ECE 445
Spring 2025
Design Document:

Digital Pitch Shifter for Guitar

Team #60:
William Chang (wqchang2)
Eric Moreno (emoren40)
Zhengjie Fan (zfan11)

TA: Shengyan Liu
Professor: Michael Oelze

1 Introduction

1.1 Problem

There are new guitar players every year that learn how to use the guitar and continue to advance their skills. Players become more advanced and eventually learn more technical skills. Unfortunately, there are different types of guitars that can perform in different ways. Guitarists without access to a tremolo system face significant limitations in their ability to create expressive vibrato and pitch-bending effects, which are essential for adding emotional depth and dynamic variation to their playing. Without these techniques, the guitar's sound can feel static or restrained, especially in genres like rock, blues, and jazz, where pitch manipulation is crucial. Traditional tremolo systems, though effective in addressing this issue, require invasive modifications to the guitar body, such as routing or altering the bridge. These changes not only compromise the guitar's original design but can also affect its sound and value. Additionally, such systems may not be suitable for all playing styles, or for guitarists who prefer a more minimalist approach. As a result, players seeking greater versatility in their instrument face the difficult choice between sacrificing their guitar's aesthetics or settling for limited expressive capabilities. This is the gap the proposed project aims to fill.

1.2 Solution

The solution to the aforementioned issue is a compact, attachable digital pitch-shifting device that uses a sonic sensor to detect the proximity of the guitarist's hand to the bridge of the guitar. This will be so that the user can attach this component to any electric guitar with a jack cable that would normally be attached to an amplifier. The user would connect their guitar in the system and this will begin the modulation of any signal put through by having the sensor and microcontroller work together in an algorithm. This software will pitch the guitar up and down according to what the user decides.

1.2 Visual Aid

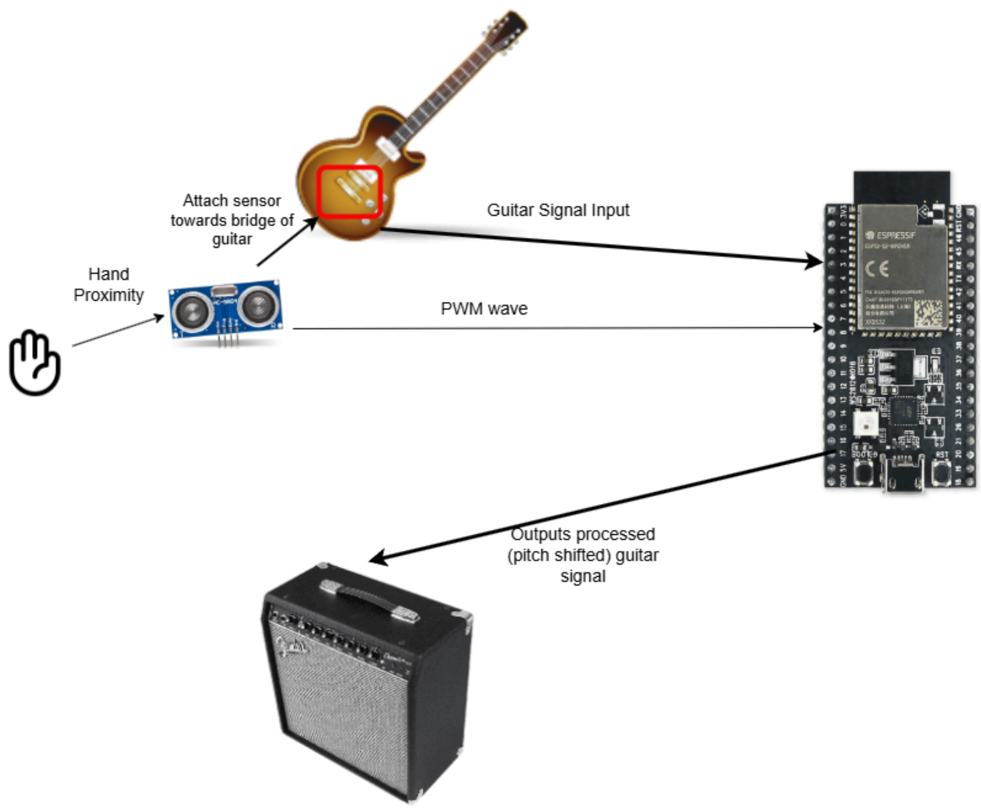


Figure 1: Design Overview

1.3 High Level Requirements

The project may only be successful under the following conditions:

Real-Time Pitch Shifting with Low Latency

- The system must process incoming guitar signals and shift their pitch without perceptible delay (<10ms total latency) to ensure natural playability.

High-Fidelity Audio Processing

- The pitch-shifted output must maintain at least 48 kHz sampling rate and 12-bit depth for ADC and 8-bit depth for DAC. (ESP32 microcontroller)

User-Controlled Pitch Adjustment

- The device must allow adjustable pitch shifting from -2 to +2 semitones.

2 Design

2.1 Block Diagram

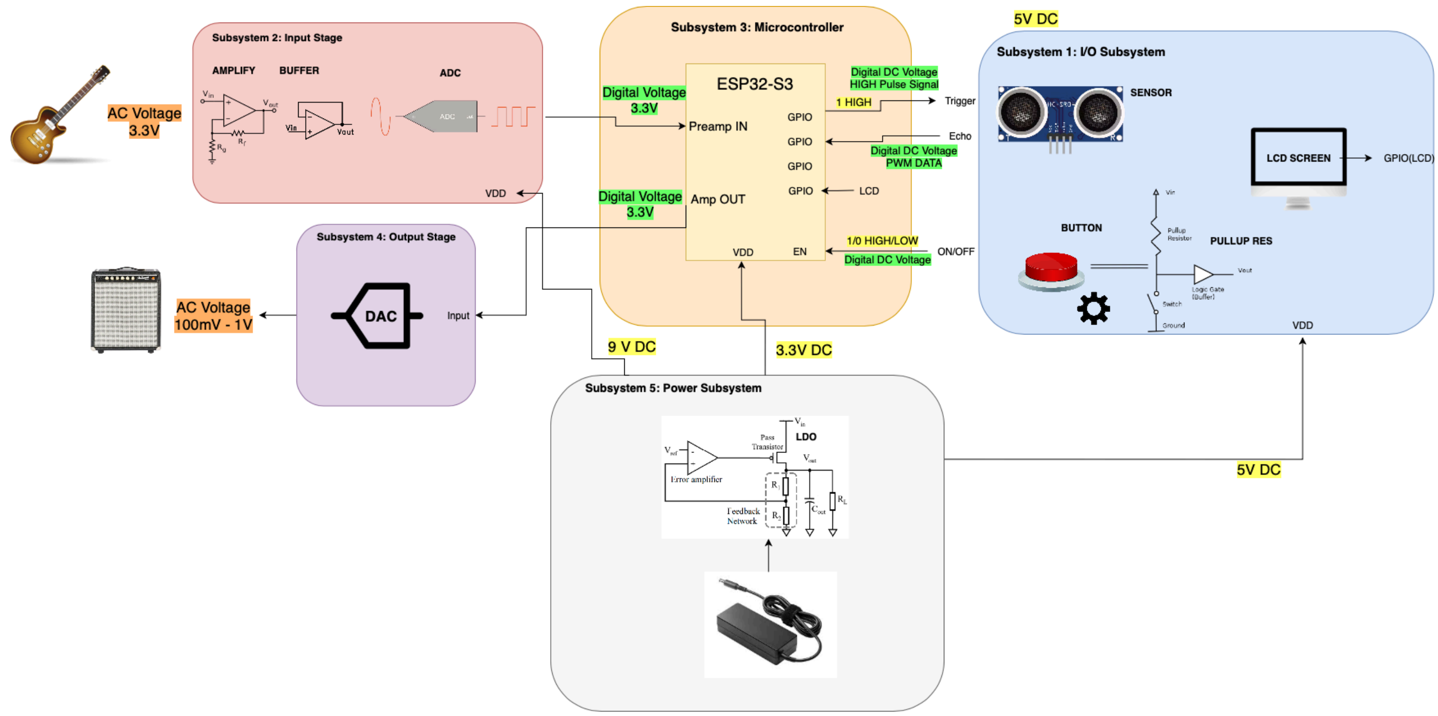


Figure x: Project Block Diagram

Our design consists of 5 different subsystems that include the ADC & preamplification of a signal to be controlled by the microcontroller. A sensor that sends back data to the microcontroller and then finally to our guitar's amplifier through a DAC. All being powered by our LDO step down subsystem to ensure correct rated voltages.

2.3 Functional Overview & Block Diagram Requirements

2.3.1 Input Stage Subsystem

The preamp subsystem is designed to condition the guitar's signal before it is processed by the ESP32 microcontroller. A guitar preamp pedal is used to boost the signal, ensuring it reaches a level suitable for the ESP32-S3. In this subsystem, the guitar's signal is first connected to a 1kΩ resistor, while a 3.3V voltage source is fed through another resistor. These signals are then processed by two op-amps, which both amplify and buffer the voltage. This ensures the signal is stabilized, eliminating any negative voltage and reducing unwanted noise. The conditioned signal is then fed into an ADC, which converts the analog signal into a digital format that the ESP32 microcontroller can interpret. The entire preamp subsystem is powered by a 9V power supply from the power subsystem, ensuring stable operation. This setup guarantees that the microcontroller receives a clean, noise-free signal within its limited voltage range for accurate processing.

Requirements	**Verifications**
The system must take in a voltage and	

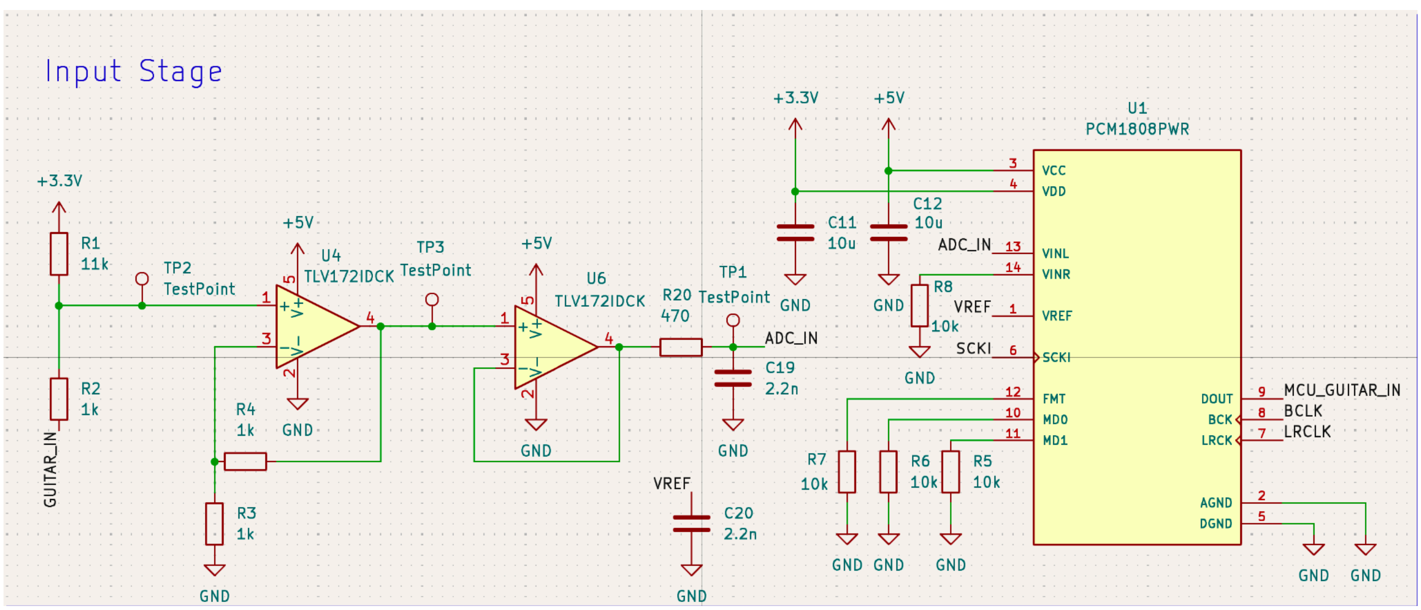
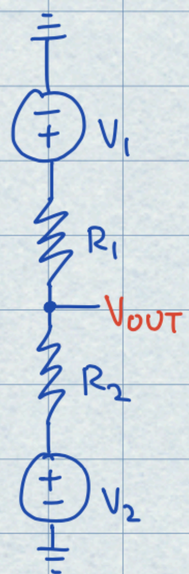


Figure x: Input Stage Subsystem Schematic

DC offset



$$\frac{V_1 - V_{OUT}}{R_1} = \frac{V_2 - V_{OUT}}{R_2}$$

$$\frac{V_1}{R_1} - \frac{V_2}{R_2} = \frac{V_{OUT}}{R_1} - \frac{V_{OUT}}{R_2}$$

$$\frac{R_2V_1 - R_1V_2}{\cancel{R_1R_2}} = \frac{R_2V_{OUT} - R_1V_{OUT}}{\cancel{R_1R_2}}$$

$$R_2V_1 - R_1V_2 = (R_2 - R_1)V_{OUT}$$

$$V_{OUT} = \frac{R_2V_1 - R_1V_2}{R_2 - R_1}$$

☆ $V_1 = 3.3V$ (from power supply)

☆ $V_2 \approx 0.3\sin(\omega t)\ V$ (guitar signal) test to see voltage range

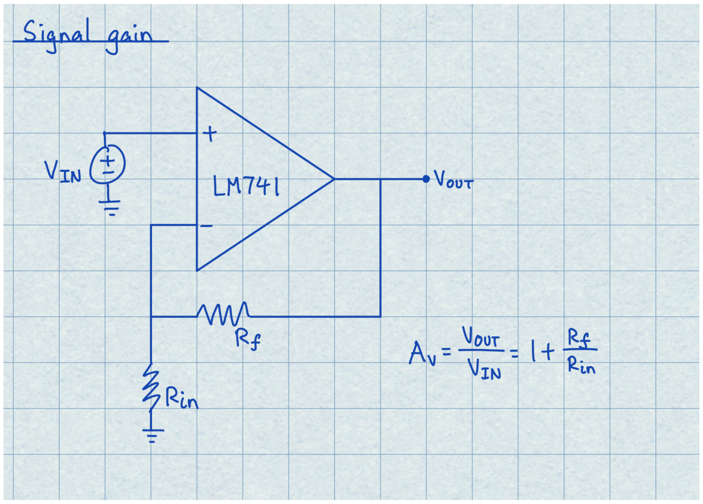
Choose R_1 and R_2 $\longrightarrow$ $0 \le V_{OUT} \le 1$ -min value = 0 V

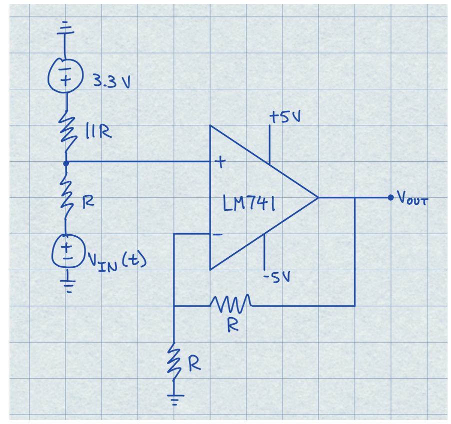
$$V_{OUT} = \frac{R_2V_1 - R_1V_2}{R_2 - R_1}$$

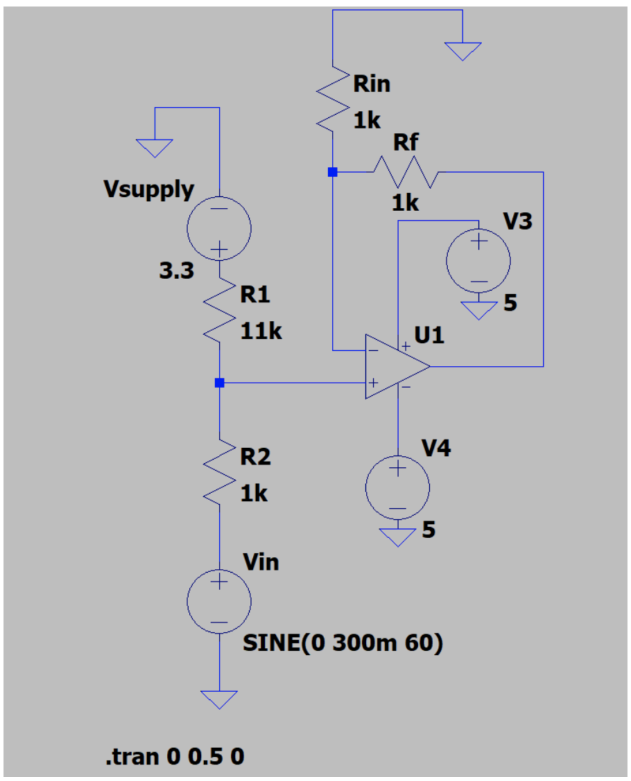
lowest sine wave voltage

$$0 = \frac{R_2(3.3) - R_1(-0.3)}{R_2 - R_1}$$

$0 = 3.3R_2 + 0.3R_1 \longrightarrow R_1 \ge 11R_2$ make sure $V_{OUT} \ge 0$







2.3.2 Microcontroller Subsystem

The ESP32 microcontroller acts as the core processing unit, making both the signal input/output operations and performing real-time signal processing critical to the system's functionality. A key responsibility of the ESP32 is managing audio data, utilizing its Analog-to-Digital Converter (ADC) and Digital-to-Analog Converter (DAC) interfaces. The analog guitar signal is first digitized by the ADC, then processed using pitch-shifting algorithms. Following processing, the signal is reconverted into an analog form via the DAC, allowing it to be output to a guitar amplifier for playback. Beyond audio signal processing, the ESP32 interfaces with an HC-SR04 ultrasonic sensor to capture real-time hand proximity data. It generates a high trigger pulse on a GPIO pin to initiate the sensor's distance measurement cycle. The pulse remains active to ensure continuous data readings, which are then returned to the microcontroller. The ECHO pin output is subsequently read by the ESP32 to calculate the distance between the sensor and the player's hand. This distance is used as a dynamic input, modulating the pitch-shifting parameters in real time. For precise and stable operation, the microcontroller relies on both internal and external clock sources to synchronize its processes and improve the accuracy of real-time pitch adjustments. The system is powered on and configured via a button in the IO Subsystem, which also allows the user to select between different operating modes or settings. Additionally, the microcontroller integrates strapping data from the software to ensure proper initialization during startup. The ESP32 also manages user interactions, such as effect toggling or parameter adjustments, through GPIO pins connected to external buttons or switches. This enables flexible user control over the system's functionality in a seamless manner.

Requirements	**Verifications**

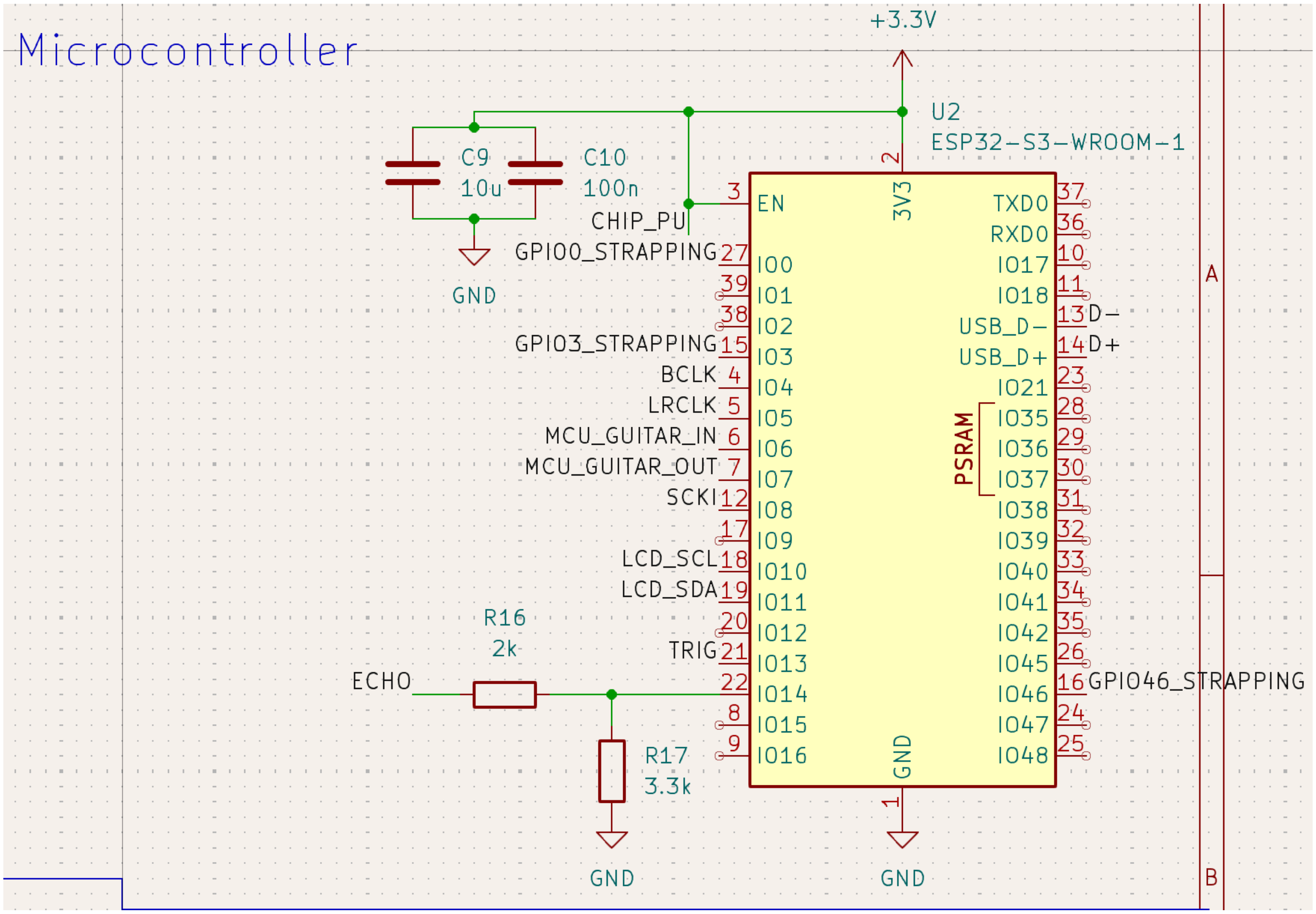


Figure x: Microcontroller Subsystem Schematic

2.3.3 I/O Subsystem

The I/O subsystem integrates multiple components to facilitate user interaction and system functionality. The ultrasonic sensor is a key input device, providing real-time distance measurements for processing. Pull-down resistors are implemented to ensure that push buttons remain in a consistent low state when not pressed, preventing unintended signals. A Vbus connection is included to enable USB connectivity, allowing for programming and power delivery to the system. Additionally, an LCD screen is incorporated to display essential information, including component statuses and labeled outputs, ensuring clear user feedback. To enhance system configuration, GPIO strapping is implemented, allowing certain pins to define system behavior during startup. These strapping pins help configure the boot mode, clock settings, and other essential parameters, ensuring that the system initializes correctly and operates as intended without requiring manual intervention each time the device is powered on.

Requirements	**Verifications**

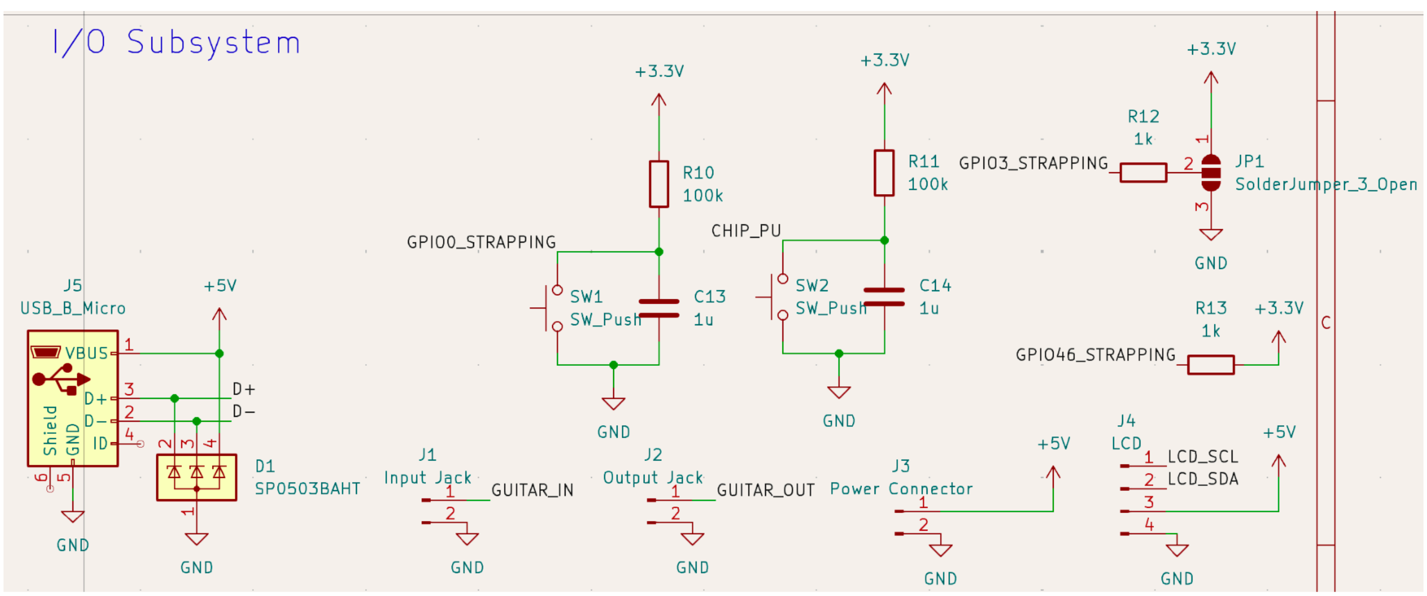


Figure x: I/O Subsystem Schematic

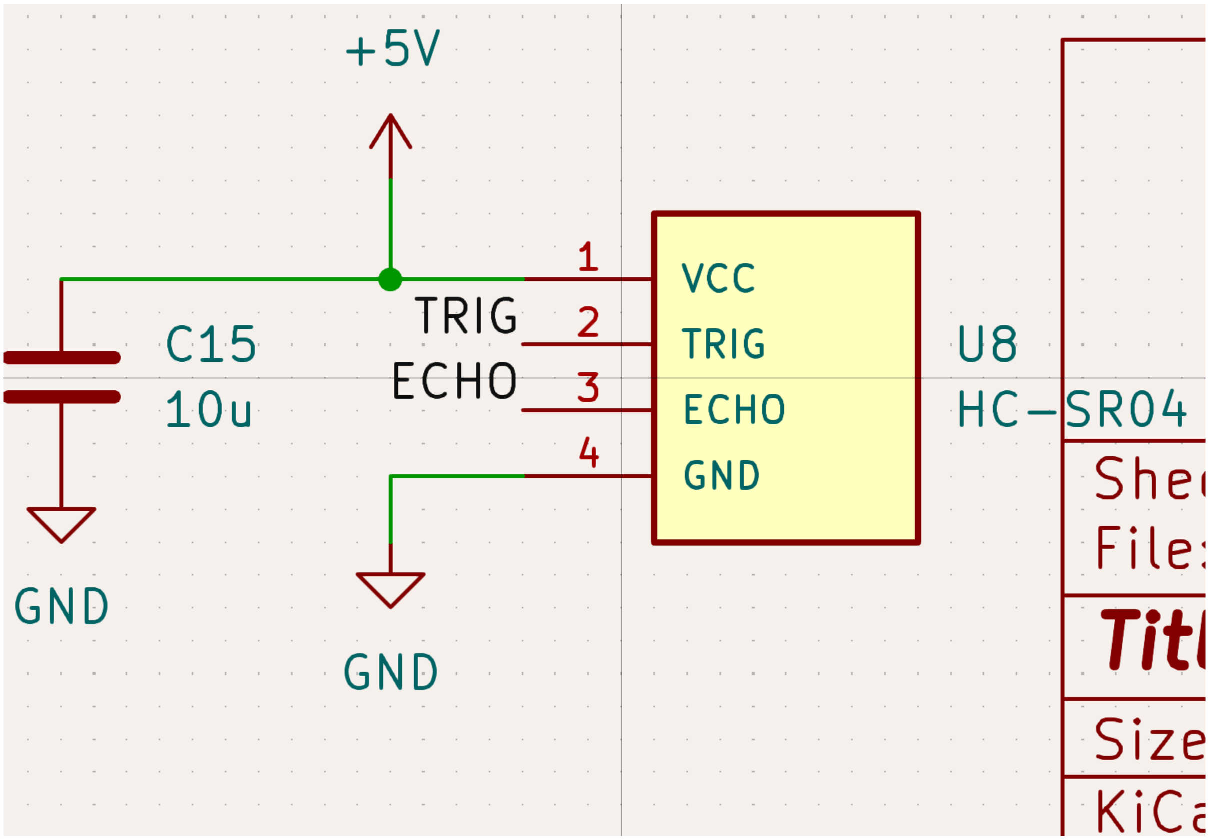


Figure x: Sensor Schematic

2.3.4 Output Stage Subsystem

The output stage of our project is responsible for converting the modified digital audio signal from the ESP32 microcontroller back into an analog signal that the guitar amplifier can process. This is achieved using a Digital-to-Analog Converter (DAC), which reconstructs the waveform from the .wav file generated based on sensor measurements. To ensure accurate signal reproduction, the DAC operates using the same internal and external clocks as the microcontroller and ADC, maintaining synchronization and preventing artifacts in the audio output. Since the ESP32 DAC outputs a signal between 0V and 3.3V, it must be shifted and conditioned to match the input requirements of the guitar amplifier. The amplifier's instrument input expects a signal level of approximately 100mV to 1V peak-to-peak. To achieve proper signal conditioning, the DAC output undergoes DC offset removal using a capacitor, ensuring that the signal is centered around 0V rather than the ESP32's 1.65V bias. Additionally, a voltage divider or an op-amp circuit is used to scale the signal to the appropriate voltage range.This stage is essential for preserving the integrity of the digitally modified audio while ensuring that the output signal is properly formatted and voltage-matched for seamless amplification.

Requirements	**Verifications**

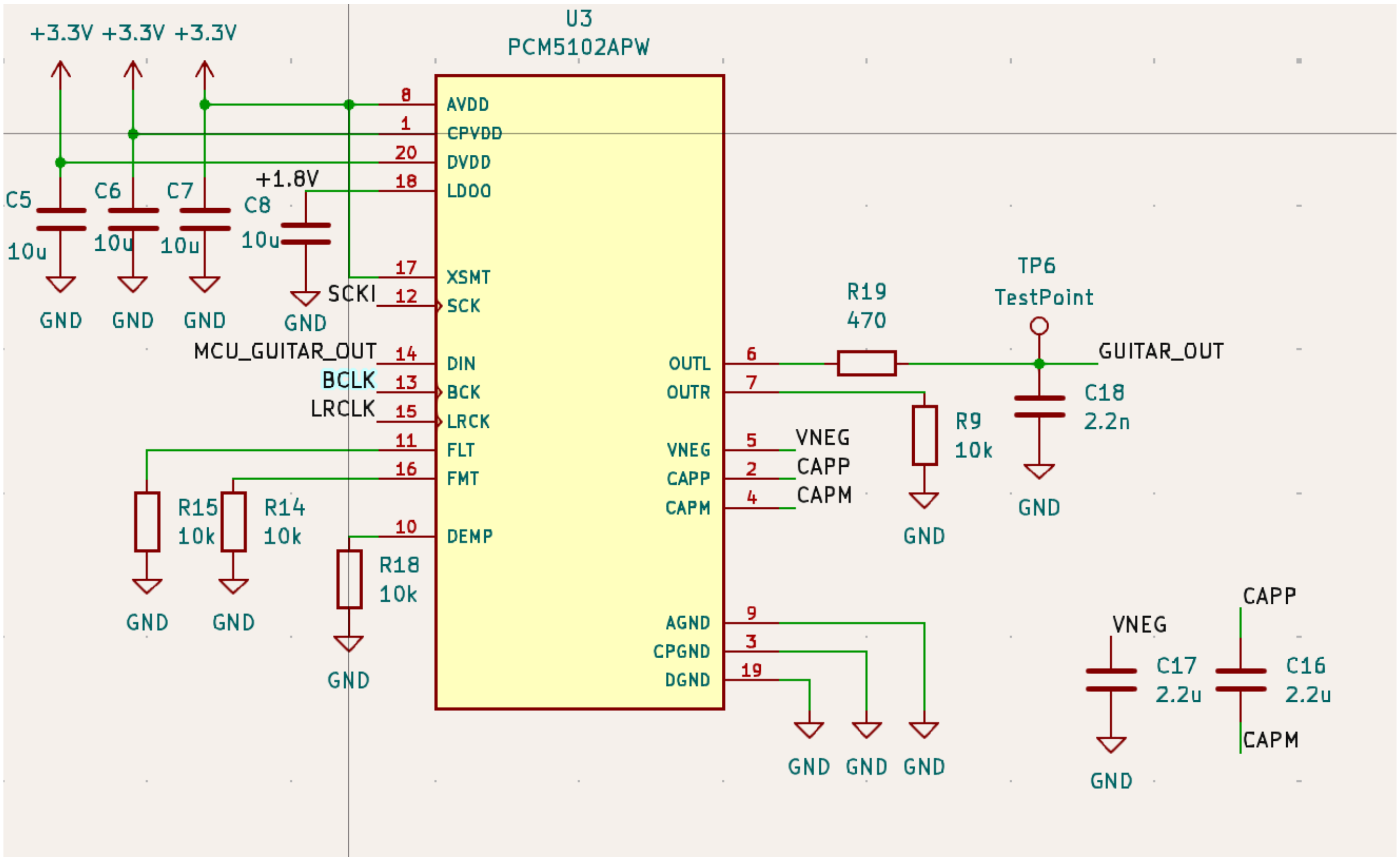


Figure x: Output Stage Subsystem Schematic

2.3.5 Power Subsystem

The power management system ensures stable operation of both the ESP32 microcontroller and the HC-SR04 ultrasonic sensor using a 5V power supply. Since the ESP32 operates at 3.3V, a Low Dropout Regulator (LDO) is used to step down the 5V supply, providing a stable 3.3V output. The LDO regulator efficiently converts the voltage while minimizing power loss and ensuring a steady supply to the microcontroller. The HC-SR04 sensor, which requires 5V, is powered directly from the same 5V source to maintain proper functionality. A common ground is shared across all components to ensure reliable communication. Additionally, since the HC-SR04's Echo pin outputs a 5V signal, a voltage divider is used to safely step down the signal to 3.3V, preventing damage to the ESP32's GPIO while allowing accurate signal detection.

Requirements	Verifications

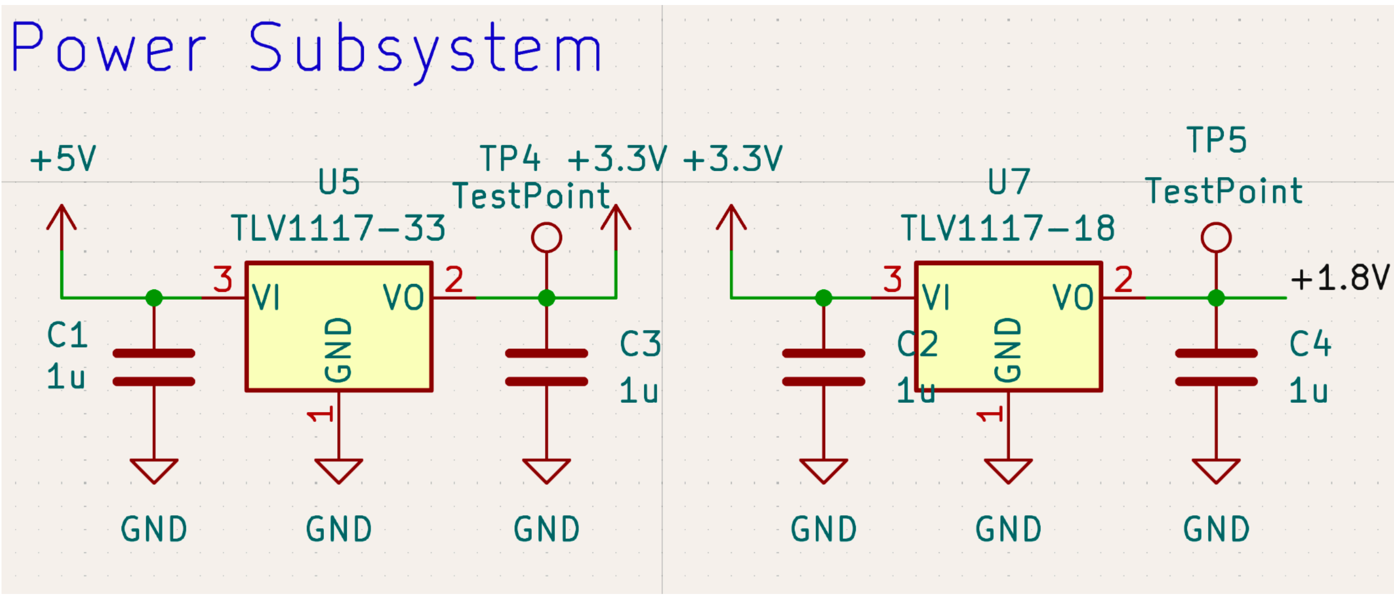


Figure x: Power Subsystem Schematic

2.4 System Software Logic & Requirements

2.4.1 Overview

2.4.2 Interface

2.4.3 Design Decisions

2.4.4 Requirements & Verification

2.5 Tolerance Analysis

3 Cost & Schedule

3.1 Cost Analysis

3.1.1 Parts & Materials

The process of picking out parts was done by overseeing the course website

3.1.2 Estimated Hours of Compensation

The members in the project are all Grainge College of Engineering students who are studying Electrical Engineering. The post-graduate average pay according to the Grainger College of Engineering website, the average starting salary for an Electrical Engineering graduate is $87,769 per year [5] which equates to $42.20 per hour.

Category **Estimated Hours**

	Eric (100)	**William (120)**	**Fan (95)**
Circuit Design	**0**	**30**	**0**
Software Design	**0**	**0**	**30**
PCB Layout	**15**	**15**	**0**
Soldering	**15**	**15**	**10**
Prototype & Debug	**40**	**40**	**40**
Documentation	**30**	**20**	**15**

Table x: Estimated Hours

Using the hours in table x and the rated pay found through research, we can compute the estimated cost for overall labor as:

$42.20(Hourly Rate) * 315(Total Hours) = $4,960

Note that this is an averaged amount taking taxes into consideration.

3.1.3 Resources

3.1.4 Total Cost

3.2 Schedule

- Week 1, 1/20

  Initial web board post
- Week 2, 1/27

  CAD assignment
- Week 3, 2/3

  Project approval
- Week 4, 2/10

  Soldering assignment due , draft proposal review and review early schematics of preamplification stage
- Week 5, 2/17

Proposal review due, begin early stages of KiCad schematic

- Week 6, 2/24

  Finish KiCad Schematic, begin drafting design document. Order parts for breadboard.

- Week 7, 3/03

  Design document due, begin construction of breadboard prototype

-
-
-
-

4 Ethics & Safety

4.1 Ethics

Ethical Considerations

Our digital pitch-shifting guitar project must follow ethical guidelines set by the IEEE and ACM Codes of Ethics to ensure fairness, responsibility, and safety in both its development and use.

1. Intellectual Property and Fair Use
   - Since our project may involve existing signal processing techniques or open-source software, we must ensure that we properly cite and follow all licensing agreements.
   - The IEEE Code of Ethics emphasizes honesty in authorship and respect for intellectual property. To follow this, we will credit all sources, avoid plagiarism, and ensure that any external components we use are legally allowed.

2. User Safety and Responsible Design
    - High audio output levels could damage hearing if not properly managed. To prevent this, we will limit volume levels, implement safety features, and test the device under different conditions.
    - The ACM Code of Ethics encourages designing systems that improve quality of life while minimizing harm. Our design will ensure that users are protected from sudden loud noises or unintended hardware malfunctions.
3. Accessibility and Inclusivity
    - Our device should be usable by all musicians, regardless of experience level. We will provide clear documentation, an easy-to-use interface, and setup guides to help users understand how to operate it safely and effectively.
    - Ensuring inclusivity aligns with the ACM Code of Ethics, which states that computing professionals should make technology accessible to a wide audience.

4.2 Safety

Safety Considerations

Our project includes electronic and software components, each of which presents potential safety risks. To ensure safe operation, we will follow industry standards and regulations while implementing features to minimize hazards.

4.2.1 Circuit Protection Safety

Our device operates on low-voltage DC power, but protection against short circuits, overvoltage, and overheating is still essential to prevent damage or injury. To mitigate these risks, fuses, voltage regulators, and a physical containment box are used to safeguard both the device and the user. Additionally, capacitors and resistors are strategically placed in the circuit to help filter voltage spikes, stabilize power delivery, and limit excessive current flow, further reducing potential electrical hazards.

4.2.2 Personal Health Safety

Prolonged exposure to loud audio can lead to permanent hearing damage, with OSHA Noise Exposure Standards indicating that sounds above 85 dB can be hazardous over extended periods. To mitigate this risk, our design will incorporate built-in volume limits and gain control features to prevent excessive sound levels from being output. Additionally, warning indicators or visual feedback may be integrated to alert users when sound levels approach unsafe thresholds. Beyond device features, personal safety measures will be emphasized in the user manual, educating musicians on the risks of prolonged exposure to high sound levels. Recommendations may include taking regular breaks, using hearing protection (e.g., earplugs), and maintaining safe listening distances. These precautions will help ensure that users can enjoy enhanced audio experiences without compromising their long-term hearing health.

4 References

1. SparkFun Electronics, "Ultrasonic Distance Sensor - HC-SR04," [Online]. Available: https://www.sparkfun.com/ultrasonic-distance-sensor-hc-sr04.html. [Accessed: 13-Feb-2025].

2. Espressif Systems, *ESP32-S3 Datasheet*, [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-s3_datasheet_en.pdf. [Accessed: 13-Feb-2025].
3. Espressif Systems, *ESP-IDF I2S API Reference (ESP32-S3)*, [Online]. Available: https://docs.espressif.com/projects/esp-idf/en/stable/esp32s3/api-reference/peripherals/i2s.html. [Accessed: 13-Feb-2025].
4. Digi-Key Electronics, *What Is the I2S Communication Protocol?*, [Online]. Available: https://www.digikey.com/en/maker/tutorials/2023/what-is-the-i2s-communication-protocol. [Accessed: 13-Feb-2025].
5. "Salary Averages." Electrical & Computer Engineering at Illinois, University of Illinois Urbana-Champaign, https://ece.illinois.edu/admissions/why-ece/salary-averages. [Accessed 4-Mar-2025].