Guitar Entertainment System

Monte Martin III, Rishabh Ruikar, Rawan Ibraheem

Concept of Operations

CONCEPT OF OPERATIONS
FOR
Guitar Entertainment System

TEAM <28>

APPROVED BY:

Monte Martin III **9/14/2023**
Project Leader Date

Prof. Kalafatis Date

T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
-	9/11/2023	Rishabh Ruikar		Draft Release
2	12/2/2023	Rishabh Ruikar		Final 403 Release
3	4/30/2024	Rishabh Ruikar		Final 404 Release

Table of Contents

List of Tables

No tables in this section.

List of Figures

1. Executive Summary

The Guitar Entertainment System is an innovative integration of an amplifier, signal processing pedal, and a mobile app with Bluetooth connectivity. Instead of relying on traditional power management methods, the app acts as a centralized control hub, enabling users to wirelessly customize both the amplifier and pedal's effects and settings. The amplifier utilizes an MCU, DSP, DAC, and ADC for advanced audio processing, while the pedal, equipped with its DSP, provides an interactive interface for effect adjustments—all of which are easily manageable through the app. This project seamlessly merges cutting-edge technology with the realm of musical expression, with the primary goal of delivering a dynamic and immersive playing experience to musicians. The user-friendly interface is designed to cater to individuals with minimal knowledge of guitar amplification systems, allowing them to experiment with various sound effects without requiring expertise in signal processing.

2. Introduction

The Guitar Entertainment System, a capstone project, blends technology with musical expression. It integrates a top-notch amplifier, signal processing pedal, and a Bluetooth-enhanced mobile app, transforming the guitar experience. Musicians can effortlessly control audio effects and settings remotely, highlighting our electrical engineering expertise and catering to modern musicians. Our goal is to revolutionize musical interfaces for a dynamic and immersive platform.

2.1. Background

For decades, guitarists have employed pedals to modify and enrich their sound, offering effects like distortion, delay, and chorus. With the advent of solid-state amplifiers, which replaced tubes with reliable and consistent transistor circuits, guitarists gained access to a clearer and more adaptable sonic palette. Now, our innovative app adds a modern twist to this evolution. By interfacing with both pedals and the solid-state amp via Bluetooth, it streamlines the process of sound manipulation. Historically, adjusting tones required manual knob tweaking, but with our app, these modifications are at a guitarist's fingertips, offering an unprecedented level of control and convenience in shaping their unique sound.

2.2. Overview

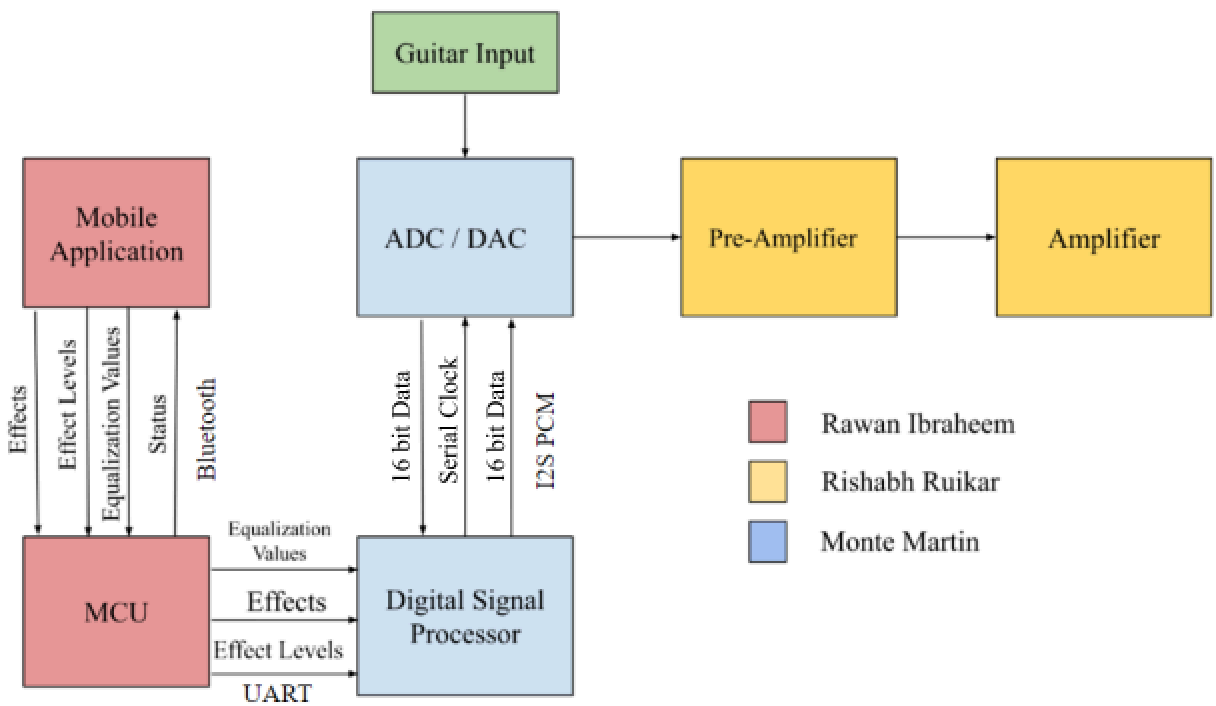


Figure 1: Guitar Entertainment System Block Diagram.

Our Guitar Entertainment System seamlessly blends traditional guitar equipment with cutting-edge technology. Its core components include a solid-state amplifier renowned for its clarity and reliability, enhancing the guitar's sound. The system incorporates pedals that introduce various audio effects, enriching the output with depth and character. A game-changing feature is the Bluetooth-enabled app, offering users unparalleled convenience in remotely controlling and customizing amp and pedal settings. This system caters to guitarists seeking an intuitive and advanced method for sound shaping, effectively merging the best of both worlds: the tried-and-true and the innovative.

Within the system the mobile application communicates sound effect levels to the MCU, which, in turn, transfers the corresponding data to the DSP. The DSP receives an input signal from the ADC in the form of 16 bit PCM data, which it then implements the effects on. It then outputs the new 16 bit PCM data out to the DAC which turns it back into analog and sends it to the preamplifier and amplifier.

2.3. Referenced Documents and Standards

The following documents, of the exact issue and revision shown, form a part of this specification to the extent specified herein:

- IEEE 802.11 Standards for Wireless and Bluetooth Communication: https://www.ieee802.org/11/
- IEEE Analog Need-To-Knows: https://technav.ieee.org/topic/analog-circuits
- TI Digital Guitar Effects Pedal: https://www.ti.com/lit/ml/sprp499/sprp499.pdf
- ESP32 Series Datasheet: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

3. Operating Concept

3.1. Scope

The Guitar Entertainment System will allow for the user to amplify their sound signal and adjust the signal to allow for effects such as distortion, reverb, and chorus. The entertainment system will be operated through an android application. This project covers both the software and hardware developments in this system. The deliverables include the amplifier, pedals, a fully functional android based mobile application, as well as a user manual. Furthermore, the device must have bluetooth connectivity. The user will be able to save presets so that they will not have to adjust the settings every time they enter the application, especially if they are repeatedly adjusting the application in the same way.

3.2. Operational Description and Constraints

The Guitar Entertainment System is designed for real-time sound modulation and customization for guitarists. Users will connect their guitars to the system, then use the Bluetooth-enabled app to select and adjust effects, modify amp settings, and even save or recall specific sound profiles. This system will be especially useful during live performances, allowing for swift sound adjustments without the need for physical contact with equipment.

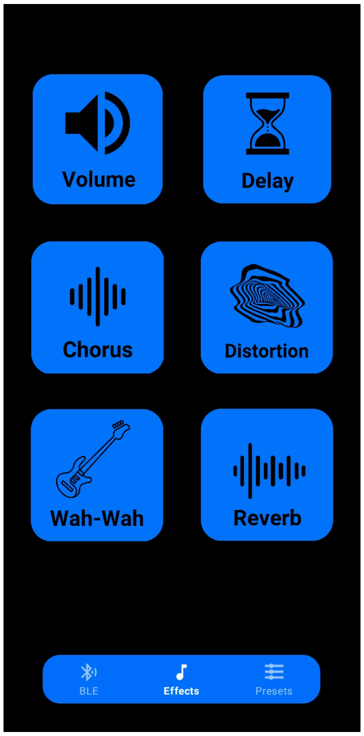
The constraints for operating the system are as follows:

- **Stable Bluetooth Connection:** A stable Bluetooth connection is essential for system operation.
- **Limited Range:** The system operates within a 10-meter range (Class 2 Bluetooth), making operation beyond this distance impractical.
- **Android Compatibility:** The mobile app must be compatible with Android devices.
- **Power Management:** Ensuring power is maintained while the app is in use is critical.
- **Maintenance Requirements:** Regular maintenance, including component cleaning and inspection, is necessary for sustained performance.
- **Budget and Time:** Parts must be budget-friendly (under $300) and available within a one-month time frame due to budget and time constraints.

3.3. System Description

The Guitar Entertainment System is an integrated solution designed to elevate the guitar playing experience. At its heart lies a solid-state amplifier, ensuring consistent and clear sound amplification. Paired with this is a pedal, which allows for diverse audio effects that alter the audio signal. The system's marquee feature is a Bluetooth-enabled mobile app, serving as a control platform. Through this app, users can wirelessly tweak the settings of both the amplifier and the pedal, customize effects, and save preferred sound profiles. Collectively, this system harmonizes traditional sound equipment with digital innovation, providing guitarists with an intuitive, modern, and versatile musical toolkit.

3.4. Users



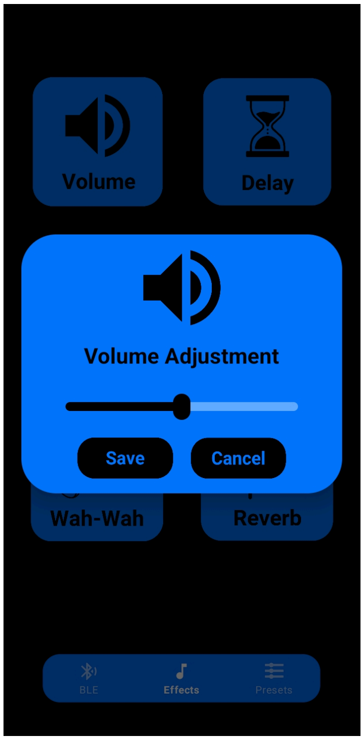


Figure 1. Effects of Sound Effect Controlling Application.

Figure 2. Sound Effect Adjustment Example.

Our Guitar Entertainment System is tailored for a broad spectrum of guitarists, from budding enthusiasts to seasoned professionals. The system is designed with user-friendliness in mind; thus, minimal training is required for installation and use. Users familiar with traditional amplifiers and pedal setups will find it intuitive, while those accustomed to digital tools will appreciate its app-based interface. Beyond the immediate musical community, sound engineers and recording studios can also benefit from the system's versatility and ease of customization. Essentially, our system appeals to anyone in the music domain seeking an amalgamation of traditional sound quality and modern control.

3.5. Support

Each system will come with a comprehensive, user-friendly manual that provides detailed instructions for setup, operation, and troubleshooting. The manual will include information about the system's components, making it easy for users to replace parts if needed. Additionally, it will offer insights into the mobile application, explaining the purpose and impact of each sound effect on the signal. This information will be valuable for users with varying levels of signal processing knowledge, enhancing their understanding of the system's capabilities.

4. Scenarios

4.1. Guitar Practice

The amplifier and pedal setup is versatile, offering the flexibility to practice guitar at a comfortable, moderate volume while maintaining clear sound quality. Even at lower volumes, the effects can be fully harnessed, enabling users to practice in a performance-like manner. Additionally, the system allows users to effortlessly create and save presets with their preferred effects during a performance, eliminating the need for time-consuming adjustments in the midst of a live show.

4.2. Performance

The amplifier is designed to deliver a powerful sound, making it suitable for live performances to small crowds. At high volumes, all the effects maintain their full functionality, ensuring users can fully harness the capabilities of the amp and pedal system during performances.

In scenarios involving multiple genres or performers, the system offers the convenience of saving distinct presets for each, facilitating quick and seamless transitions between different musical styles or artists. This feature streamlines performance preparations and enhances adaptability, catering to various performance contexts.

4.3. Recording Sessions

Musicians can use the system for recording sessions in a studio environment. The amplifier and effects offer the versatility to capture different tones and textures, enhancing the recording process.

5. Analysis

5.1. Summary of Proposed Improvements

Our Guitar Entertainment System revolutionizes the traditional guitar setup in several key ways. Firstly, it introduces wireless convenience through the Bluetooth-enabled app, eliminating the need for physical tweaking and allowing on-the-fly adjustments. This seamless integration of amp, pedals, and digital control ensures an efficient and clutter-free setup. Furthermore, the system offers a central platform for customizing and saving preset sound profiles, a boon for versatile players. By merging analog sound quality with digital enhancements, the system also promises consistency and reliability, especially with the solid-state amp design. In essence, our solution modernizes the guitar experience, making it more intuitive, adaptable, and in tune with today's tech-driven world.

5.2. Disadvantages and Limitations

While our Guitar Entertainment System offers several advantages, it's not without potential drawbacks. Relying on Bluetooth connectivity can introduce latency issues, potentially disrupting real-time play. The system's technological complexity, combining pedals, amp, and an application, might pose a learning curve for traditionalists. Additionally, as with any tech-driven solution, there's a dependence on battery life and software updates. Maintenance complexity, including the periodic need for cleaning and inspecting system components, can be cumbersome. Signal interference, often encountered in crowded or noisy environments, may affect the system's performance. Limited compatibility may restrict usage to specific mobile devices or operating systems, potentially excluding users with non-compatible devices.

5.3. Alternatives

6. Traditional Analog Setups: Using standalone amp and pedal setups without digital interfacing.
   - 6.1. *Trade-offs:* Provides tactile control and often preferred "warmth" of sound, but lacks the convenience of remote customization and digital enhancements.

7. Digital Multi-effects Processors: All-in-one units that combine multiple effects and amp simulations.
   - 7.1. *Trade-offs:* Portable and versatile, but may sacrifice the authenticity of individual pedal sounds and the tactility of standalone units.

8. Guitar Plug-ins on Computers: Software that simulates amps and effects.
   - 8.1. *Trade-offs:* Highly customizable and ideal for recording, but less practical for live performances and dependent on computer processing power.

9. Dedicated Hardware Controllers: Physical units that control digital effects, eliminating the need for an app.
   - 9.1. *Trade-offs: Offers a more tactile experience than an app but lacks the flexibility of wireless control and may require a bulkier setup.*

9.2. Impact

The production and disposal of electronic components in our Guitar Entertainment System could contribute to e-waste if not managed responsibly. Ethical concerns arise if the materials sourced for manufacturing aren't obtained sustainably or if production practices harm the environment. Ensuring responsible sourcing, energy-efficient design, and promoting recycling or safe disposal methods for end-of-life products are pivotal to address these concerns.

Guitar Entertainment System

Rawan Ibraheem, Rishabh Ruikar, Monte Martin

Functional System Requirements

REVISION – Draft
25 September 2023

Functional System Requirements
for
Guitar Entertainment System

Prepared by:

Team 28 4/30/24
Author Date

Approved by:

Monte Martin III 4/30/2024
Project Leader Date

John Lusher, P.E. Date

T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
-	9/25/23	Rishabh Ruikar		Draft Release
2	4/30/2024	Rishabh Ruikar		Final 404 Release

Table of Contents

List of Tables

All relevant tables are appropriately cataloged in the subsequent sections of this document.

List of Figures

1. Introduction

1.1. Purpose and Scope

The Guitar Entertainment System represents a groundbreaking integration of advanced audio technology, including a preamplifier, solid-state amplifier, Analog-to-Digital Converter (ADC), Digital-to-Analog Converter (DAC), Digital Signal Processor (DSP), and Microcontroller Unit (MCU). The MCU will be controlled by an Android application through a Bluetooth connection. This innovative system is aimed at elevating the guitar playing experience to new heights.

The scope of this project encompasses the design, development, and integration of these critical components, both on the hardware and software fronts. Specifically, it involves the construction of a preamplifier that conditions the incoming guitar signal, a solid-state amplifier renowned for its clarity and reliability, and the incorporation of ADC and DAC modules to facilitate high-quality signal conversion. The DSP unit is responsible for advanced audio processing, while the MCU serves as the central control unit for the system.

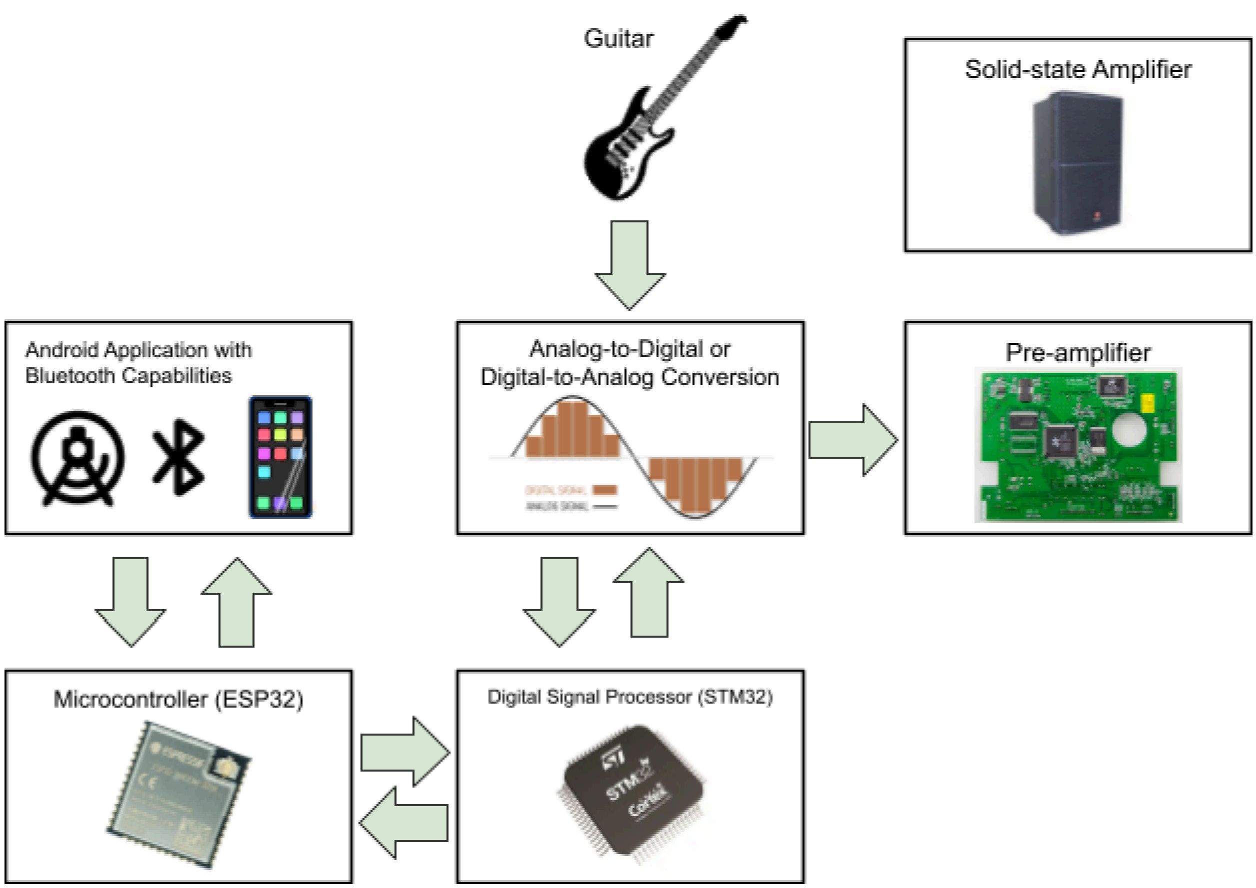


Figure 1. Guitar Entertainment System Communication Diagram

The ESP32-S3 has been chosen as the MCU for the project, and it will be programmed using Visual Studio Code with the PlatformIO extension. The Android mobile application will be developed using Android Studio, an integrated development environment (IDE) specifically designed for programming and previewing applications. As far as signals go, the sine wave from the pedals will travel to the amplifier via a ¼ inch tip sleeve. This jack will also go from the guitar to the pedals.

1.2. Responsibility and Change Authority

The team leader, Monte Martin, is responsible for ensuring all requirements for the project are met. Furthermore, Peng-Hao Huang has the authority to make changes to the plan.

Subsystem	Responsibility
Microcontroller Unit & Android Application	Rawan Ibraheem
Pre-amplifier & Amplifier	Rishabh Ruikar
ADC, DAC, and DSP	Monte Martin

Table 1. Responsibility for each subsystem.

2. Applicable and Reference Documents

2.1. Applicable Documents

The following documents, of the exact issue and revision shown, form a part of this specification to the extent specified herein:

Document Number	Revision/Release Date	Document Title
IEEE 802.15.4	6/4/2002	IEEE Standard for Telecommunications and Information Exchange Between Systems - LAN/MAN - Specific Requirements - Part 15: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs)
1910.95	8/27/1971	Occupational Safety and Health Standards - Subpart G
IEEE 1613	3/20/2003	EEE Standard Environmental and Testing Requirements for Communications Networking Devices in Electric Power Substations

2.2. Reference Documents

The following documents are reference documents utilized in the development of this specification. These documents do not form a part of this specification and are not controlled by their reference herein.

Document Number	Revision/Release Date	Document Title
ISO/IEC/IEEE 12207:2017	11/2017	Software Life Cycle Processes

2.3. Order of Precedence

In the event of a conflict between the text of this specification and an applicable document cited herein, the text of this specification takes precedence without any exceptions.

All specifications, standards, exhibits, drawings or other documents that are invoked as “applicable” in this specification are incorporated as cited. All documents that are referred to within an applicable report are considered to be for guidance and information only, except ICDs that have their relevant documents considered to be incorporated as cited.

3. Requirements

The following section defines the minimum requirements for the Guitar Entertainment System.

3.1. System Definition

Our project is a guitar sound system that contains 3 main subsystems, with subcomponents inside each subsystem.

- Amplifier
  - The amplifier will contain an amplifier, a pre-amplifier, and a speaker.
- Pedals
  - The pedals will modify the audio signal to implement effects and volume control.
- App
  - The app will have Bluetooth connectivity which will allow the user to customize the effects to their liking. In addition, the app will also allow users to save presets for added convenience.

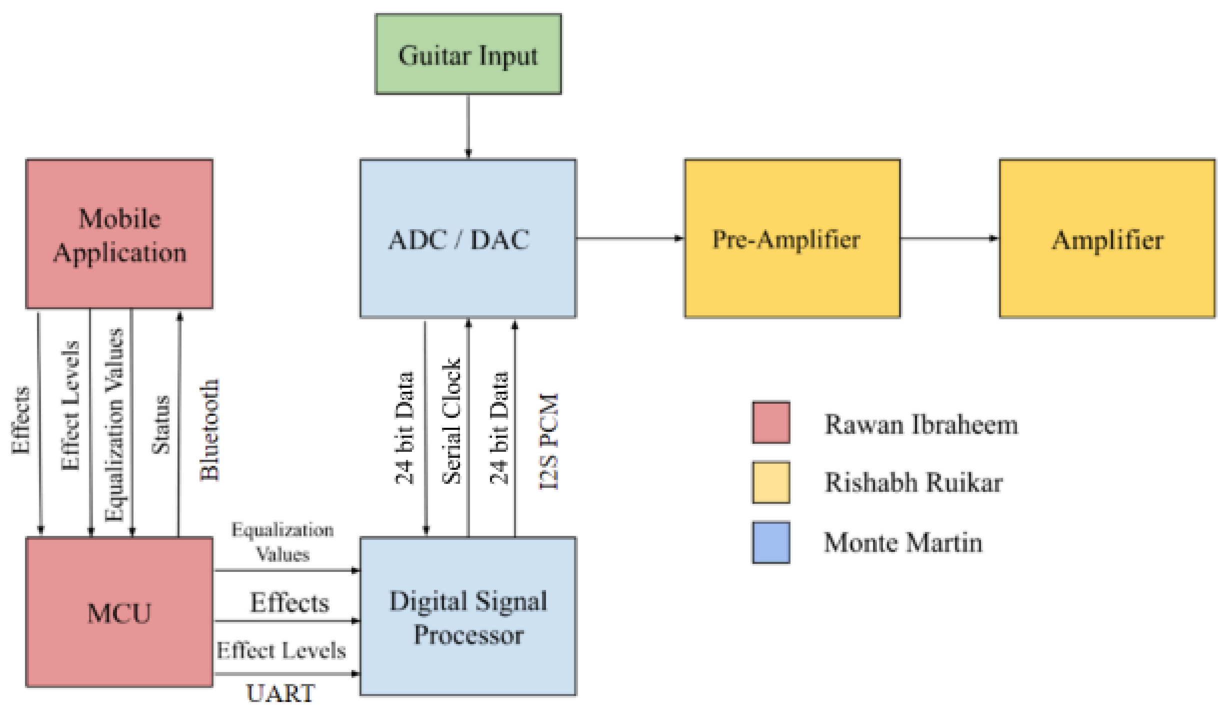


Figure 2. Block Diagram of System

This block diagram illustrates the relationship between our three subsystems. As seen the first "stage" will be the guitar signal, which will travel to the pedals (it is to be noted that the user will set the effects to be used via the app prior to using the guitar). The MCU relays the status of the DSP to the mobile application, and the app sends what effects and levels for the DSP to use. The input from the guitar goes to the ADC to convert into a digital signal so that it can be read by the DSP. Then the DSP can take the digital signal and run functions to implement the effects. Then the DAC converts the digital signal back into an analog signal which is sent to the pre-amplifier. The pre-amplifier amplifies the voltage of the signal to line level so that the amplifier can amplify the voltage and current enough to drive the speakers.

3.2. Characteristics

3.2.1.1. Frequency of Measurements

The system will be taking continuous measurements.

Rationale: The entertainment system should have a continuous audio output when in use.

3.2.1.2. Accuracy of Measurements

Less than 5 dBs of noise between the pedal analog input and output. The effects will produce waves within 5 dB of calculated values. It will also produce the output signal within 10 ns of the input signal, and can store up to 2 seconds of signal.

Rationale: The pedal system will modify the signal without reducing its quality.

3.2.1.3. Lifespan and Maintenance

The Guitar Entertainment System will be expected to last at least 15 to 20 years. If there is any loss in sound quality, the capacitors might need to be replaced. Check capacitors every 5 years, don't open the housing unless for maintenance.

Rationale: The system doesn't have anything that needs to be constantly monitored or checked on, so it should last for a long period of time.

3.2.1.4. Communications Range and Connection Time

The communication range for the system will be a minimum of 10 meters. The mobile device shall be able to connect to the system by Bluetooth within 5 seconds.

Rationale: The amplifier shall be able to be controlled by the consumer through an Android application, which may be operated from a distance.

3.2.1.5. Communications Range and Connection Time

The mobile application will communicate with the MCU to obtain status updates and send control signals when the user selects a certain guitar effect. The application is designed to run on any Android mobile device. The mobile application shall be tested on an emulator before it is tested on a mobile device.

Rationale: These requirements focus on the functionality and testing of the mobile application, ensuring it is versatile, compatible with various Android devices, and undergoes rigorous testing before consumer use.

3.2.2. Physical Characteristics

Listed below are the physical characteristics of the Guitar Entertainment System.

3.2.2.1. Mass

The mass of the Guitar Entertainment System shall be less than or equal 13 to kilograms.

Rationale: This is a requirement in order for the system to be easily portable.

.

3.2.2.2. Mounting

The PCB boards will be mounted inside of a speaker box using screws, which will have holes for plug-ins and power cables.

.

3.2.3. Electrical Characteristics

3.2.3.1. Inputs

The primary input will be a voltage signal from an electric guitar, through a 6.3 mm mono cable connection. This will be the same input from the pedal to the amplifier, although rather than using a 6.3 mm mono jack connection from the pedal to the preamplifier, we will just be using a wire soldered to the respective pedal output and then preamplifier input. The cable receptacles will be securely connected to the frame of the boxes to prevent damage upon use. The system should allow for various inputs, and the user shall be able to plug in an active and passive pickup guitar.

3.2.3.1.1 Power Consumption

The power consumption specifications for the various components of the Guitar Entertainment System are as follows. The ESP32 Microcontroller Unit (MCU) consumes approximately 792 milliwatts (mW) of power. The Analog-to-Digital Converter (ADC) requires 330 watts, while the STM32 MCU Digital Signal Processor (DSP) draws 43 watts. The Digital-to-Analog Converter (DAC) has a power consumption of 32 milliwatts (mW), and finally, the amplifier component utilizes 60 watts of power. The TL072 recommended supply voltage is 15V and the voltage that the amplifier board will be running at will be around ~24V. Both boards will be powered using a DC power supply. These power consumption figures are essential considerations to ensure efficient and stable operation of the system, aligning with the overall design and performance requirements.

3.2.3.1.2 Input Voltage Level

The input voltage level for the Guitar Entertainment System shall be +15 VDC. The voltage coming from the amp to drive the speaker to 25W is approximately 14.14 volts RMS. The input voltage level for the ESP 32 shall be +3.3 VDC and shall not exceed +3.6 VDC. Input for the STM32 will be +3.3 VDC and wont exceed +3.6 VDC. Input for the ADC will be +5.2 VDC and +3.3 VDC for the DAC. Input level for the preamplifier will be +15 VDC and for the amplifier will be +24 VDC.

3.2.3.2. Outputs

3.2.3.2.1 Sound Output

The Guitar Entertainment System sound output shall not exceed 95 dB.

3.2.3.2.2 Diagnostic Output

The diagnostic output sent from the MCU to the app will be displayed to the user on the Android application. Further details can be found in the Interface Control Document (ICD).

3.2.3.3. Connectors

The Guitar Entertainment System shall use a ¼ inch tip sleeve to transfer signals between the guitar and the pedals, and then a wire to transfer signals from the pedal to the preamplifier.

3.2.4. Environmental Requirements

The Guitar Entertainment System shall be designed to withstand and operate in the environments and laboratory tests specified in the following section.

3.2.4.1. Thermal

The Guitar Entertainment System may be operated in the temperature range 0°C to 35°C but shall not exceed 43°C.

3.2.4.2. Humidity

The Guitar Entertainment System shall not be operated in an environment with the humidity exceeding 95%.

4. Support Requirements

The application boasts an intuitive user interface, eliminating the need for supplementary documentation. Users will seamlessly tailor effects to align with their individual preferences.

Appendix A: Acronyms and Abbreviations.

ADC	Analog-to-Digital
DAC	Digital-to-Analog
Hz	Hertz
ICD	Interface Control Document
IDE	Integrated Development Environment
MCU	Microcontroller Unit
mW	Milliwatt
PCB	Printed Circuit Board
RMS	Root Mean Square
DSP	Digital signal processor(ing)
PCM	Pulse code modulation
BCK	Bit clock
LRCK	Left Right select clock or word clock
I2S	Inter-IC sound
PLL	Phase locked loop
JTAG	Joint test action group

Guitar Entertainment System

Rawan Ibraheem, Monte Martin, Rishabh Ruikar

INTERFACE CONTROL DOCUMENT

REVISION – 1
30 April 2024

Interface Control Document
for
Guitar Entertainment System

Prepared by:

Team 28 4/30/2024
Author Date

Approved by:

Monte Martin III 4/30/2024
Project Leader Date

John Lusher II, P.E. Date

T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
-	9/25/23	Rishabh Ruikar		Draft Release
2	12/2/23	Rishabh Ruikar		Final 403 Release
3	4/30/2024	Rishabh Ruikar		Final 404 Release

Table of Contents

List of Tables

All relevant tables are appropriately cataloged in the subsequent sections of this document.

1. Overview

This Interface Control Document (ICD) will provide a comprehensive overview of the interfaces governing our guitar sound system. The project consists of three critical subsystems: an amplifier, effect pedals, and a Bluetooth-controlled application. Each of these subsystems possesses distinct interfaces which interact in harmony to allow users to manipulate and customize their guitar sound with precision and ease.

2. References and Definitions

2.1. *References*

The following documents, of the exact issue and revision shown, form a part of this specification to the extent specified herein:

- IEEE 802.11 Standards for Wireless and Bluetooth Communication: https://www.ieee802.org/11/
- IEEE Analog Need-To-Knows: https://technav.ieee.org/topic/analog-circuits
- TI Digital Guitar Effects Pedal: https://www.ti.com/lit/ml/sprp499/sprp499.pdf
- ESP32 Series Datasheet: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

2.2. *Definitions*

API	Application Programming Interface
BLE	Bluetooth Low Energy
°C	Degrees Celsius
DSP	Digital Signal Processor
ESP32	Espressif System's ESP32 microcontroller
Inch	Inches (2.54 centimeters)
GATT	Generic Attribute Profile
MAC	Media Access Control
MCU	Microcontroller Unit
PCB	Printed Circuit Board
W	Watts (power unit)
XML	Extensible Markup Language

3. Amplifier: Physical Interface

The amplifier itself will contain 3 “sub-components” within it. It will contain a speaker, an amplifier circuit, and a pre-amplifier circuit. The speaker will be a Jensen C8R 8-inch 25W speaker, and it, along with the amplifier and preamplifier circuit will be contained with a 8 in. x 5 in. printed circuit board (PCB).

3.1. Mounting Locations

The amplifier involves housing the entire setup within a specialized audio enclosure or box. This design choice is driven by our aim to centralize the system components and maintain a streamlined appearance while ensuring the functional integrity of the amplifier. During 404, we were able to successfully integrate the amplifier boards within a wooden housing, ensuring stability and reducing vibration during the demo process.

4. Pedals: Physical Interface

The pedal system consists of three subsystems: the digital to audio converter, analog to digital converter, and digital sound processor. They will all be housed within the same box, on the same printed circuit board.

4.1. Mounting Locations

The Pedal System board will be securely connected to the housing using screws and brackets. The board will be directly across from the amplifier to easily connect to it.

5. Amplifier: Thermal Interface

Our amplifier has been designed to perform optimally within the standard commercial temperature range, spanning from 0°C to 35°C. This range has been selected based on typical ambient conditions, ensuring reliable operation under most everyday environments. It's worth noting that our sound system is not tailored for extreme conditions or high-demand scenarios often found in concert venues or outdoor performances. Instead, our design focus is directed towards more controlled settings, mirroring the conditions where our system will be showcased during demonstrations. Consequently, we anticipate a stable performance without temperature-related disruptions, as we have factored in the environmental variables of our intended use-case and are not subjecting the amplifier to the higher stresses seen in more demanding professional settings.

6. Pedals: Thermal Interface

The pedals are engineered for peak performance within the standard commercial temperature range of 0°C to 35°C. This range has been determined by prevalent ambient conditions to ensure dependable functionality in most common environments. However, it's pertinent to highlight that these pedals are not designed for extreme conditions or intense demands often encountered in live concert settings or outdoor events. Our primary design objective caters to controlled environments, akin to those where our system would typically be demonstrated. As such, we expect consistent and uninterrupted performance, given that we've calibrated the pedals with the environmental variables of our target use-case in mind and have steered clear of the rigorous conditions seen in some professional settings.

7. Electrical Interface

The amplifier and pedal system will be powered via a wall outlet. In short, the power will come to the circuit from the outlet, and then go into a transformer, and that signal will go to a linear voltage regulator.

7.1. Primary Input Power

Our sound system's amplifier and pedal systems will derive their power from a standard wall outlet. To ensure the power is compatible and safe for our circuitry, we will employ a transformer as a critical component in the power input stage. The transformer will serve to step-down and possibly isolate the incoming voltage from the mains to a level suitable for our amplifier. Following the transformation process, we will implement a series of voltage and current regulations to further refine the power quality. These techniques will ensure a consistent and clean power supply, eliminating fluctuations and potential risks before the energy reaches the circuits. This multi-step approach not only ensures the safety and longevity of our components but also guarantees optimal audio performance by minimizing electrical noise and interference.

8. Communications / Device Interface Protocols

The Guitar Entertainment System utilizes Bluetooth Low Energy (BLE) as its primary communication protocol, allowing wireless control between the Android application and the system components. The system defines error codes and messages to facilitate communication between the mobile app and the MCU, ensuring effective error handling and user feedback during operation.

8.1. Bluetooth Communication Protocols

The Guitar Entertainment System employs the Bluetooth communication protocol to enable seamless wireless connectivity and control. Specifically, it utilizes Bluetooth Low Energy (BLE), also known as Bluetooth 4.2, for efficient and low-power data exchange. BLE is chosen for its compatibility with modern mobile devices and its capability to facilitate real-time communication between the system components. To enable specific functionalities, the system implements Bluetooth profiles and services, with a focus on the Generic Attribute Profile (GATT). GATT is used for data exchange, allowing users to interact with the amplifier and pedal settings via the dedicated mobile application. This standardized Bluetooth protocol ensures reliable and user-friendly wireless control, enhancing the overall musical experience.

The Android application will be connected to the amplifier via Bluetooth using an ESP32. The ESP32 will be configured as a Bluetooth peripheral using the BLEPeripheral library for Bluetooth Low Energy (BLE). The Android app developed using Android Studio will have a user interface for controlling and interacting with the ESP32 device, using elements like buttons and sliders for sending commands and data. Bluetooth functionality will be integrated into the Android app by leveraging the Android Bluetooth API and ensuring proper permissions in the AndroidManifest.xml file. Pairing the mobile device and ESP32 will require the ESP32's Bluetooth MAC address for a secure connection. Bluetooth sockets, such as BluetoothSocket, will be used to exchange data between the devices.

To ensure seamless operation, logic will be implemented in both the Android app and ESP32 to handle responses to commands or data sent. This shall involve updating the app's user interface based on ESP32 responses and executing specific actions on the ESP32.

8.2. User Control Interface

Our user control interface will consist of an app which allows the user to customize the sound system to their liking. Effects that the user will be able to customize include distortion, reverb, chorus, and echo. Currently, these are subject to change, and will be finalized as the semester progresses.

Key Features:

Sound Effects Customization: Users can tailor specific sound effects to their preference. At the present stage, effects available for customization include distortion, reverb, chorus, and echo. The application interface will offer adjustable parameters for each effect, enabling users to fine-tune their sound to a precise degree.

Bluetooth Connectivity: The app will connect to the sound system via Bluetooth, ensuring a wireless, hassle-free connection. This enables real-time adjustments and feedback, allowing users to make changes on-the-fly and immediately hear the results.

User Profiles: (If applicable) Users can save their preferred settings as profiles, allowing for quick access to their favorite sound configurations in future sessions.

8.3. Real Time Control

The MCU continuously monitors user inputs, both from the physical controls on the hardware (if available) and from the mobile app. It processes these inputs and communicates with the DSP in real-time to provide immediate feedback and control over signal processing parameters.

8.4. Synchronization and Timing

The DSP shall provide feedback and status reports to the MCU, indicating the current processing state, applied effects, and any error conditions. The MCU shall use this information to update the user interface on the mobile app or to trigger specific actions based on system status.

8.5. Error Handling Protocol

A list of error codes will be defined with corresponding error messages for communication between the mobile application and MCU. The error codes shall include data validation error, communication timeout, bluetooth connection lost, etc. When the Android app receives an error code from the ESP32, it shall display a message to the user on the application. When the ESP32 receives a command, if the command fails or is not validated, the ESP32 should respond with an error code. Error handling scenarios such as Bluetooth connection lost and invalid commands shall be tested during the validation process.

ECEN 404 Execution Plan

Task	Owner	1/22	1/29	2/5	2/12	2/19	2/26	3/4	3/11	3/18	3/25	4/1	4/8	4/15		
Send out new pre-amplifier and amplifier PCBs for manufacturing.	Rishabh															In Progress
Solder parts onto amplifier PCBs	Rishabh															Not Started
Utilize a signal generator to produce sine waves, square waves, and pink noise for testing the amplifier's signal integrity and frequency response, aiming for a flat response within ±0.5 dB across 20 Hz to 20 kHz.	Rishabh															Completed
Conduct dynamic range measurements, aiming for a dynamic range of over 75 dB to capture the full spectrum of audio without distortion or noise intrusion.	Rishabh															Behind schedule
Execute thermal management tests, ensuring the amplifier maintains operational temperatures below 50°C under full load conditions to guarantee long-term stability.	Rishabh															
Test pedal PCB and redesign.	Monte															
Send out pedal PCB for manufacturing.	Monte															
Solder parts onto pedal PCB	Monte															
Test and Validate PCB	Monte															
Ensure that there isnt any signal loss when transfering to amplifier	Monte															
Test the Pedal PCB with the application and ensure there is sound output	Monte															
Ensure that ESP32 can send signals to STM32 to adjust sound effects	Rawan															
Code each sound effect in STM32 using development board and function generator	Rawan															
Program each sound effect onto STM32 on Pedal PCB																
Begin designing housing	All															

House all parts	All															

Subsystem	**Paragraph**	**Deliverable**	**Methodology**	**Owner**	**Completion**	**Completion Date**
MCU & Application	3.2.1.5	ESP32 can communicate with STM32	Terminal COM port connected to STM32 displays the applied sound effects	Rawan	Completed	2/15
MCU & Application	3.2.1.5	STM32 can apply each sound effect (volume, distortion, reverb, chorus, delay, wah-wah)	Use a function generator and an osciliscope to test input and output signal when sound effect applied and process input and output data through USB connection to computer	Rawan	Incomplete	
MCU & Application	3.2.1.4	Application reports errors connecting to ESP32 or STM32	Implement pop-up window that displays warnings in Android application	Rawan	Incomplete	
MCU & Application	3.2.1.4	MCU can send signal to STM32 within 1 second	Use oscilloscope to measure the input and output signal	Rawan	Completed	4/10
ADC/DAC, DSP	3.2.1.2	Practical Filter eliminates noise and creates a differential signal	Use oscilloscope to provide an input signal and the output signal	Monte	Completed	1/15
ADC/DAC, DSP	3.2.1.2	STM32 communicates with the ESP32 with less then 10 ns of delay	Check data readouts from STM32 and ESP32	Monte	Complete	4/7
ADC/DAC, DSP	3.2.1.2	All effects work as intended, with outputs within 5 dB of calculated values	Use oscilloscope to provide an input signal and the output signal	Monte/ Rawan	Incomplete	
ADC/DAC, DSP	3.2.1.2	have less than 10 ns of delay between signal input and output	Use an oscilloscope to measure delay between input and output signals	Monte	Complete	4/7
ADC/DAC, DSP	3.2.1.2	The delay function can create up to 2 seconds of delay without loss of signal quality	Use a timer and oscilloscope to measure delay and signal quality	Monte/ Rawan	Incomplete	
ADC/DAC, DSP	3.2.1.2	ADC, DAC, and DSP all can communicate within 10 ns of delay when tested seperately	Osciliscope to send and recieve analog, ESP32 dev to send and recieve digital through i2S	Monte	Complete	4/7
ADC/DAC, DSP	3.2.1.2	The system can take in up to 2.1 V rms signals and output them without any clipping or loss of signal quality	Oscilloscope and function generator	Monte	Completed	4/18
Amplifier	3.2.3.3	Preamplifier is able to create at least a 10dB gain from 3 types of sine waves- 50Hz(bass), 1kHz (mid), 5kHz (treble)	For each frequency test (50 Hz, 1 kHz, 5 kHz), apply a constant-level sine wave, measure the preamp's output, and analyze the amplitude and harmonic content to evaluate its frequency response and distortion characteristics.	Rishabh	Completed	3/3

Amplifier	3.2.3.3	Validate the frequency response curve by comparing it against the amplifier's specified performance criteria, ensuring it meets the expected flatness within ±0.5 dB across the 20 Hz to 20 kHz range, using calibrated measurement equipment for accuracy.	Using software-based audio signal generators, generate a frequency sweep from 20 Hz to 20 kHz to test audio equipment.	Rishabh	Completed	4/6
Amplifier	3.2.3.3	Aim for a flat response within ±0.5 dB across 20 Hz to 20 kHz	Using a signal generator to produce sine waves, square waves, and pink noise for testing the amplifier's signal integrity and frequency response,	Rishabh	Completed	4/6
Amplifier	3.2.3.1	Aim for a dynamic range of over 120 dB to capture the full spectrum of audio without distortion or noise intrusion.	Generate 1 kHz sine, measure output and noise, calculate dynamic range.	Rishabh	Completed	4/8
Amplifier	3.2.3.2	Ensure the amplifier maintains operational temperatures below 100°C under full load conditions to guarantee long-term stability.	Load testing, send peak sine waves constantly for about 30 minutes	Rishabh	Completed	4/9
All	N/A	User can adjust sound effect signal multiple consecutive times.	One user will repeatedly adjust sound effects in one minute as another user plays the guitar, to ensure system operates as intended.	All	Completed	4/10
All	N/A	User can play the guitar while adjusting the sound effect.	One user will control the mobile application while another plays the guitar.	All	Incomplete	
All	3.2.4	System experiences no failure when tested outdoors.	Fully integrated system is taken outside into an open and windy area, each sound effect is tested.	All	Incomplete	
All	3.2.3.1	User can plug in an active and passive pickup guitar.	Function generator to simulate active and passive pickups and oscilloscope will be used to see the readings.	All	Complete	4/7
All	N/A	System experiences no failure when tested in a place with high signal noise pollution.	System will be tested in the FEDC where other teams are working on Bluetooth-based projects.	All	Complete	4/10
All	N/A	SNR of each sound effect vs Pedal PCB output waveform is a minimum of 25 dB	Matlab will be used to compute the desired effect from the input waveform measured by the oscilloscope. Ouput of pedal PCB will be saved to a csv file and the data will be compared in Matlab.	All	Incomplete	

Guitar Entertainment System

Monte Martin III, Rishabh Ruikar, Rawan Ibraheem

SUBSYSTEM REPORT

SUBSYSTEM REPORT

FOR

Guitar Entertainment System

TEAM <28>

APPROVED BY:

Monte Martin III 4/30/2024
Project Leader Date

____
Prof. Lusher Date

____
T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
-	9/11/2023	Rishabh Ruikar		Draft Release
2	12/2/2023	Rishabh Ruikar		Final 403 Release
3	4/30/2024	Rishabh Ruikar		Final 404 Release

Table of Contents

List of Figures

1. Pedal Subsystem Report

1.1 Subsystem Introduction

The pedal subsystem works by taking in an analog signal from the guitar and sending a modified analog signal out to the amplifier subsystem. When it takes in the signal, It first goes through a practical filter which eliminates high frequency noise and any other form of interference to prevent contamination from aliased noise. It also converts a mono signal into a differential signal, which is then fed into the Analog to Digital converter (ADC). The digital signal is sent to the STM32 which is acting as the Digital Signal Processor (DSP). Then, based on the inputs received from the ESP32 which communicates with the application, it modifies the signal by running functions to implement sound effects. Once modified, it is sent to the Digital to Analog converter (DAC) to be converted back into an analog signal.

1.2 Subsystem Design

The design of the pedal subsystem is explained below. A block diagram of the pedal subsystem is shown in figure 1.

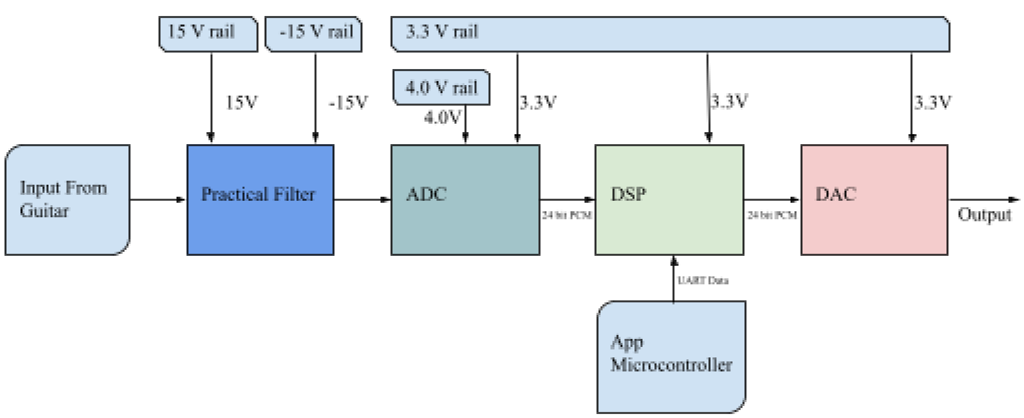


Figure 1: Pedal Subsystem Block Diagram

1.2.1 Analog to Digital Converter PCM4222

The PCM4222 was chosen as it is a high performance ADC designed for use in professional audio applications. Because of its support of 24 bit linear PCM it is ideal for digital audio processing applications. The PCM4222 has three PCM sampling modes, supporting sample rates from 8k to 216kHz. In order to function the ADC needs a supply of 3.3V for its digital functions and 4V for its analog functions, with a maximum power dissipation of 300mW. The chips sampling mode, PCM audio data formats and word length, decimation filter response, and high pass filter are all controlled through dedicated control pins. The ADC should be able to take in an analog waveform and produce a corresponding PCM signal. The Schematic of the PCM4222 layout is shown in figures 2 and 3.

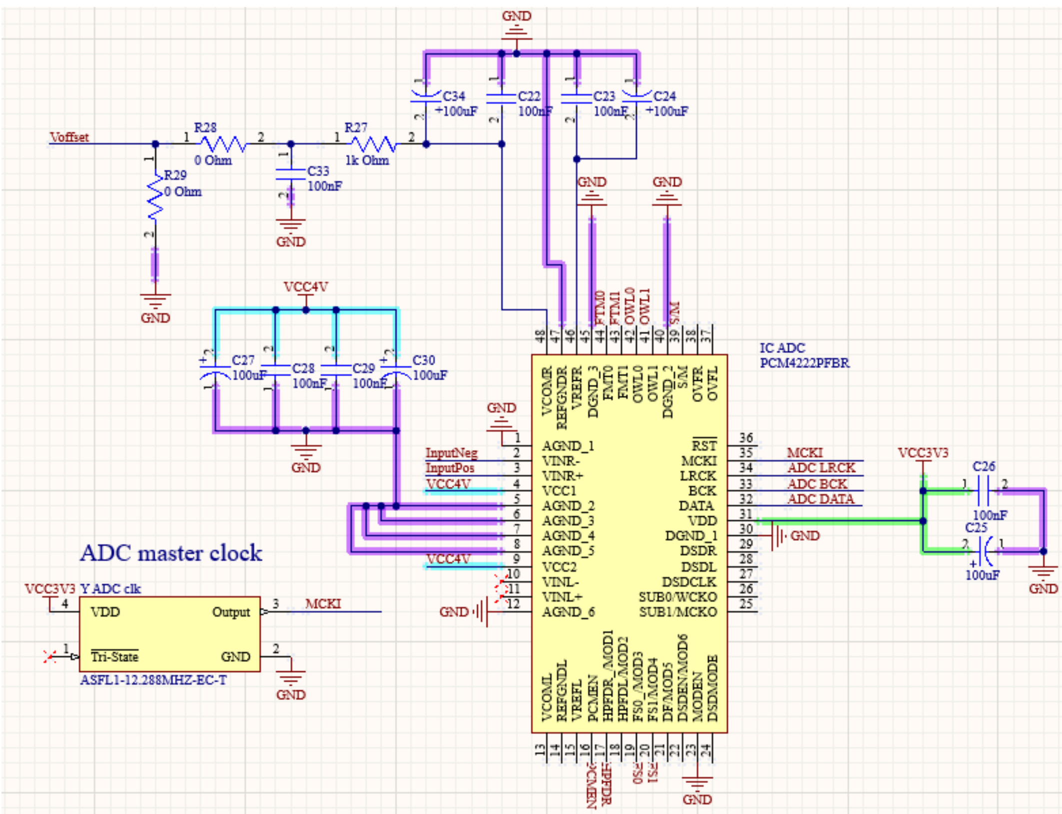


Figure 2: Detailed PCM4222 Schematic

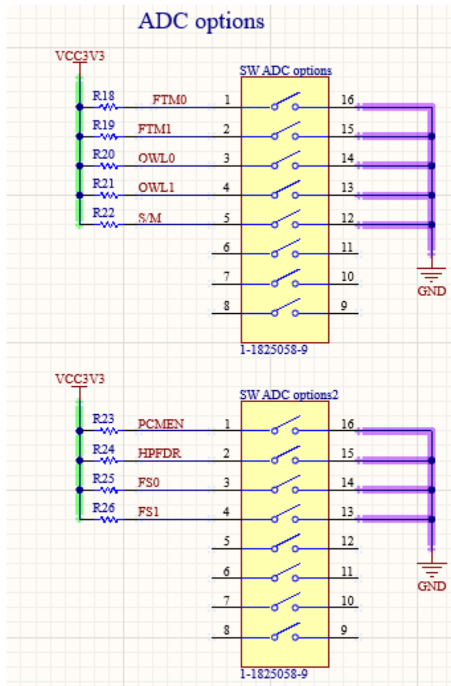


Figure 3: PCM4222 Schematic Control Pins

1.2.2 Practical Filter

The practical filter in this application converts the single ended signal into a differential signal to help filter out noise, and to also perform antialiasing. There is also an adjustable gain knob for inputs that are substantially higher than a typical guitar input. The operational amplifier chosen for the filter is the OPA1622 as it is designed for high fidelity audio applications. The schematic of the one used is in figure 4.

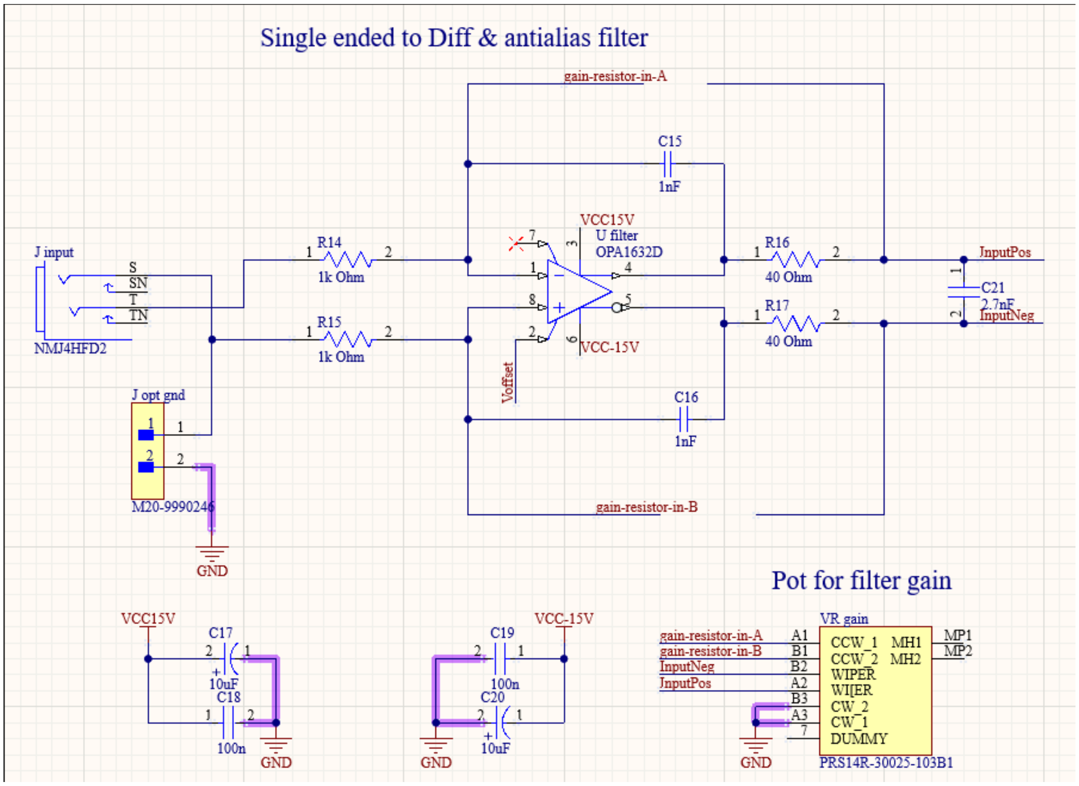


Figure 4: Practical Filter

1.2.3 Digital to Analog Converter PCM5142

The DAC in our project takes in PCM data from the DSP and outputs an analog audio signal. The PCM5142 is ideal for this application as it accepts a wide range of PCM data formats and lengths, while having a 2.1Vrms ground centered output. The ground centered output eliminates the need for DC blocking capacitors and external muting circuits. It also has an integrated PLL which eliminates the need for a system clock, allowing for a 3-wire I2S input connection, and for power the PCM5142 only requires a 3.3V supply. The DAC should be able to take in an I2S PCM input signal and output the corresponding analog audio signal. The onboard filters, as well as the PCM input format and digital gain can be adjusted using control pins. The schematic used can be seen in figure 5.

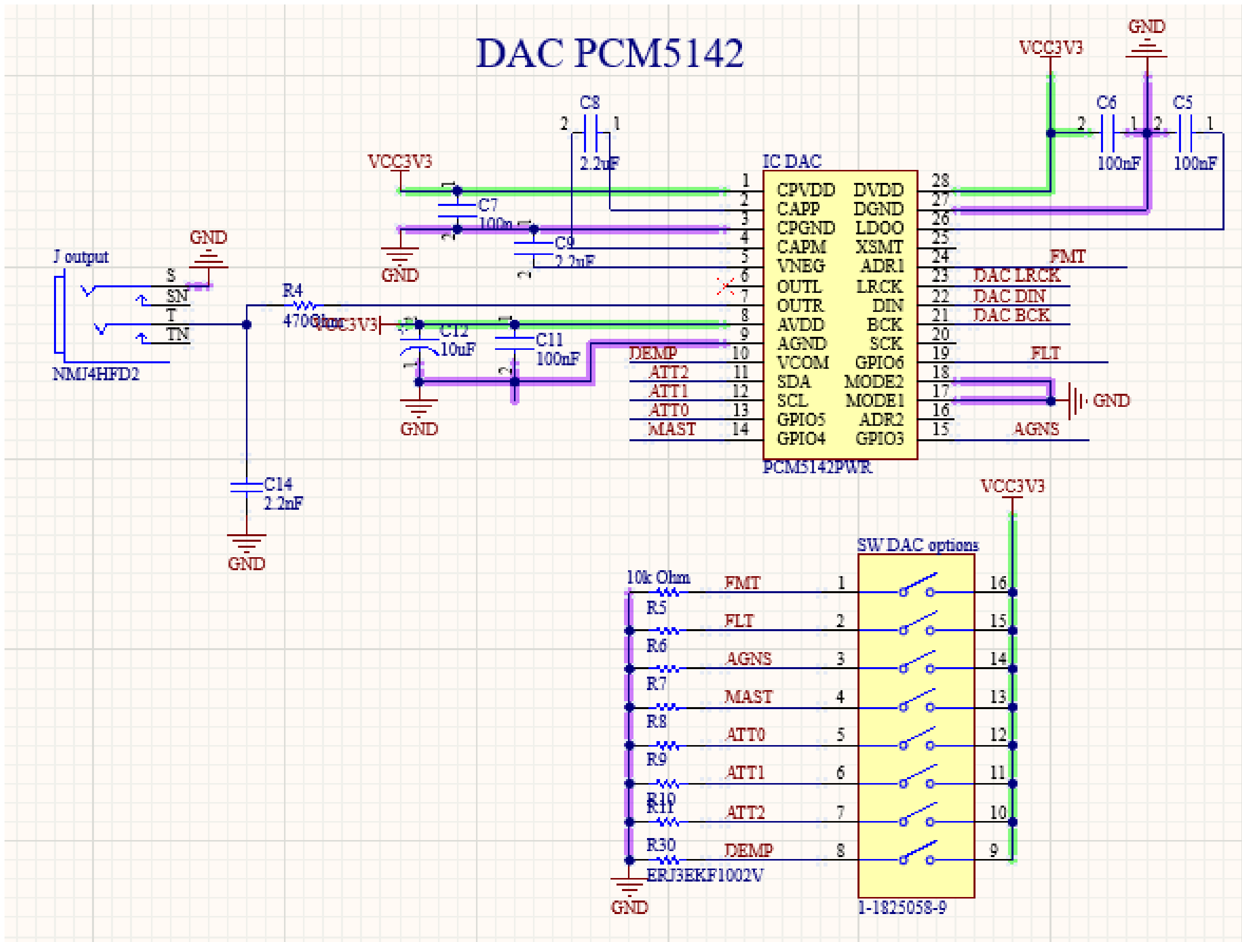


Figure 5: PCM5142 schematic and control pins.

1.2.4 Digital Signal Processor STM32F446

The DSP for the project must be able to handle at a minimum 16 bit 44.1 kHz sample rate PCM audio, and it must have two I2S peripherals. The STM32F446 was chosen as the DSP as it has two dedicated audio PLLs, three half duplex I2S, and two serial audio interfaces supporting full duplex I2S. The 32 bit 180 MHz processor also has DSP instructions and has a large amount of available resources that can be used as reference for programming. For power the chip requires a 3.3 V supply with a maximum power dissipation of 606mW.

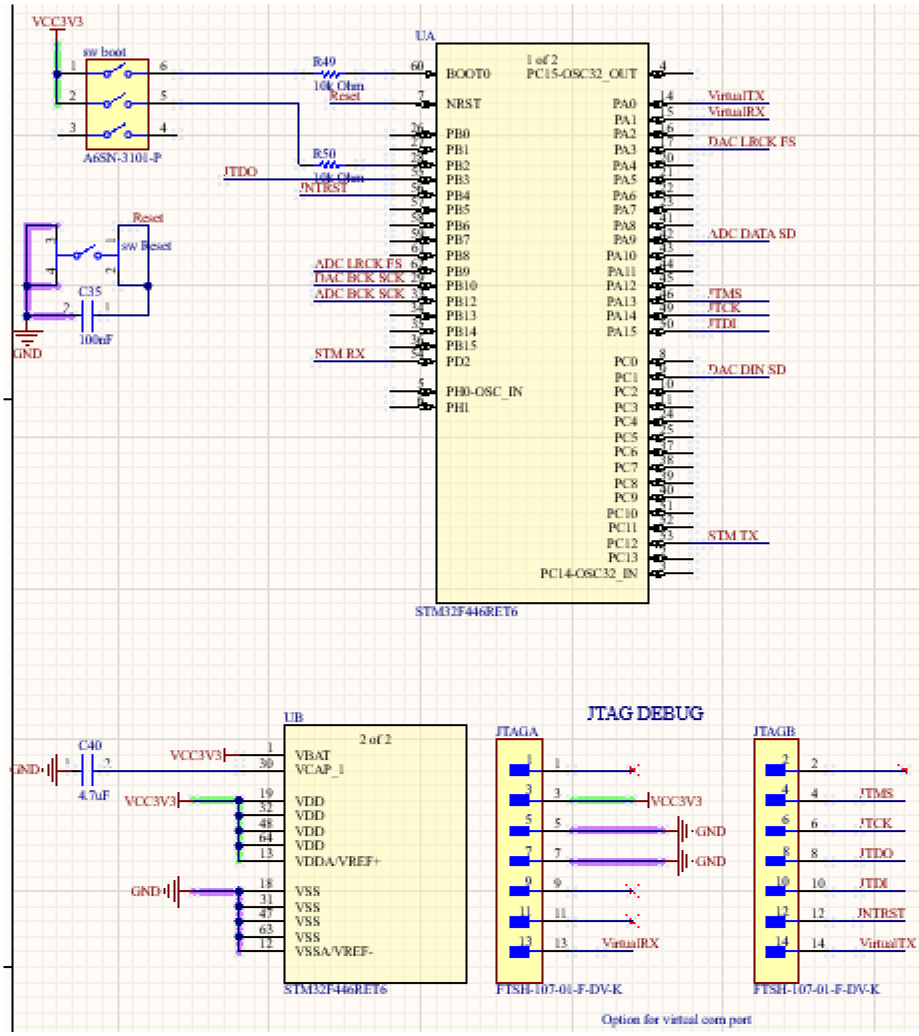


Figure 6: STM32F446 schematic

1.2.5 App Microcontroller ESP32-S3 WROOM-1U

The microcontroller for the application will be on the pedal system board, and will need to have bluetooth low energy capabilities. For this, the ESP32-S3 WROOM-1U was chosen because of its 16 MB flash size, 240 MHz clock speed, and external antenna. The chip requires a 3.3 V supply and a maximum power dissipation of 1.65 watts. The schematic used can be seen in figure 7.

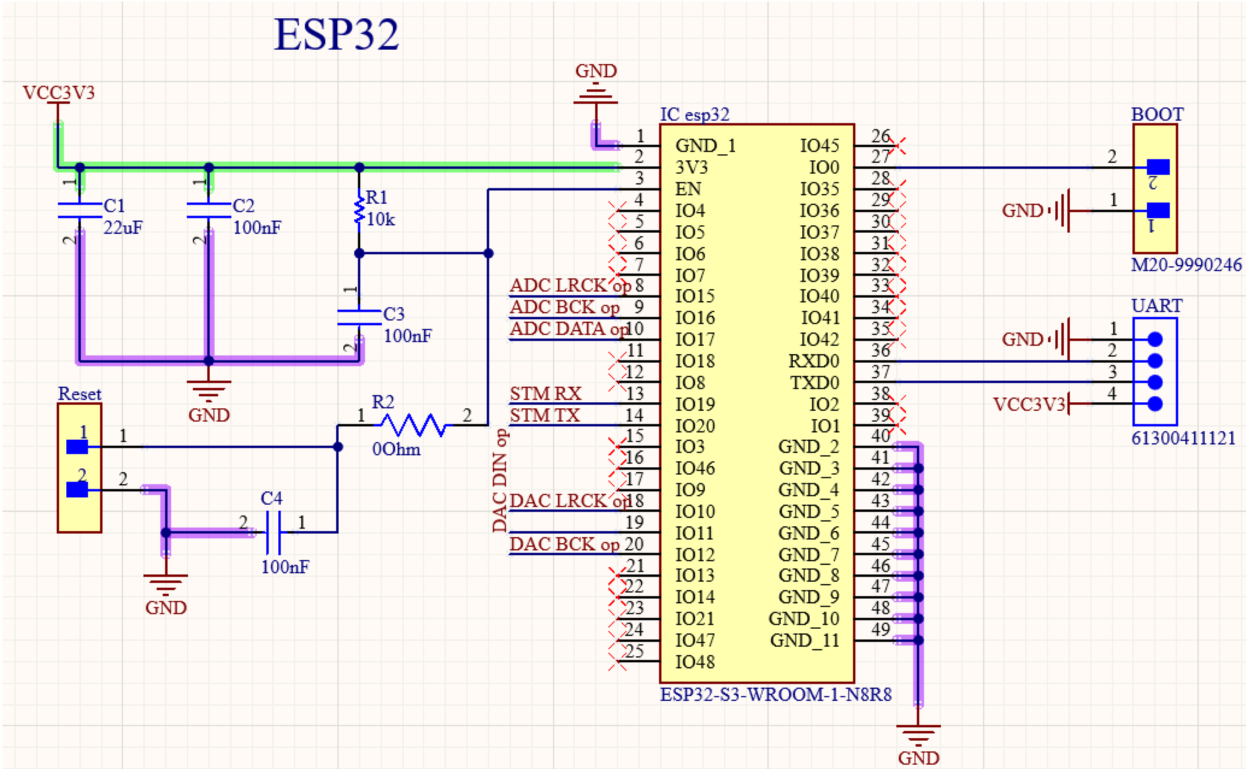


Figure 7: ESP32-S3 WROOM-1U schematic

1.3 Operation

The PCB for the pedal system is a four layer board with a ground plane and a 3.3V power plane.

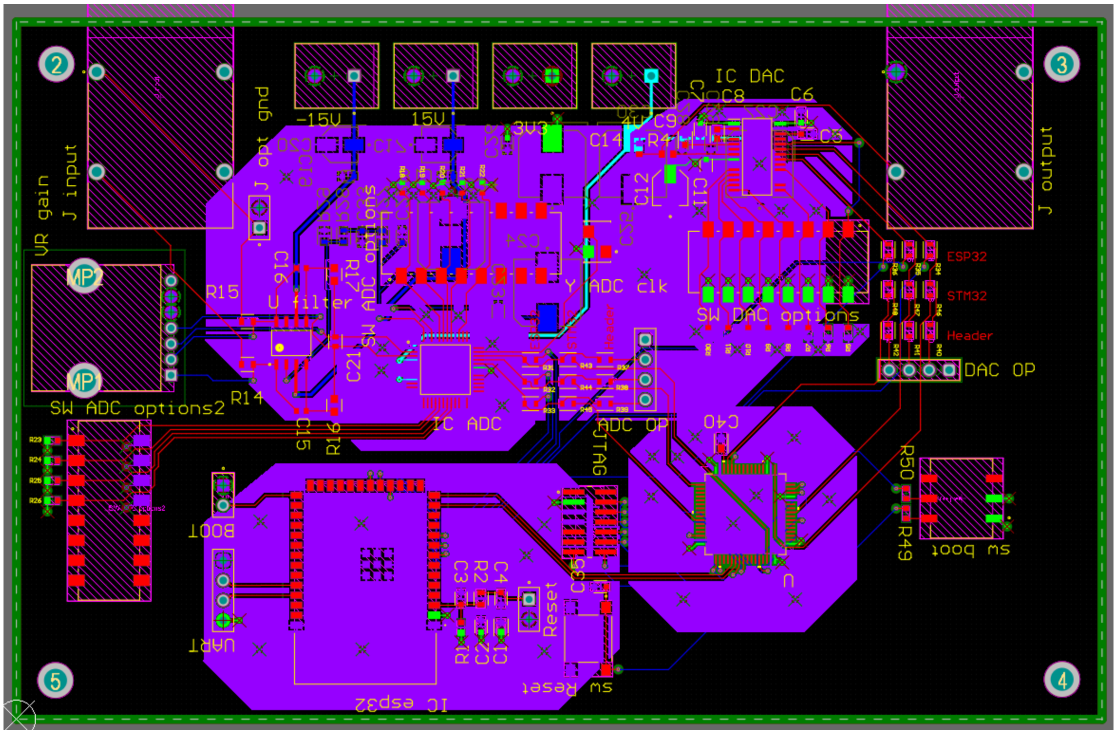


Figure 8: Pedal system PCB layout

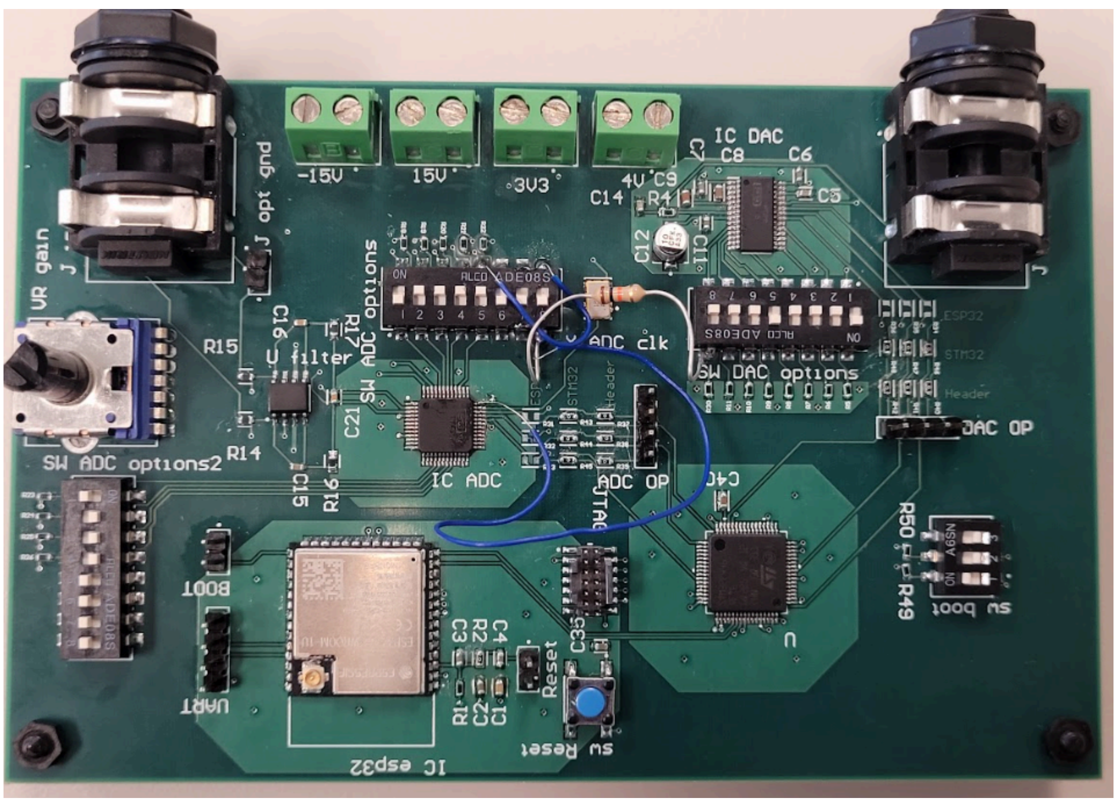


Figure 9: Pedal System PCB

After ordering the PCB it was found that the reset pin for the ADC needs to be held high for normal operation, so a pull up resistor was added. The ADC and DAC were both able to function and could be tested by connecting the I2S output of the ADC to the input of the DAC.

1.3.1 Analog to Digital Converter PCM4222

Once the pull up resistor was added to the ADC, it began producing a PCM signal for the corresponding analog input. The BCK and LRCK can be seen in figures 10 and 11, which were fed into the ADC by the DSP. Then the ADC can turn the analog input into a digital signal, as seen in figure 12.

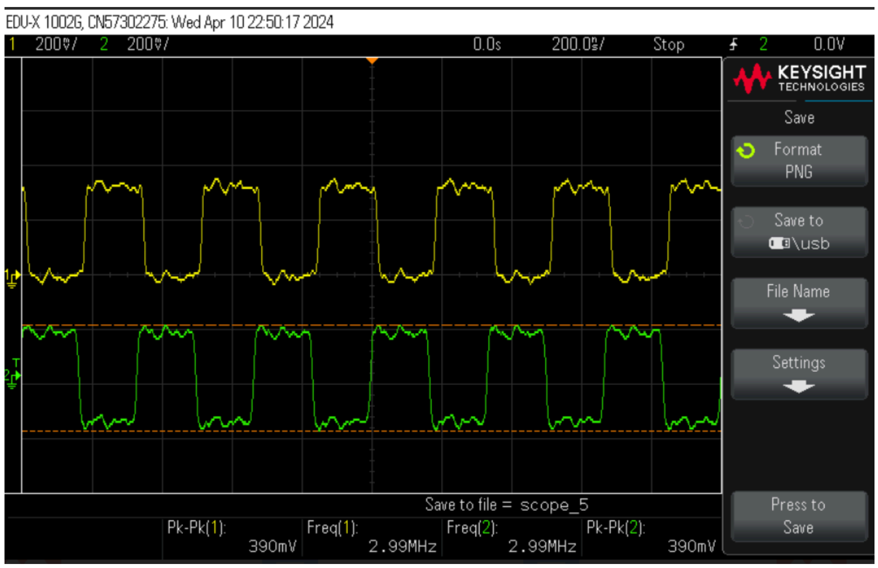


Figure 10: BCK for ADC and DAC at frequency 3MHz

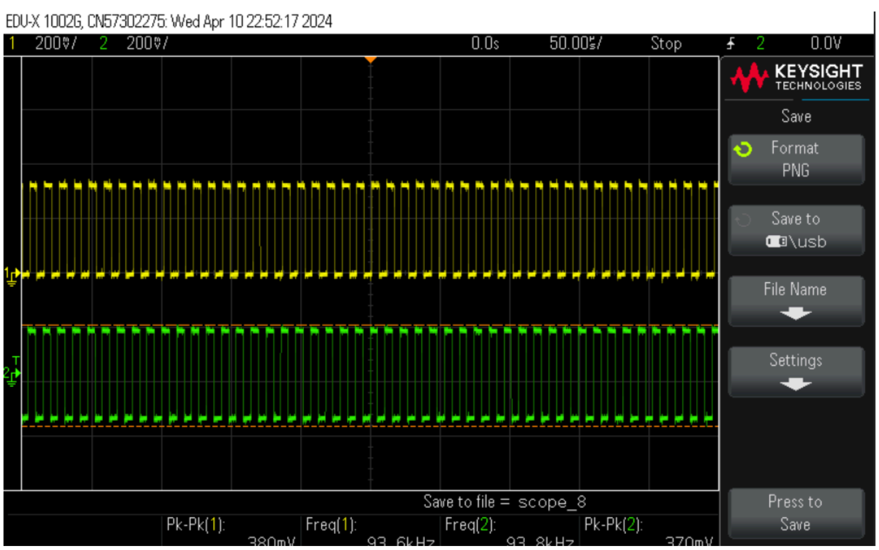


Figure 11: LRCK for the ADC and DAC Frequency 93 kHz

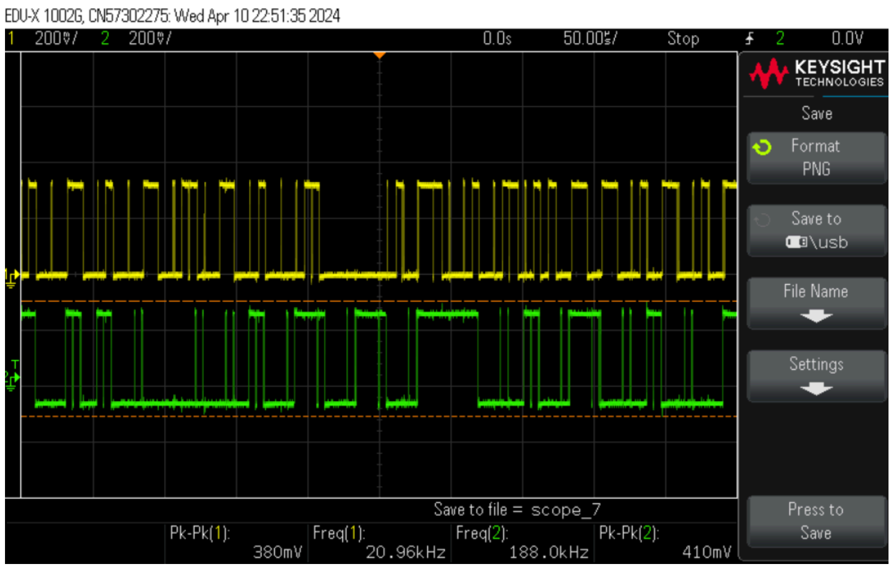


Figure 12: Data lines for ADC and from DSP

1.3.2 Digital to Analog Converter PCM5142

The DAC was able to be tested by using the ADC as an input assuming that the ADC is working. To test this the system was fed a 400mV 1kHz sine wave, and observing the input and output as seen in figure 13.

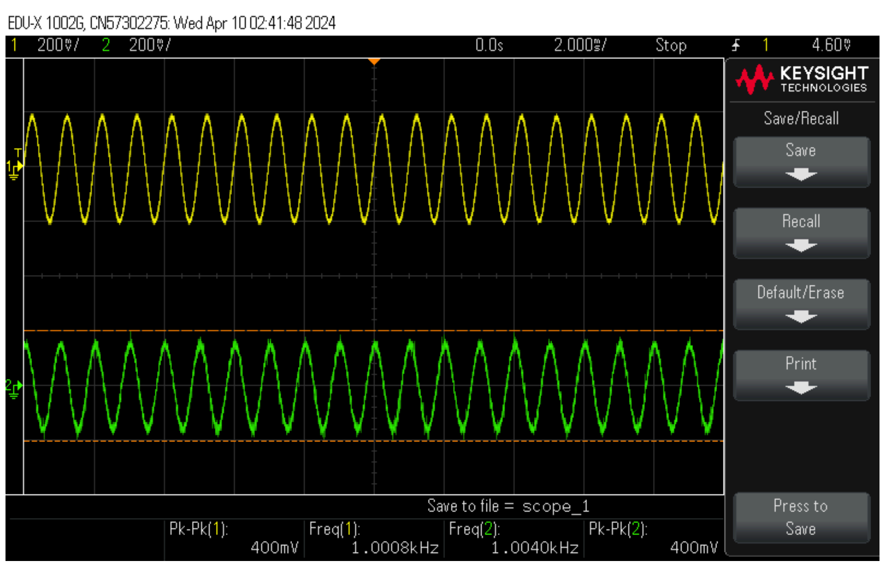


Figure 13: Results from feeding the ADC output into the DAC input

1.3.3 Digital Signal Processor STM32F446

The STM32F446 was able to power on and be successfully programmed. To program the DSP, an ST link was connected to the board through JTAG, and then connecting the

STM32 with the STM32cubeIDE. The functionality can be seen when it was used to test the ADC and DAC by feeding them BCK and LRCK.

1.4 Conclusions

The pedal subsystem is a vital part of the guitar entertainment system, as without it there wouldn’t be a sound output. The major accomplishments were a working ADC and DAC, able to handle 24 bit data at up to 216kHz sample rate, a programmable DSP in the form of a STM32F446, and a programmable bluetooth LE enabled ESP32-S3 WROOM-1U. The subsystem was completed and was able to be fully integrated with the bluetooth application and the amplifier subsystems.

1.4.1 Learnings

There were multiple learning outcomes as a result of completing the project. The most important one was learning how to work and communicate with a team. The project took almost nine months to complete, and there were many opportunities to build teamwork capabilities and trust. Another important lesson was to recognize when you can’t do something or when you have too much to do. It is important to not try and do everything yourself, especially in a group setting where the workload is meant to be shared. This prevents burnout and frustration, and helps improve productivity and performance.

2. Android Application Subsystem Report

2.1 Subsystem Introduction

Figure 14. Guitar Sound Effects Logo

The Android application titled “Guitar Sound Effects” is used to send control signals through a Bluetooth Low Energy (BLE) connection to the microcontroller, the ESP32. If the user desires to change effects such as the volume, chorus, reverb, distortion, delay, or wah-wah, they may update the value of these variables in the application, and then the corresponding update is sent in string format to the ESP32.

Although DSP programming is different from Android application development, it is part of a broader subsystem that encompasses programming the ESP32, the Android app, and the DSP. This subsystem requires a comprehensive understanding of audio protocols, the mathematical principles behind sound effects, and programming across different platforms including the STM32 microcontroller.

2.2 Subsystem Hardware

Figure 15: Samsung Galaxy A03s in Black, 32GB

The subsystem was validated with an android application that runs on the Samsung Galaxy A03s with a RAM of 3 GB and a storage capacity of 32 GB. The mobile runs Bluetooth 5.0 and supports BLE.

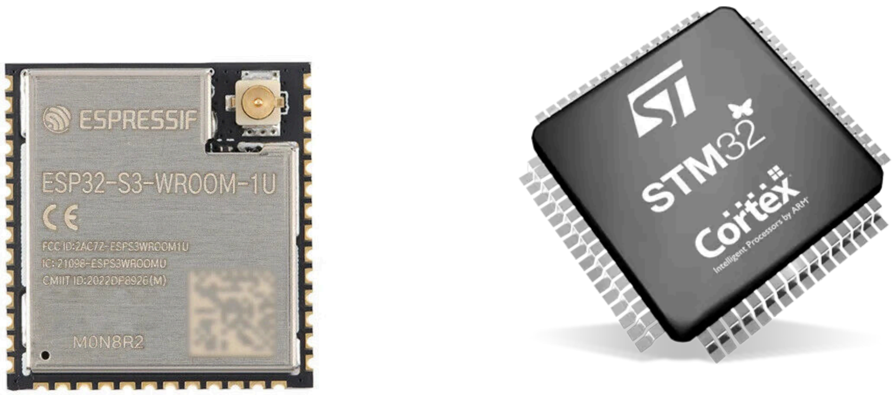


Figure 16: ESP32-S2-WROOM-1U and STM32 Microcontrollers

The microcontroller used to validate the subsystem was the ESP32-S3-WROOM-1U on the pedal PCB. The ESP32 WROOM has the Bluetooth LE specification. The MCU supports data transmission at a rate of 150 Mbps. The radio transmission is through a Near Zero Intermediate Frequency (NZIF) receiver with a sensitivity of –97 dBm. This allows for data to transmit to a distance of over 10 meters.

The application was developed in Android Studio on the Surface Book 3 with the Intel(R) Core(TM) i7-1065G7 processor with a CPU of 1.30 GHz. The device has 32 GB of RAM and has a 64-bit operating system.

The microprocessor or digital signal processor used in the guitar entertainment system is the STM32F446RE. The MCU features an ARM 32-bit CPU capable of running up to 180 MHz and is equipped with 512 KB Flash and 128 KB SRAM for extensive memory resources. It supports an array of peripherals including USB OTG, multiple USARTs, UARTs, I2C, SPI ports, and advanced analog features with multiple ADCs and DAC. The communication protocols used for this project include I2S, UART, and SWD for debugging purposes.

2.3 Subsystem Software

2.3.1 Application and MCU Software

The Samsung Galaxy A03s runs on Android Version 13 and on One UI Core version 5.1. The development environment was downloaded on the Surface Book running Windows 10 Pro.

Figure 17: Android Studio Logo

The application was developed in Android Studio Giraffe, Version 2022.3.1. The application was developed in the language Java, specifically version 8. The target Software Development Kit (SDK) is 33 and the compile SDK is 34. The Kotlin plugin version is managed through the Kotlin Bill of Materials (BOM) with the specified version 1.8.0. Several AndroidX libraries are included, such as appcompat, material, constraintlayout, lifecycle-livedata, lifecycle-viewmodel, navigation, gson, kotlin, legacy-support-v4, recyclerview, and bluetooth. Special permissions required by the application include Bluetooth, Bluetooth Scan, Access Fine Location, Access Coarse Location, Bluetooth Admin, and Bluetooth Connect to ensure that the device can scan for BLE devices and establish a secure connection.

The ESP32 development board was programmed in the Visual Studio (VS) Code IDE version 1.84.1, using core version 6.1.11 of the PlatformIO extension.

2.3.1 DSP Software

The programming of MCU involved the STM32Cube IDE. The IDE allows you to configure your peripherals based on the selected chip. You are able to select the parameters of UART and I2S, such as number of bits, interrupt or DMA, and sample frequency through their graphical user interface. The I2S peripherals were configured in Master mode so that it would drive the clock to the ADC and DAC. The sample frequency was 16 kHz, and the direct memory access (DMA) stream was activated for both I2S ports (send and receive). The DMA controller was set to circular to support the incoming flow of data, and the data width was set to one byte. While a higher sample frequency was desired, the clock configuration for a higher frequency (96 kHz) led to no output due to the maximum clock frequency for Advanced Peripheral Bus 1 (APB1) being 45 MHz. The audio was transmitted as 24 bits on a 32 bits frame through the standard I2S Phillips protocol.

2.4 Application Screenshots

The application home page is the BLE scanner page. Upon opening the application, the application begins scanning for nearby BLE devices with a signal strength of minimum -100 dBm. This is in the case that the ESP32 is far away or the Guitar Entertainment system is across the room of the performer. The user can then navigate to the effects page, where they can press on the desired effect and change the level then save the adjustment. Lastly, the user can save and recall preset profiles if they repeatedly use the same combination of effect levels. The preset profile will be saved to the mobile phone and retrievable upon closing and opening the application.

Figure 18: BLE Scanner Home Page

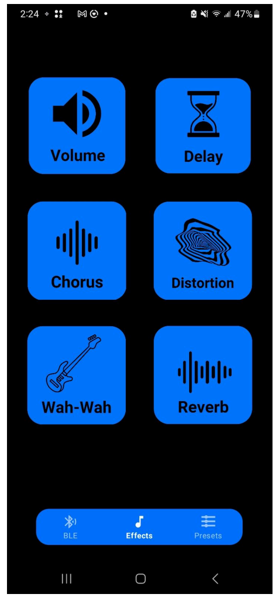


Figure 19: Sound Effect Page

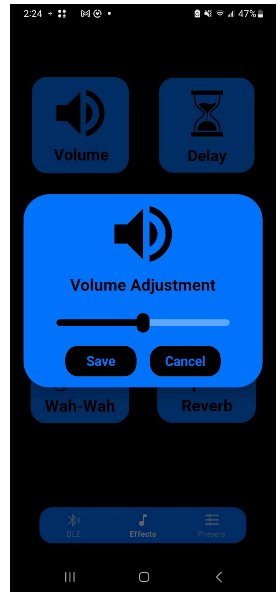


Figure 20: Volume Adjustment Dialog Pop-up

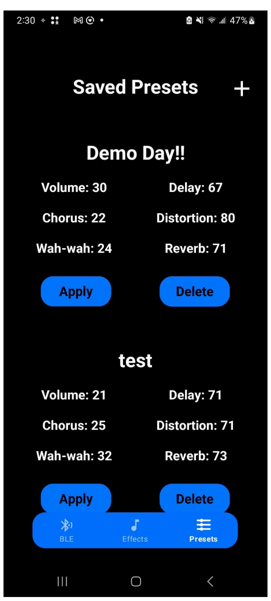


Figure 21: Presets Page with User Saved Presets

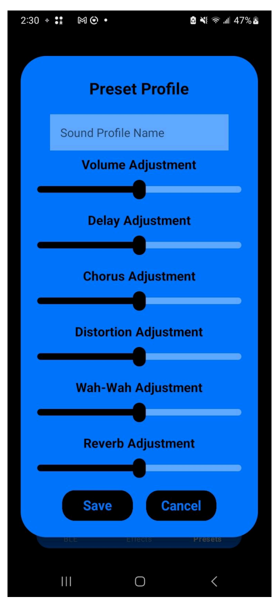


Figure 22: Preset Profile Dialog Pop-up

2.5 User Manual

Below is the user manual, which shows how to operate and navigate throughout the application, and update desired sound effects. While the app is self explanatory, the user manual is to ensure it is accessible to all users, even those who may be unfamiliar with modern technology and software.

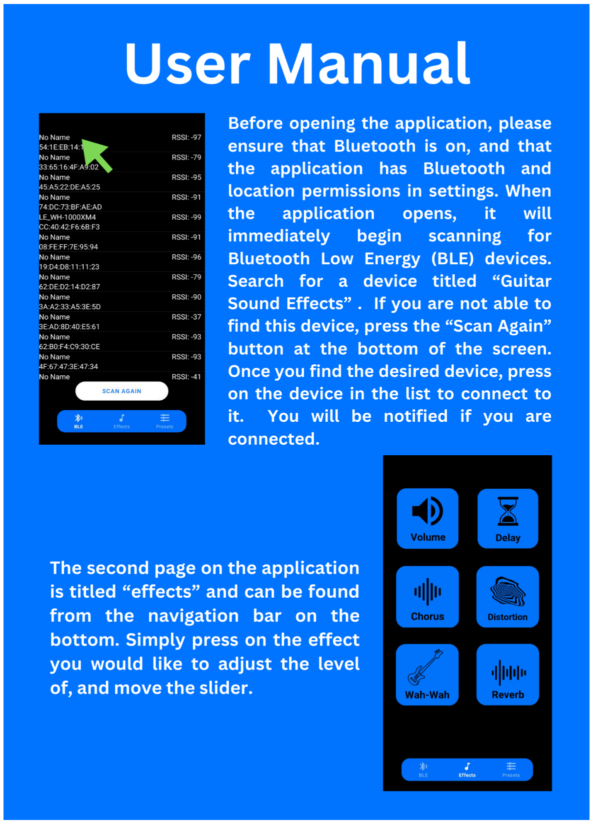


Figure 23: Page 1 of the User Manual.

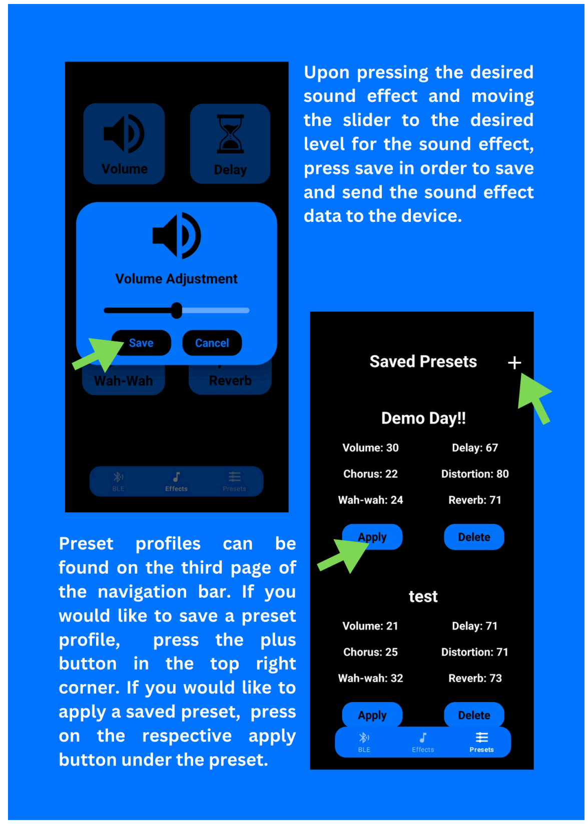


Figure 24: Page 2 of the User Manual

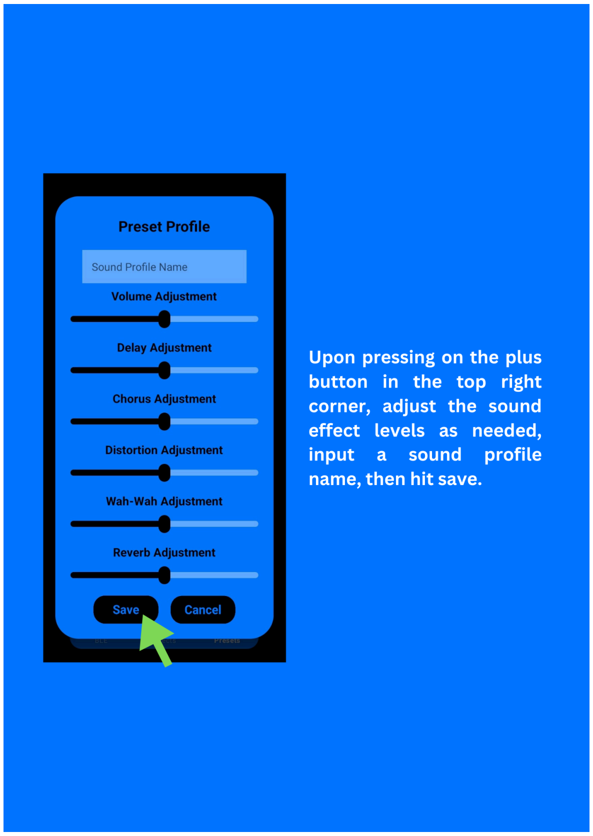


Figure 25: Page 3 of the User Manual

2.6 Subsystem Validation

Figure 26: Guitar Sound Effects Application Prototype

Above is a screenshot from the original application ran on an emulator in Android Studio, specifically the Pixel 7 with an API of 26. This was prior to incorporating Bluetooth into the application and redesigning the user interface. The subsystem was then tested and validated on the Samsung Galaxy A03s to ensure Bluetooth functionality.

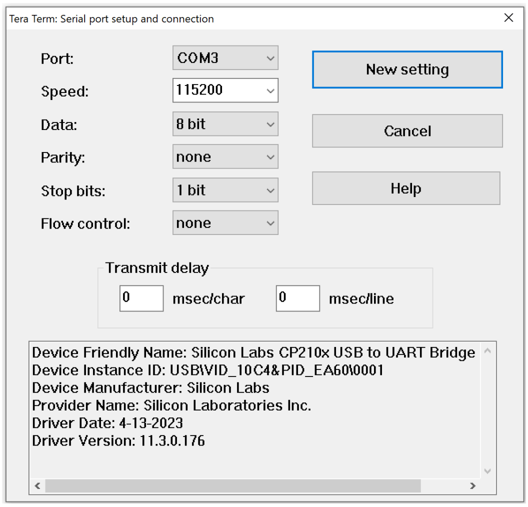
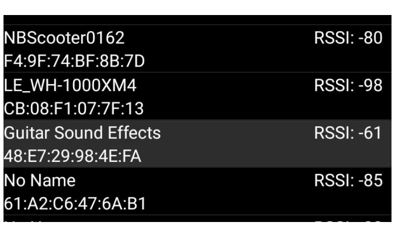
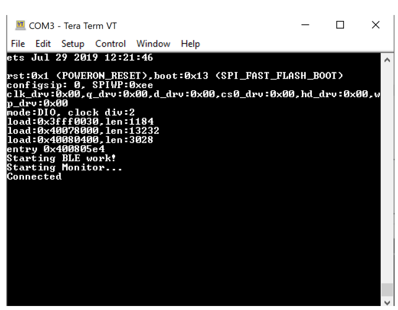
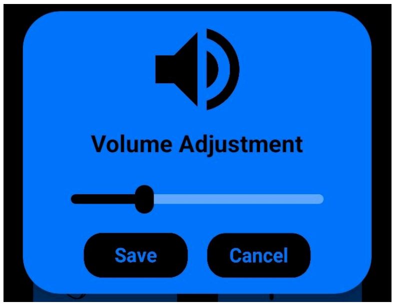
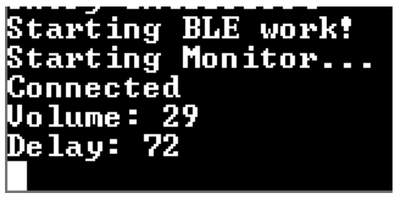
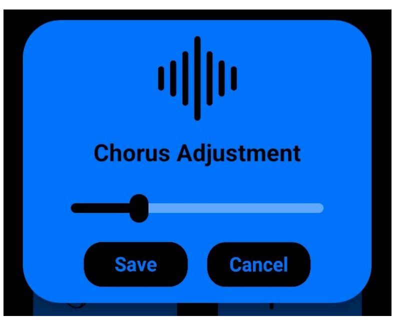
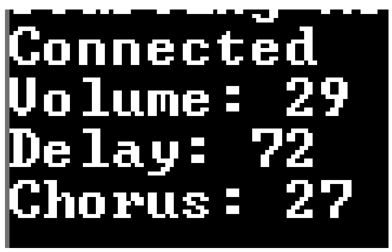


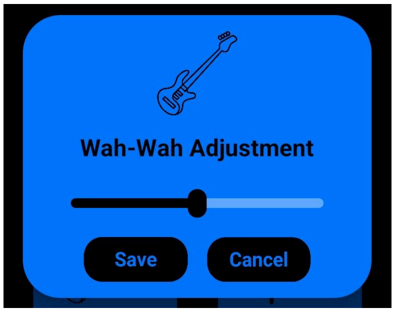
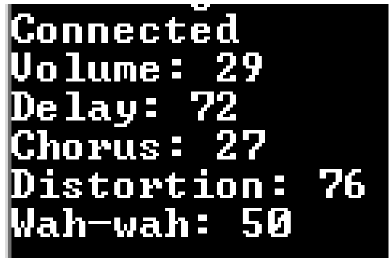
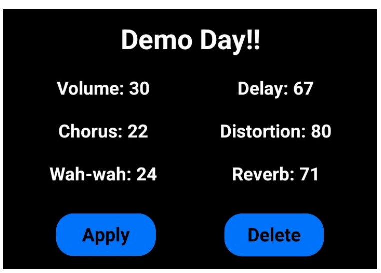
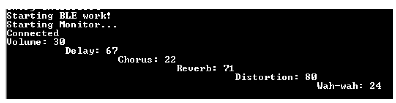
Figure 27: Serial port setup for the ESP32

Shown below is the time it takes to send a sound effect adjustment from the "Guitar Sound Effects" application on the Samsung Galaxy A03s. The microcontroller used to validate the time to receive each signal was the ESP32 WROOM on a development board purchased from Flutesan. The ESP32 signals were read through Tera Term, by connecting the development board to the Surface Book through a micro-USB cable. The setup details are shown in Figure 20.

Table 1: Signals sent from the application to the ESP32 and the time it took to receive the signal.

<table>
<tr><td>Signal Sent</td><td>Signal Received</td><td>Time To Receive Signal (seconds)</td></tr>
<tr><td>
"Connected" upon pressing on the "Guitar Sound</td><td></td><td>2.51</td></tr>
</table>

Effects" device, the discoverable name of the ESP32.		
 Volume: 29		1.06
 Delay: 72		0.71
 Chorus: 27		0.63
 Distortion: 76		0.63

 Wah-wah: 50	 	1.03
 Reverb: 66	 	1.26
 Preset: “Demo Day!!” Volume: 30 Delay: 67 Chorus: 22 Distortion: 80 Wah-wah: 24 Reverb: 71	 	1.53

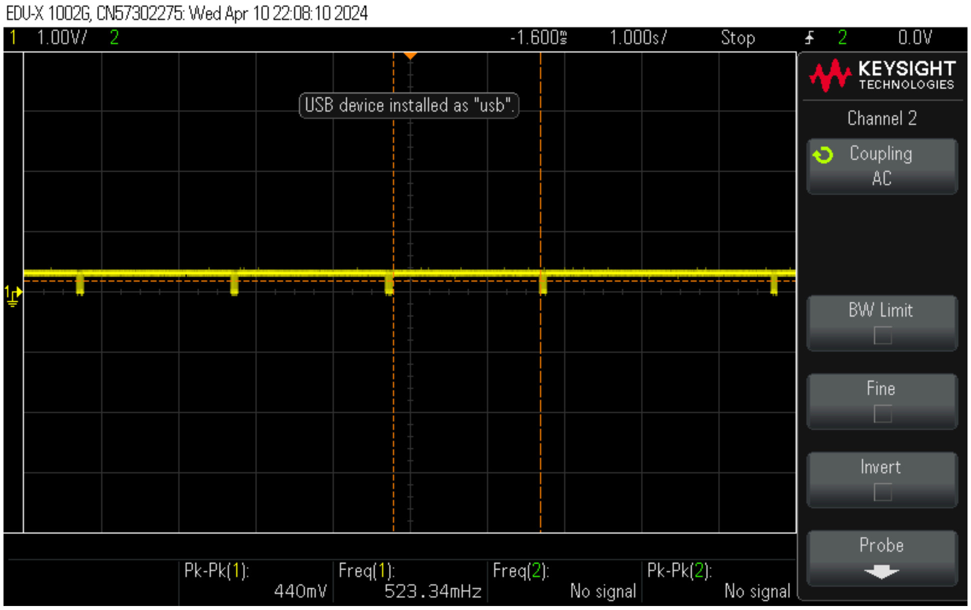


Figure 28: UART transmission of sound effect updates from ESP32 to STM32. The user can change the sound effects multiple times within 1 second.

To ensure the application's stability, it is crucial to test its ability to handle multiple consecutive signal transmissions without crashing. The oscilloscope image (refer to Figure 22) displays the results when the probe is connected to the UART transmission pin on the ESP32 that transmits information to the STM32. The effects were modified five times within one second, and each adjustment was successfully transmitted through UART, as shown by the oscilloscope readings.

To verify that the STM32 was transmitting and receiving data, it was imperative to check the bit clock, word select, and the data line. These were evaluated by placing the oscilloscope probes on the corresponding header pins.

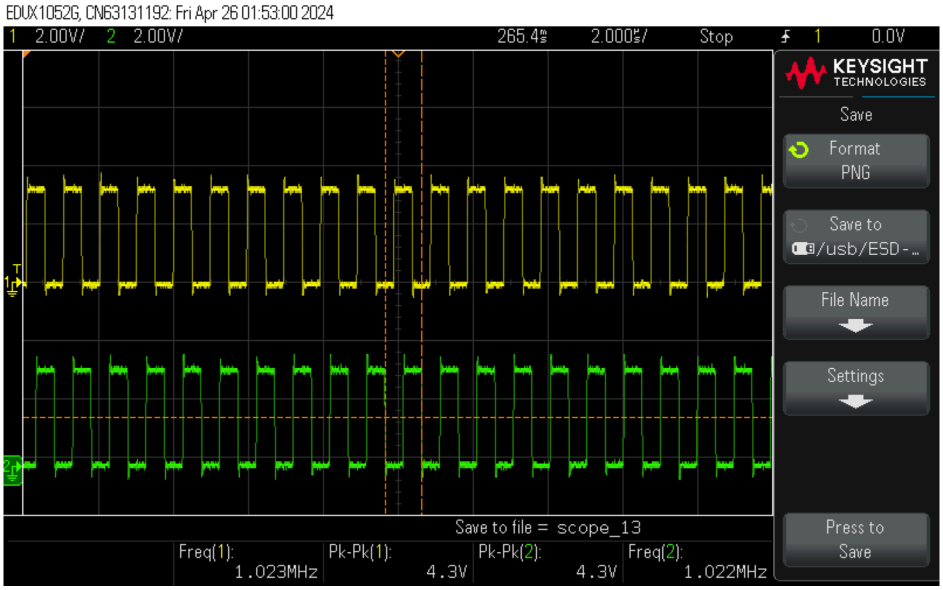


Figure 29: Bit clock transmission from the STM32.

The bit clock is the product of the number of channels, number of bits, and sampling frequency. The I2S channel was configured as stereo, or 2 channels, and 24 bits on a 32 bits frame, so 32 bits of data, with a sampling rate of 16 kHz. 2 * 32 * 16000 = 1.024 MHz. In Figure 23, you can see the frequency of the bit clocks is 1.023 and 1.022 MHz going to the ADC and DAC respectively, showing the bit clocks are operating as intended.

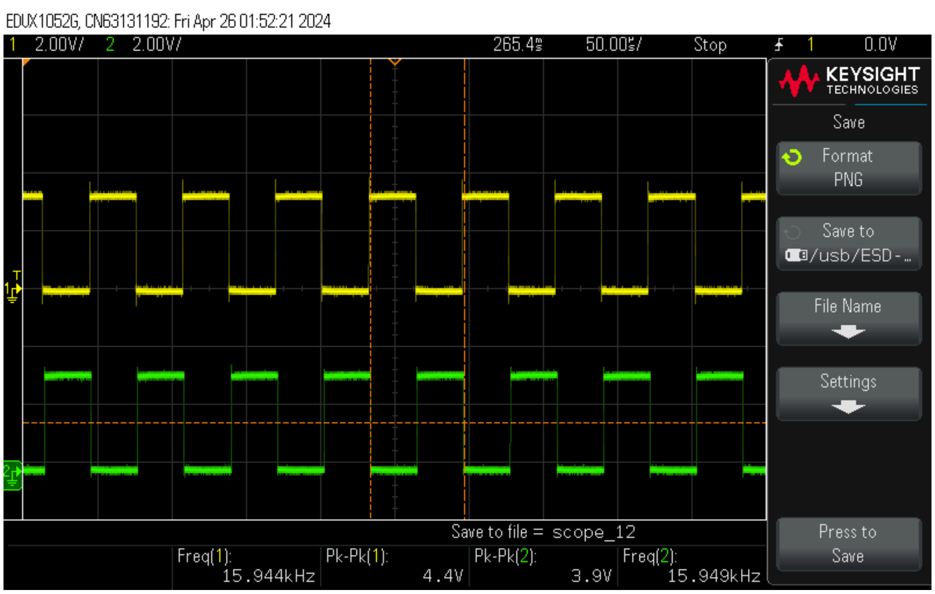


Figure 30: Word select transmission from the STM32.

The word select indicates which channel the sent data is for (left or right). The word select clock should be the sampling frequency (16kHz). The word select lines were 15.944 kHz and 15.949 kHz respectively. The MCU is unable to transmit exactly 16 kHz due to issues with synchronization of the internal clocks, but it is expected to be at 15.957 kHz, therefore it is operating with very little error.

2.7 Subsystem Conclusion

The Android Application was fully functional, able to transmit and receive data through a Bluetooth Low Energy connection. The ESP32-S3 WROOM-1U on the pedal PCB was programmed and verified to be transmitting data to the STM32.

The DSP programming was not able to be completed in time due to a combination of software and hardware issues. Initially the pedal PCB was designed to use the SAI (serial audio interface) pins, but when used with the DMA controller, there were random periodic spikes in the output voltage, no matter the configuration of the audio (16, 24, or 32 bits, master or slave, etc.) Through online investigation of the issue, it was found some STM32 chips have this issue when operated with the DMA controller. To work around this issue, we decided to use two available I2S channels and solder on wires in order to connect I2S to the ADC and DAC. The issue was that this connection was very loose, and any other method of attachment would risk damaging the chip, as a result, the wires were very flimsy and repeatedly fell off, making the debugging process much more difficult.

Despite the code working on the development board, configuring the chip to work with an external ADC and DAC as opposed to an internal one was much more difficult than anticipated. We figured out how to reproduce the input at the output in late April, there was minimal time to debug the written code for the sound effects. If we were given the chance to work on this project again, we would have used a chip that is easier to configure and program. Without prior knowledge of audio processing, it was difficult to figure out how to synchronize the clocks and understand why the only method that worked was 24 bits on a 32 bit frame. Despite understanding the fundamental concepts, configuring the clock was much more difficult to execute than it should have been. If we were given the chance to do this again. We definitely would have simplified the project earlier on so that we could have spent more time on full system testing and validation. We should have prioritized the functionality over complexity of the model.

3. Amplifier Subsystem

3.1 Subsystem Introduction

The amplifier subsystem consisted of a preamplifier as well as an amplifier. The role of the preamplifier is to boost the weak electrical signal from a guitar (in this specific instance, it would boost the signal from the pedal system) for further processing. The preamplifier is built off the TL072 op-amp. This op-amp was chosen due to its low-noise, and high-input impedance, which ensures minimal signal loss. The preamplifier also contains potentiometers for mid, bass, treble, and gain, which allow for frequency-specific equalization; this means that the user is able to modify tonal balance by boosting or reducing the frequencies in these ranges. The gain potentiometer deviates a little from the others in terms of role; it allows for the user to control the input signal strength, which is important for managing the signal-to-noise ratio. Once the signal has been processed inside the preamplifier, the output goes to the input of the amplifier. The LM1875 chip is a power amplifier IC that was utilized. The LM1875 boosts the voltage and current of the preamplifier output, which allows for an audio signal to be outputted out of an 8 ohm 25 W speaker.

3.2 Subsystem Hardware

The preamplifier and amplifier were both made on 2 separate printed circuit boards PCBs. They were attached together using terminal connectors.

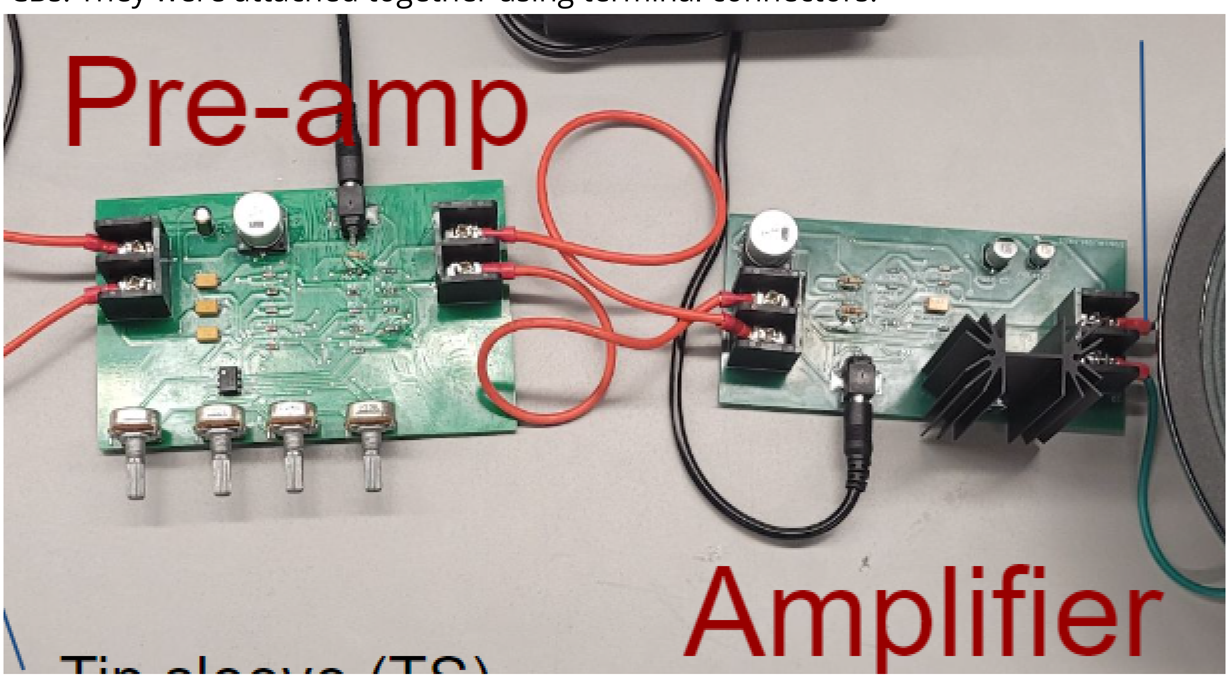


Figure 31. Amplifier PCB setup (preamplifier on the left, amplifier on the right)

3.2 Subsystem Hardware (Cont.)

As seen in the implementation image above, the left-side preamplifier contains the op-amp (bottom left) as well as the potentiometers which allow for tone controls. The amplifier on the right contains a LM1875 chip on the bottom right of the board, while having more capacitors, which is crucial for power amplification stages where signal stability and power handling are paramount. The red wires in the middle of both is the audio signal transmission path, which carries the processed signal from the preamplifier to the amplifier.

3.3 Subsystem Validation

Through the capstone course, the subsystem was validated in multiple ways. The subsystem as a whole went through a rigorous testing process. This included conducting multiple tests with a multitude of signal types: pure tones, complex tones, range sweep, etc.

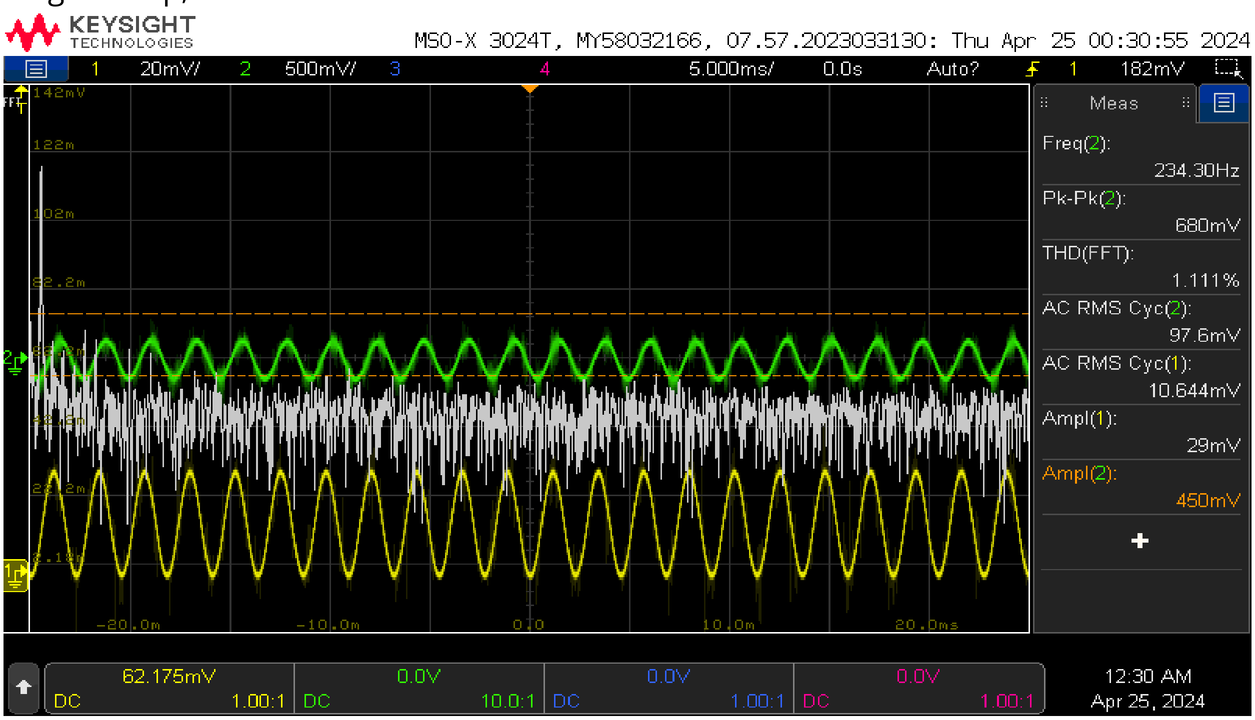


Figure 32: Signal response from the amplifier (440 Hz wave)

As seen, when being sent a 440 Hz wave at 500mV, the amplifier outputs an amplified wave at ~1.11% THD. What THD represents is total harmonic distortion. The reason the THD needs to be low is because it is an indicator of how noisy your system output is. Low THD means that the amplifier is able to produce signals more cleanly and accurately. In this case, since our THD is around 1.11%, it means that ~99% of our input signal is able to be reproduced. Although 1.11% seems extremely low, this is still a high number, as professionally designed amplifiers usually have a THD of about .01%.

For the amplifier design, it was taken from the TI LM1875 datasheet. The specific application that was used was one for a single DC supply.

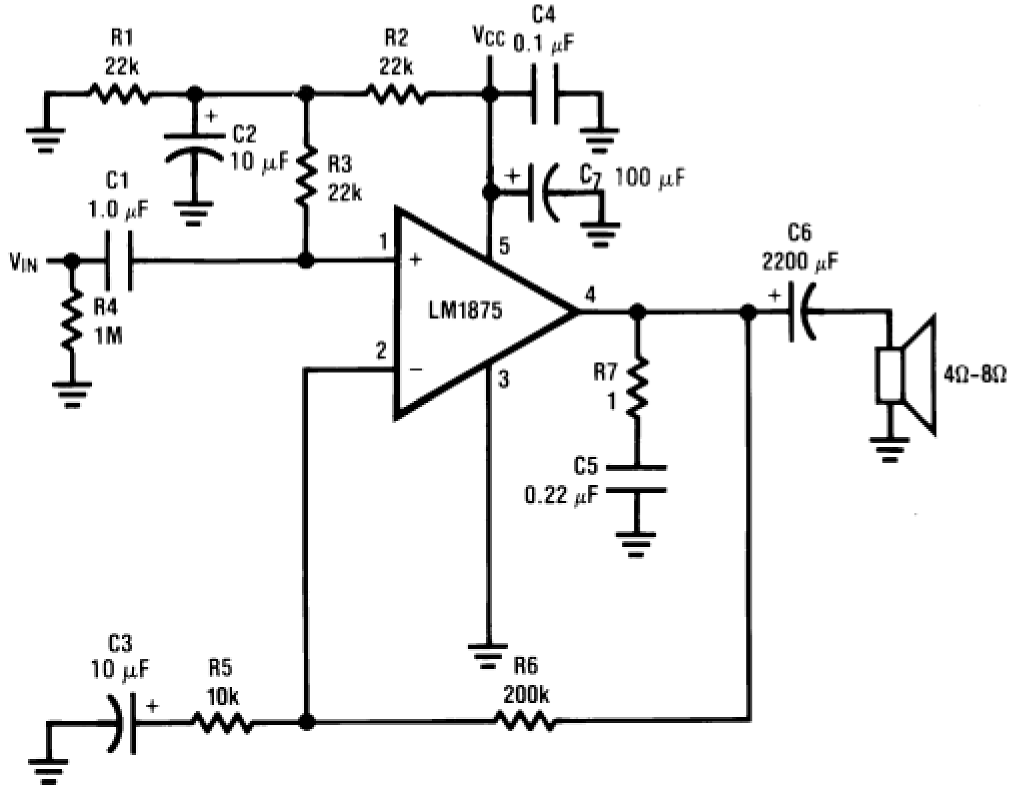


Figure 33: Amplifier design via TI datasheet

This schematic shows the typical LM1875 application. It includes a non-inverting configuration where the gain is set by the resistor ratio of R3 to R7 and is stabilized for high frequencies by C5. The output to the speaker includes a Zobel network to counteract the inductive nature of the speaker load and ensure amplifier stability. The circuit is designed for straight-forward operation with minimal external components, suitable for driving a speaker with impedances between 4-8 ohms.

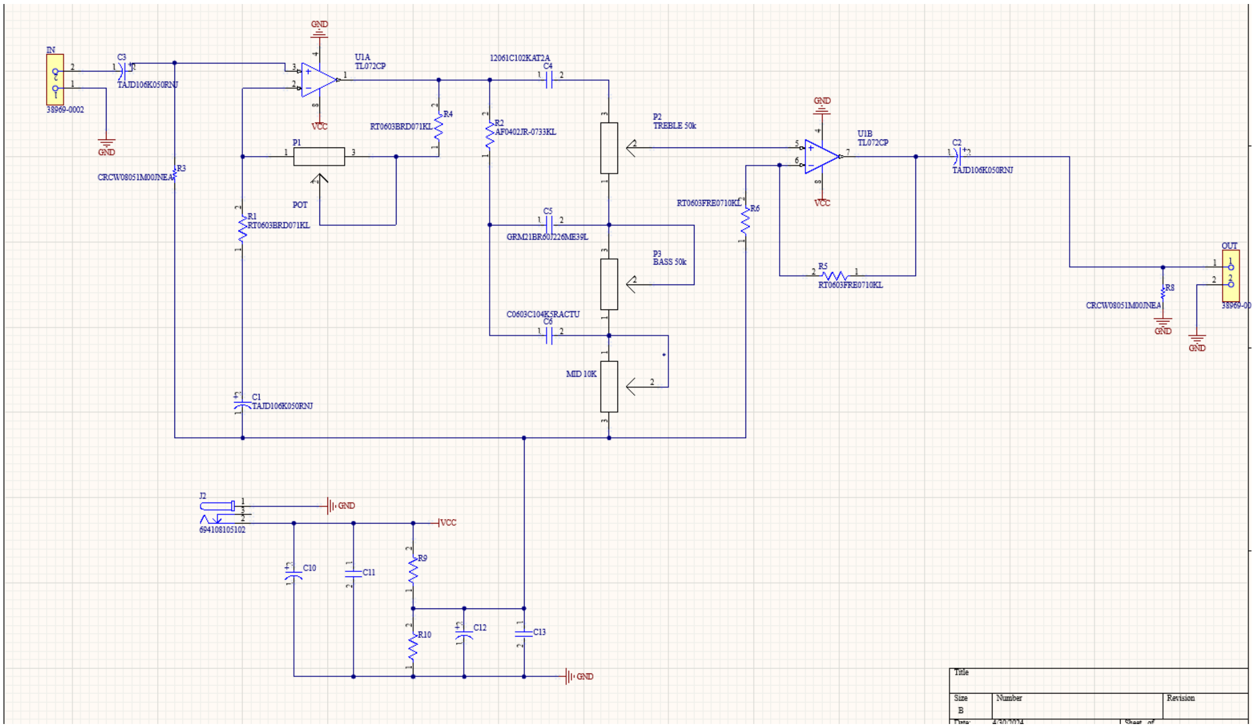


Figure 34: Preamplifier Design

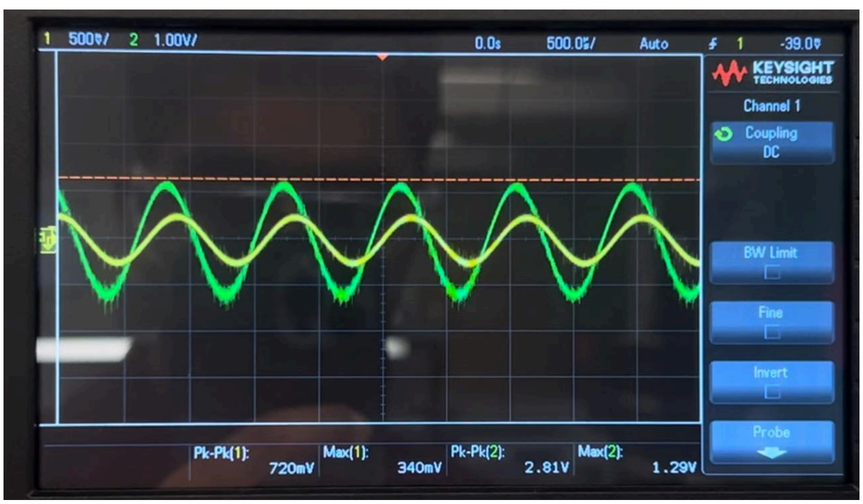


Figure 35: Amplifier Signal Response

As seen above, where the yellow wave represents input and the green wave represents output, the amplifier demonstrated an effective amplification capability when tested with a 1 kHz sine wave input. The input signal, characterized by a maximum voltage of approximately 340 mV and a peak-to-peak voltage of approximately 720 mV, was successfully amplified to an output signal with a maximum voltage of around 1.3 V and a peak-to-peak voltage of approximately 2.8 V. This corresponds to a gain of about 3.82, indicative of the amplifier's robust amplification performance. Moreover, the output maintained the integrity of the sine wave form, suggesting that the amplifier exhibits good linearity and low distortion, which are critical for preserving the quality of the audio signal. These results confirm that the amplifier that was designed and built is capable of effectively enhancing the signal strength without compromising the signal quality, fulfilling key design objectives for audio fidelity in guitar amplifiers.

3.4 Subsystem Failure Analysis

Looking at what went wrong, the preamplifier potentiometers failed to operate as intended, with the potentiometers designed for frequency-specific adjustments uniformly functioning as generic gain controls across all signal frequencies. This highlighted significant shortcomings in our simulation processes, which, although accurate in modeling ideal circuit behavior, failed to account for practical variations such as component tolerances and the non-ideal characteristics of operational amplifiers. Additionally, the design oversight, particularly in the selection and configuration of components and circuit layout, likely led to unintended interactions such as parasitic capacitance or inductive coupling. The issues encountered underline the necessity for more rigorous, varied, and realistic simulation parameters, coupled with comprehensive physical testing under diverse operating conditions to ensure circuit reliability and functionality. This experience has provided valuable insights into the importance of integrating robust simulation and thorough testing phases in circuit design to better mimic actual operating conditions and prevent similar discrepancies in future projects.

3.5 Subsystem Conclusion

Throughout the development of our guitar system's preamplifier and amplifier boards, we achieved significant milestones in terms of functionality and output accuracy. However, not all components performed optimally, particularly the potentiometers, which did not function as intended. Despite robust testing designed to validate overall system operation, these issues with the potentiometers were not identified, highlighting a gap in our testing approach. To prevent such oversights in future projects, we will incorporate more comprehensive digital simulations that focus on individual components like potentiometers. This step will help verify functionality at a more granular level before proceeding to physical assembly. The performance of the preamplifier and amplifier, however, met our primary objectives. The preamplifier adeptly conditioned the signal from the pedal board, effectively preparing it for amplification. The amplifier then successfully boosted this conditioned signal, preserving the original tone and minimizing any loss of signal quality. This successful core functionality not only confirms the effectiveness of our design but also demonstrates the potential of our system to meet professional audio processing standards. Moving forward if needed, it would be good to refine our design and testing methodologies. By enhancing our simulation protocols and expanding our testing procedures, we can aim to improve the reliability and performance of our system, ensuring that even components like potentiometers meet the high standards required for professional audio equipment. This project has laid a solid foundation for future innovation in audio system design, paving the way for further advancements and professional application.

1. System Validation

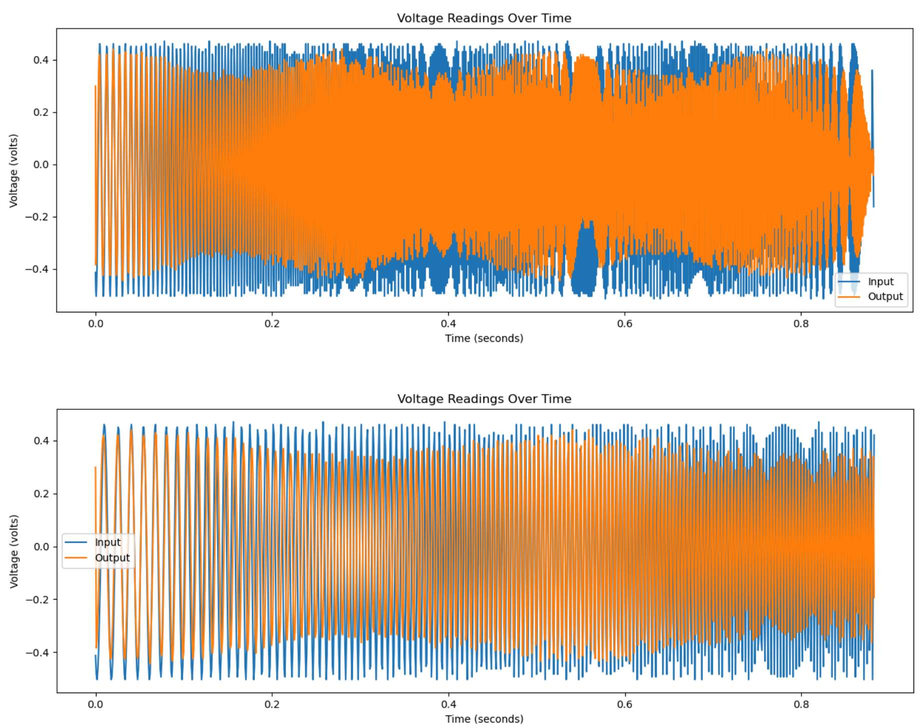


Figure 36: Frequency sweep on Pedal PCB

In Figure 25, it displays the frequency sweep from 20 Hz to 20 kHz. At some intermediate frequencies, the gain decreases, this could be due to capacitance and filtering in the pedal PCB. At all frequencies the signal remains visible and audible.

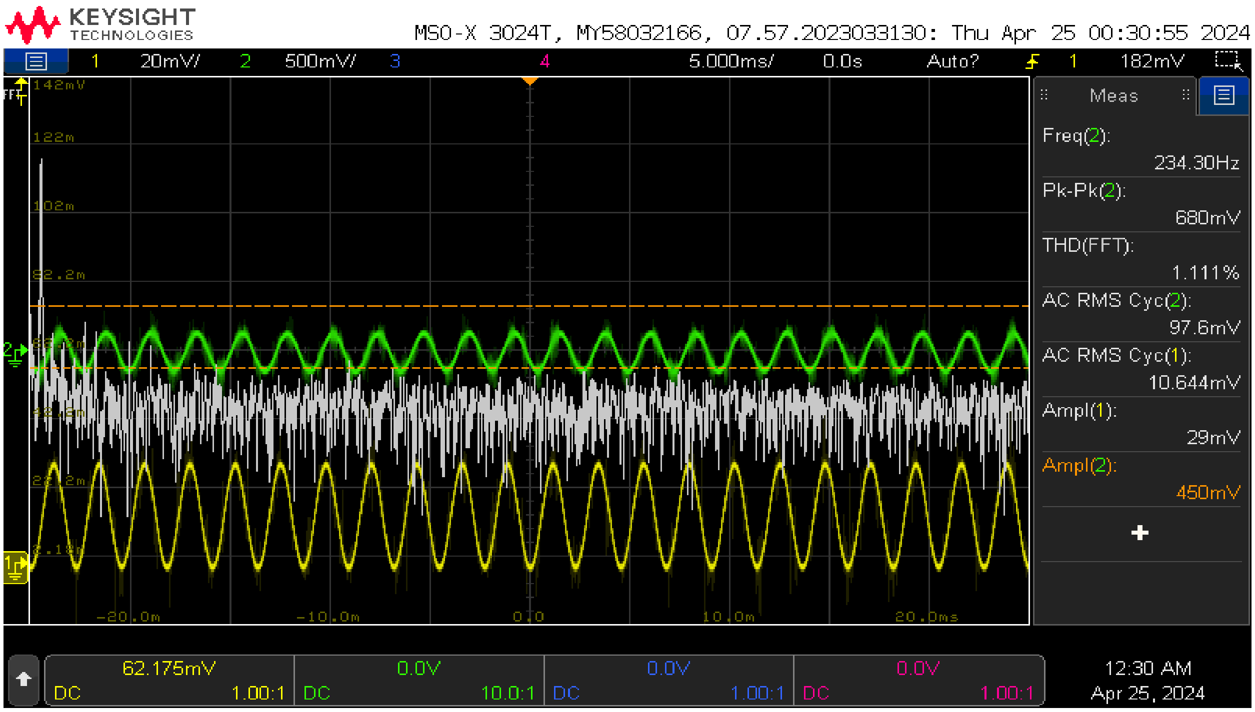


Figure 37: Amplifier response to pedal PCB input

As seen, in this test, the pedal PCB as the signal source instead of a conventional function generator to simulate real-world conditions more accurately. Analyzing the waveform traces, the yellow line (amplifier output) clearly shows effective amplification of the green line (pedal output, and in turn, amplifier input), with the amplifier maintaining the integrity of the waveform and significantly increasing the amplitude. The THD measured at approximately 1.11% indicates minimal harmonic distortion, preserving the quality of the original audio signal, which is essential for high-fidelity applications. Furthermore, the frequency response remains consistent with an output signal frequency of about 234.30 Hz, matching the input and affirming the system's capability to handle standard audio signals without frequency bias. This robust performance in handling real pedal board outputs confirms the system’s readiness for practical use and its potential in professional audio applications. The successful validation points towards the system's commercial viability and lays the groundwork for further comprehensive evaluations including user testing and performance under varied operational conditions.