Header 2x05	Sullins	1	$0.36
TRS Audio Jack	CUI Devices	2	$2.52
TS Audio Jack	CUI Devices	2	$2.52
Screw Terminal 01x02	Phoenix	1	$0.40
SPDT Switch	C and K	2	$1.22
10k Resistor SMD 0805	Yageo	5	$0.30
100k Resistor SMD 0805	Yageo	3	$0.30
1k Resistor SMD 0805	Yageo	1	$0.10
100k Potentiometer	Precision Electronics	1	$18.15
AL-48P Diecast Aluminum Enclosure	Polycase	1	$23.88
Total			$166.91

Table 2: Parts List

4.3 Total Cost

All said and done, including both parts and labor the total cost comes down to $36,166.91. However, if made into mass production a possible retail value would be in the neighborhood of $300.00. This would make the product only a little bit more expensive than the normal high end volume pedal.

5 Conclusions

The final product that was created has a great deal of potential. It is still in the early stages of its prosperity. The current product and design will accomplish what needs to be done, however, this a fair amount of room for improvement in most areas. For starters the PCB can be condensed and the circuits created after the fact can be added to it reducing clutter inside the casing. The microphones initially chosen did not obtain the signal quality that was expected. Thus, higher quality microphones can replace the old ones allowing for smooth signal processing. In addition, there was issues with electromagnetic interference which has many simple solutions and will be fixed in the future. With all of these being simple fixes the future of the product is bright as it will only get better.

Looking at things from an ethics stand point the goal of this project is to provide a quality of life product that does not replace or make obsolete other products attempting to solve similar issues. In section I.3 it states, “to avoid real or perceived conflicts of interest whenever possible, and to disclose them to affected parties” [1]. To ensure this doesn’t happen we intend to make this product to solve unique use cases and to be used alongside existing products that also deal with volume alteration. Also to comply with section I.5 of the IEEE code of ethics we intend to credit all preconstructed systems that use or take inspiration from. We also intend to be fully transparent with any shortcomings of this project and won’t attempt to cover up or omit information regarding any faults.

On the topic of safety, this system will be using a Lithium Ion Battery contained in the guitar attachment. We intend to make this system as safe as possible mainly by not having the recharging system contained in the attachment. Recharging will be done externally using the proper equipment. The attachment will also allow adequate airflow as to not allow for the battery to overheat while contained within the system. Other concerns are, “Thermal runaway may be triggered if a battery has certain defects that can lead to short-circuiting, is overheated, is subject to high pulse power usage, or is punctured” [2]. To solve this the placement of the battery will also not facilitate any high heat environments and will not be in danger of suffering water damage or puncture.

References

[1] IEEE. “”IEEE Code of Ethics”.” (2016), [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 02/08/2020).

[2] L. Kong, C. Li, J. Jiang, and M. G. Pecht, “Li-ion battery fire hazards and safety strategies,” *Energies*, vol. 11, no. 9, 2018, ISSN: 1996-1073. DOI: 10.3390/en11092191. [Online]. Available: https://www.mdpi.com/1996-1073/11/9/2191.

Appendix A Abbreviations

Unit or Term	Symbol or Abbreviation
analog-to-digital converter	ADC
decibel	dB
equalization	EQ
hertz	Hz
joint test action group	JTAG
kilohertz	kHz
megahertz	MHz
microcontroller unit	MCU
ohm	Ω
printed circuit board	PCB
volt	V