# Automatic Guitar Tuner Final Report

By:

Ethan Lin

Nathan Kim

Final Report for ECE 445, Senior Design, Spring 2025

TA: Shenyang Liu

07 May 2025

Team 65

## Abstract

This paper presents an automatic guitar tuner that simultaneously adjusts all six strings to standard tuning (EADGBE) with ±12 cents accuracy, eliminating manual peg adjustments. The system uses a piezoelectric transducer to capture string vibrations, processes frequencies in real time via an STM32H7 microcontroller running FFT analysis, and drives six torque-limited motors via L298N H-bridges to rotate tuning pegs. Key results include: tuning completion within 60 seconds, robust vibration sensing in noisy environments, and thermal validation of power regulators (LM2937-3.3, µA7805) under load. The device operates on a 9V battery (50-minute runtime) and meets all design criteria, including attachment time (<2 minutes) and torque safety (≤0.5 N·m). Testing confirmed frequency resolution of ±1 Hz across 80–350 Hz, ensuring studio-grade tuning precision. This project demonstrates an embedded solution to a common musician challenge.

## Content

# 1. Introduction

## 1.1 Problem Statement

For many guitar players, keeping their guitar in tune can be a hassle. Looking at the tuners currently on the market, the most common type of guitar tuner is a clip-on tuner where the player is required to manually tune each string using the attached tuner as a pitch guide. There also exist automatic guitar tuners but these are limited by either the number of strings that can be tuned at once, the price of the tuner, or the amount of work needed to be done by the player (i.e. the player still has to move the tuner around the pegs and strum the strings) [1].

## 1.2 Solution

Our solution is to develop a portable automatic guitar tuner that attaches to all six tuning pegs of the guitar and can tune each string to the standard 6-string guitar tuning (EADGBE). The user will intermittently strum all six strings until an LED flashes which indicates that all strings are correctly tuned. To accomplish this task, four essential subsystems are introduced: a power subsystem, motor subsystem, vibration-sensing subsystem, and a processing subsystem.

## 1.3 Criterion For Success

- Ability to attach and remove the system within two minutes total
- Ability to tune all six strings within ±12 cents of the set tone per string (the value where people can start to detect when something is out-of-tune [2])
- Ability to finish tuning all strings within a minute, flashing an LED to visually signify completion

## 1.4 Broader Impact

This project has potential positive impacts across multiple domains. Economically, this project offers musicians an affordable alternative to expensive professional tuning devices. Going broader, societally, the device could improve accessibility for players with physical limitations who struggle with manual tuning, while globally, by open-sourcing our design documentation, further innovation is enabled.

## 1.5 Visual Overview

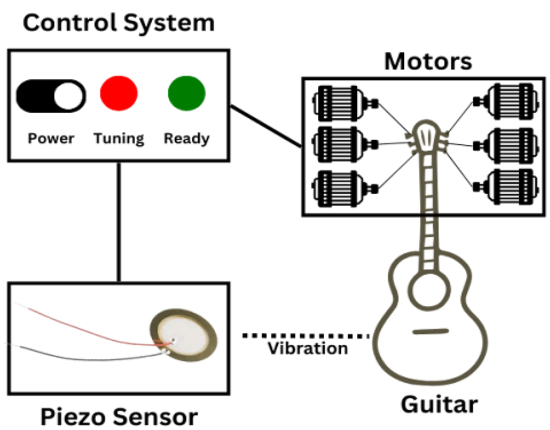



*Figure 1: High-Level Visual Overview of the Automatic Guitar Tuner System*

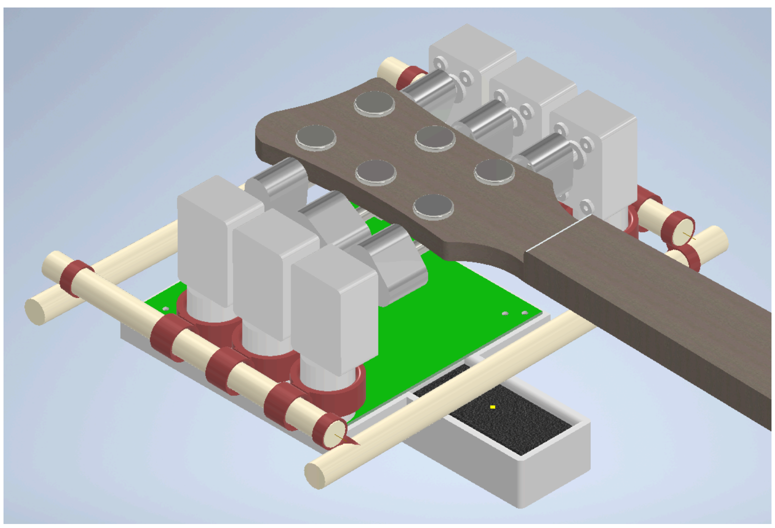


*Figure 2: Motor Mount with Control System and PCB*

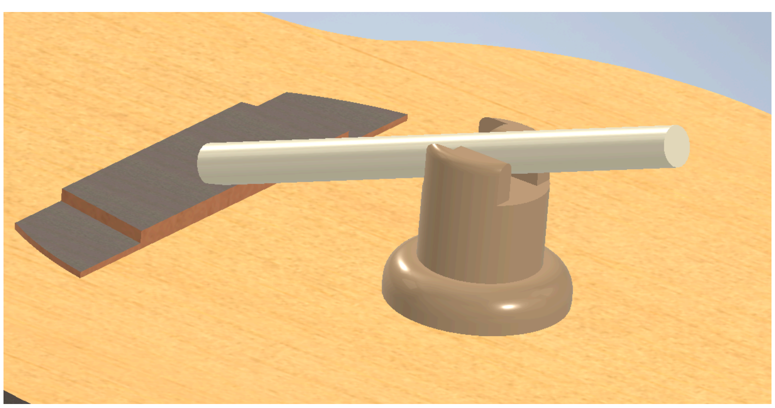


*Figure 3: Piezo Sensor Mounted at Bridge of Guitar with Magnets*

# 2. Design

## 2.1 Block Diagram

The block diagram shown in Figure 4 has gone through minor iterations throughout the project. The main changes are to remove the amplifier in the sensing subsystem, as the processing subsystem was able to accurately detect the guitar frequencies without it and it added additional noise, and the change from GA12-N20 motors to Turbo Worm Gear motors for higher torque.

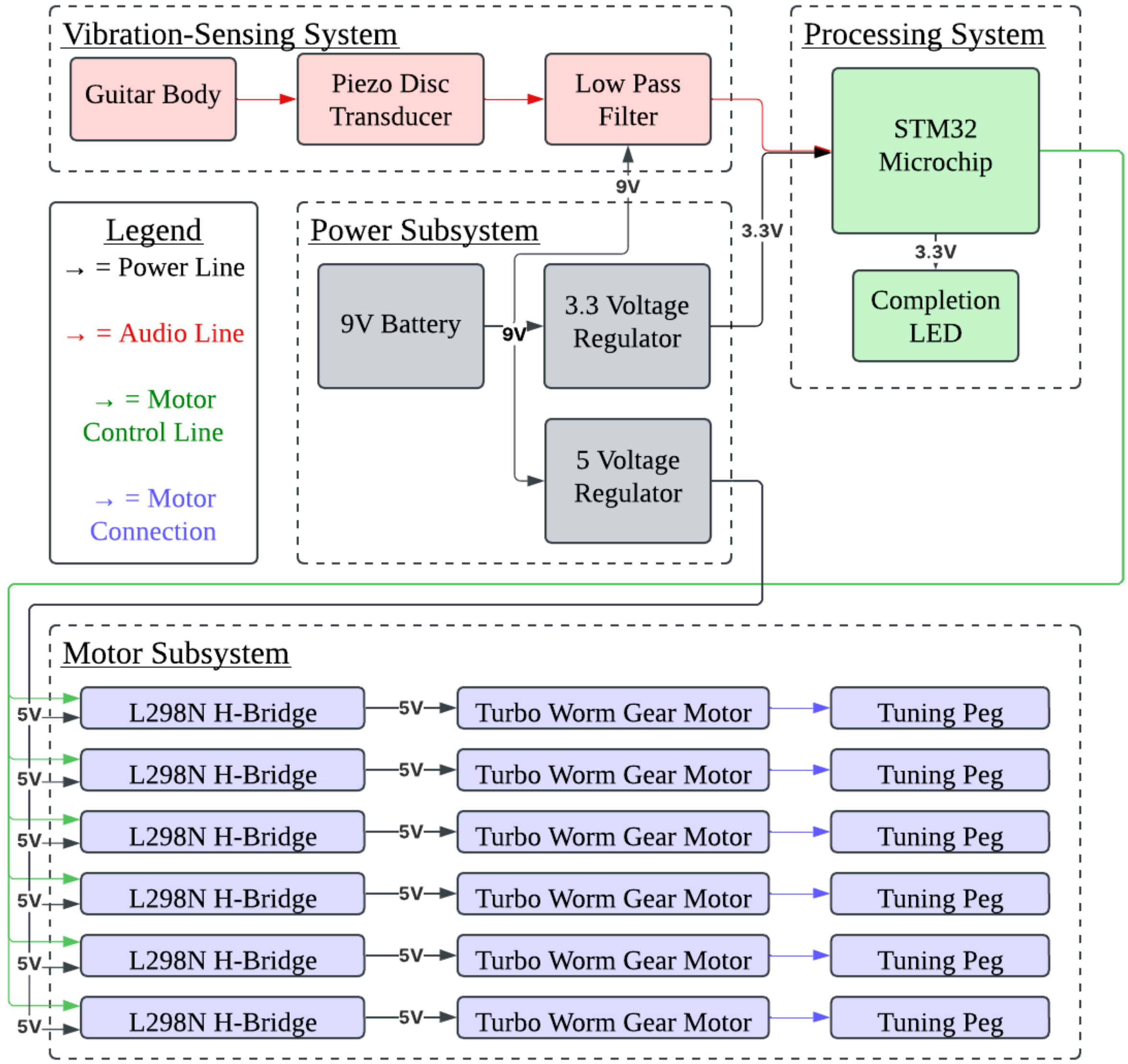



*Figure 4: Block Diagram of the Automatic Guitar Tuner System*

## 2.2 Subsystem Overview

### 2.2.1 Subsystem 1: Power Subsystem

The power system will provide power for the motors and processing system. As the design will be portable, it will be run from a 9V battery (233) and require step down voltage regulators (LM2937 and µA7805) to get the power to 5V and 3.3V for our motor and processing systems respectively.

### 2.2.2 Subsystem 2: Motor Subsystem

The motor system will be responsible for turning the tuning pegs based on the processing system output. There will be 6 motors (Turbo Worm Gear Motor), one for each tuning peg, and they will be driven by H-bridges (L298N) on the PCB. They will also have limited torque and power in order to ensure the system will not damage the guitar.

### 2.2.3 Subsystem 3: Vibration-Sensing Subsystem

This system will take input from a piezo-electric disk transducer (TXJ-055-US) which will read the vibrational frequency from the guitar bridge. This system may also take input from multiple transducers placed at multiple locations on the guitar and combine them for a more accurate and reliable input. This signal will then be put through a low pass filter with a cutoff frequency of ~1000 Hz, and a Zener diode to clamp the voltage to 5V.

### 2.2.4 Subsystem 4: Processing Subsystem

The processing system is the heart of the project, as it will take input from the vibration system, distinguish between all six strings, process which direction to tune each string, and finally send power to the motor system to tune the guitar. We will utilize the STM32H7 microcontroller to run Fast Fourier Transform (FFT) techniques on the signal from the vibration system in order to identify the six strings' individual frequencies. This will be done by having predetermined ranges for each string and sweeping through that range for the peak. Once the frequencies are separated, we will utilize a tuning algorithm to determine the direction the motors need to tune the guitar.

## 2.3 Tuning Algorithm

### 2.3.1 Flowchart Overview

Looking closer at the tuning algorithm, it can be broken down into three subtasks: transforming the output from the Vibration-Sensing Subsystem into a frequency spectrum, determining the frequency of each string from the given frequency spectrum, and finally calculating how much each motor should turn based on a given string's frequency. A visual pseudocode flowchart is shown below to help visualize the individual subtasks and how they logically interact with each other.

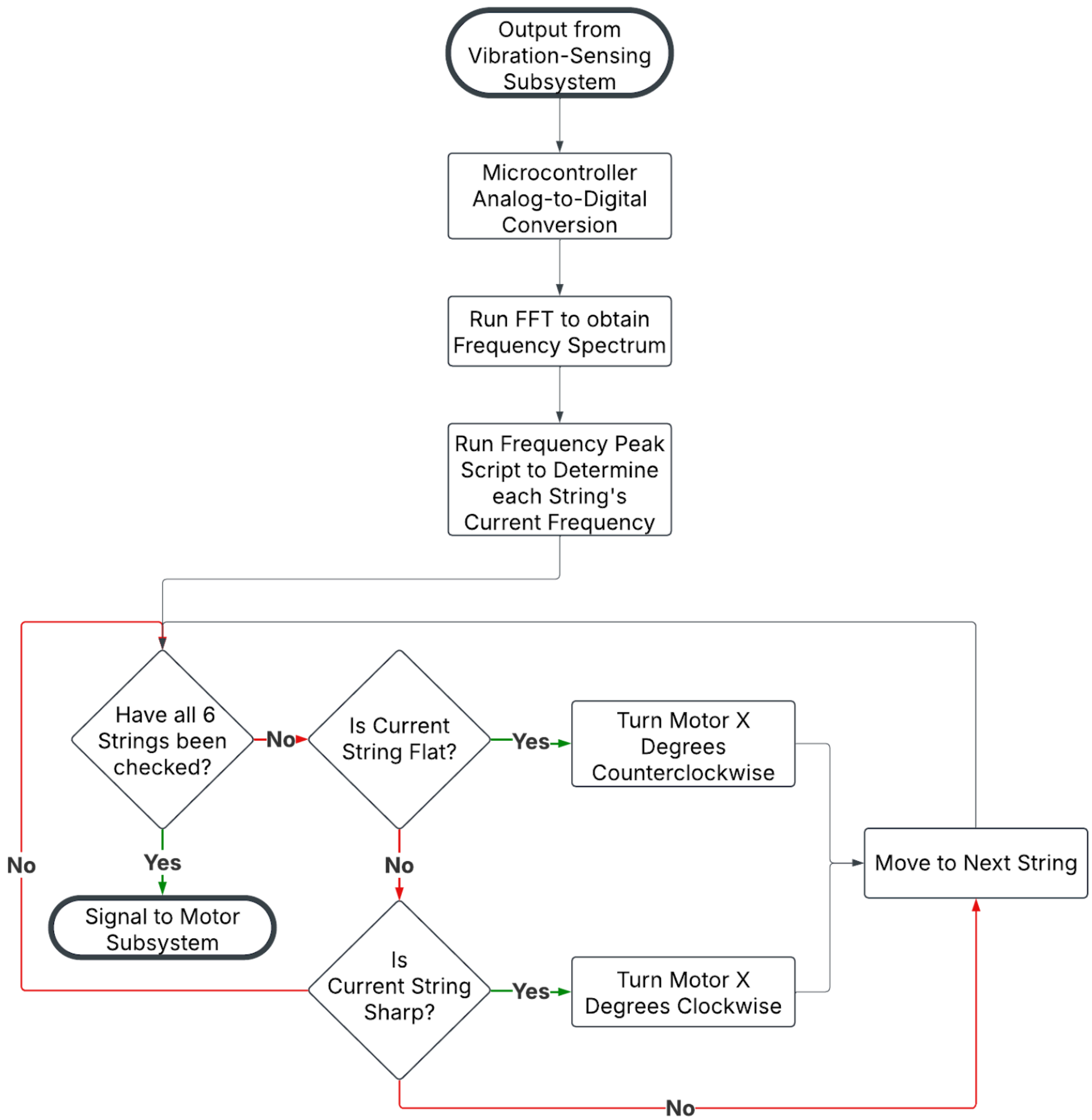



*Figure 5: Tuning Algorithm High-Level Logical Flowchart*

### 2.3.2 Obtaining the Frequency Spectrum

The first subtask is obtaining a frequency spectrum from the Vibration-Sensing Subsystem output. Once the guitar has been strummed, the piezoelectric sensor will output a voltage signal over time which can be read in and converted to a digital signal via the microcontroller's built in 16 bit ADC. A FFT is then taken with a sampling rate of 1024 Hz and a size of 1024 samples to obtain a high-resolution frequency spectrum of the guitar strum as seen below in Figure 6. It is important to note that the project is able to sample at a lower rate as it only needs to capture the standard guitar tuning frequency range of 75-370 Hz (with padding on each end to account for out-of-tune boundaries). Thus, a sampling rate of 1024 Hz would have a Nyquist frequency of 512 Hz which still captures the full guitar tuning range. In addition, Equation (1) is utilized to determine that the frequency resolution of the FFT is 1 Hz. Using these values, the frequency spectrum of the guitar strum can be precisely captured with a resolution of 1 Hz for accurate tuning while keeping computation costs low through a lower sampling rate and size.

$$\Delta f = \frac{f_s}{N} \tag{1}$$

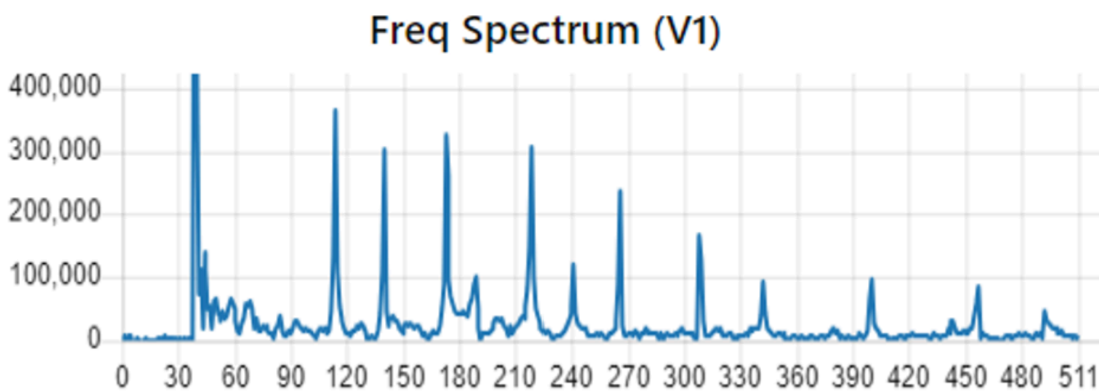



*Figure 6: Example FFT Frequency Spectrum Output for Out-of-Tune Strum*

### 2.3.3 Determining String Frequencies

After a frequency spectrum has been generated, the next subtask is to assign the correct fundamental frequency to each string. In order to accomplish this, the raw frequency spectrum is first filtered to locate all frequency peaks in the range of 75 Hz-370 Hz over a given amplitude. The list of frequency values is then further filtered via a harmonic reduction algorithm to remove all harmonic content, leaving just each strings' fundamental frequency. This is possible due to the mathematical nature of harmonics being integer multiples of the fundamental frequency. For example, the note E3 has a fundamental frequency of 440 Hz with 2nd, 3rd, and 4th harmonic frequencies at 880 Hz, 1320 Hz, and 1760 Hz respectively.

Once all harmonic frequencies are filtered out, the remaining list of frequencies is split into specific note bins to determine the possible corresponding string for each remaining frequency. Due to the nature of how each string is spaced, the algorithm detects out-of-tune notes up to one whole step away in each direction. Thus, the boundary frequencies of each bin reflects that limitation such that the E2 string accepts frequencies between D2 and F#2, the A2 string accepts frequencies between G2 and B2, the D3 string accepts frequencies between C3 and E3, the G3 string accepts frequencies between F3 and A3, the B3 string accepts frequencies between A3 and C#3, and finally the E4 string accepts frequencies between D4 and F#4. As one can see, there is an overlap between the G3 string and the B3 string buckets which is where the limit of one whole step out-of-tune in either direction originated from.

After all frequencies have been sorted into their corresponding note bins, a last check is made to ensure that only one frequency is assigned to each note. This is accomplished by calculating the frequency distance for each frequency in a note bin to the correctly-tuned frequency and keeping the frequency with the smallest distance calculation. As a result, any additional noise in the system is filtered out, leaving only a singular corresponding frequency for each string.

### 2.3.4 Frequency-to-Motor Control Loop

Once each string has an assigned frequency, the final subtask is to determine the motor control based on whether a string is flat or sharp. As shown in Figure 5, it will iterate through each string to check its current frequency against the correct in-tune frequency. After the string has been determined to be either flat or sharp, it then calculates which direction the motor should turn as well as how long the motor should turn.

Looking first at the directional control, the motor should turn counterclockwise if the string is flat and turn clockwise if the string is sharp. In order to determine the duration of how long to run the motor for, a simple calculation based on the rotational speed of the motor is done. Since the motors run at 0.5 RPM (3 degrees of rotation a second), and a given string can be out of tune by at most a whole step, which corresponds to about 180 degrees of rotation, a ratio can be set up such that the out-of-tune frequency difference maps to a degree of rotation within the 180 degree limitation. Finally, this degree of rotation is converted back into time via Equation (2) which represents how long the motor should turn for to achieve the calculated degree of rotation.

$$\textit{Duration of Rotation (s)} = (\textit{Degree of Rotation} / 3) * 1000 \tag{2}$$

## 2.4 Physical Design

### 2.4.1 PCB Design

The PCB design is critical in incorporating and integrating all subsystems of our project. Starting from the power subsystem, a voltage regulator analysis was done on the chosen linear regulators to ensure they operate within proper thermal thresholds. Currently, the 5V regulator's maximum temperature is 108 ℃, and the 3.3V regulator is 72 ℃, which is far below their respective maximum operating temperatures. These linear regulators run off an external 9V battery, and are placed at the edge of the PCB to minimize noise and voltage drop. Test points are also added for each regulator to allow for easier monitoring. The 3.3V regulator feeds the MCU and programming header, which is placed relatively central in the design. The 5V regulator feeds the motor drivers and the signal amplifier, which is routed along the edges of the PCB. This separation allowed for a cleaner design experience. The layout of all subsystems and the full PCB can be seen in Appendix A.

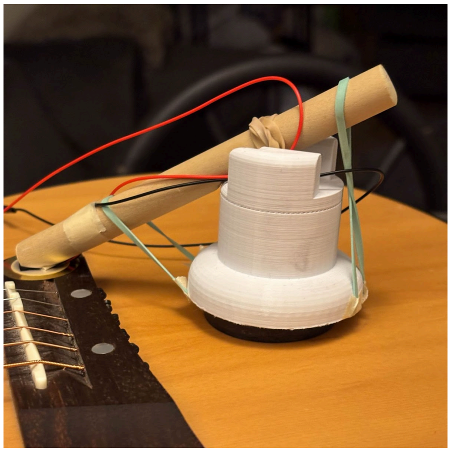
The design of the vibration subsystem is a crucial component of this project. Using the class resources, the STM32 example board, and the recommended Phil's Lab videos [3][4], some best practices when designing for mixed signal applications were found. As such, the analog ground was connected between the MCU and the vibration subsystem, and the physical placement of the whole subsystem was distinct from the digital signals. Back board connections and vias were avoided for the analog signal, as they would add unwanted inductance and capacitance to the signal. A low pass filter with a cutoff frequency around 1kHz was then used in order to filter out any high frequency noise, seen in Equation (3), and test points were added to monitor the unfiltered and filtered signals from the sensor.

The motor subsystem is fairly simple, with three dual H-bridges controlling all six motors. Since the PCB is going to be placed behind the headstock of the guitar, the outgoing motor connectors were placed near the edges of the board to facilitate easier connections. The L298N (motor driver) datasheet states that the supply voltage must be 2.5V higher than the Input High Voltage used to control the driver [5]. As such 9V was routed from the battery for the supply voltage and the 5V from the regulator for the logic supply voltage.

$$F_C = \frac{1}{2\pi RC} = \frac{1}{2\pi(10000\ Ohms)(16\ nF)} \approx 994.72\ Hz \tag{3}$$

### 2.4.2 Mechanical Chassis Design

The mechanical chassis design was primarily created using 3D printed materials and ½" dowel rods. The major design goals of the mechanical chassis design were twofold: to fulfill the high level requirement of a two minute attachment and removal time, and to be able to adjust the motor placement to allow for compatibility with different guitar headstocks. These goals were met through utilizing a magnetically attached sensor mount, a clamshell design for intuitive attachment, and motor mounts that can move along the ½" dowel rods, which can be seen in Figures 7 and 8.



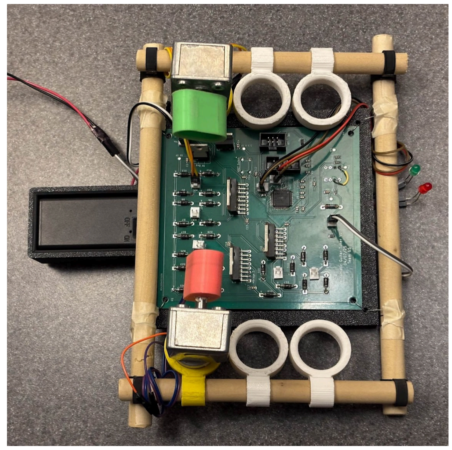


*Figure 7: Sensor Mount* *Figure 8: Tuning Assembly*

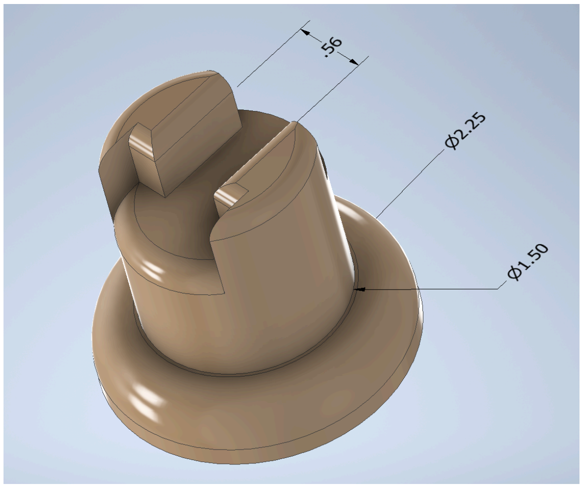
Initially, the design for the sensor mount presented many challenges. The mount not only had to be easily attached and removed, but also hold the sensor solidly to allow for reliable readings. Through consultations with mechanical engineering students, the magnetic mounting system was theorized and implemented, seen in Figure 9. This allowed for quick attachment with minimal noise.

The next component attaches the motor’s D-shaft to the guitar’s tuning peg. This piece needed to fit snugly on the motor shaft while also accommodating the guitar’s tuning peg. Through consulting tuning peg winders this piece was able to fit industry standards and can be seen in Figure 10.

Figure 11 shows the PCB mount, which accommodates the battery housing and is compatible with both the third and fourth wave of PCB boards ordered through the course. This flexibility was due to uncertainty in the timing of the PCB orders, due to customs and shipping issues. The mount ensures that the PCB is oriented the correct way such that the 9V terminal on the board is closest to the battery housing and the motor connectors are close to the motors themselves. The housing is critical in ensuring the dowel rods do not flex too much during operation, and it provides convenient cable management channels.

Finally, Figures 12 and 13 show the different connectors used between the dowel rods and between the dowel rod and motors. These parts are critical to the adjustability factor mentioned previously, allowing the system to become wider according to the guitar’s headstock, and the spacing between motors to change according to the guitar’s tuning peg spacing.

All the components can be seen in the assembly in Figures 2 and 3, and assembled in physical form in Figures 6 and 7.



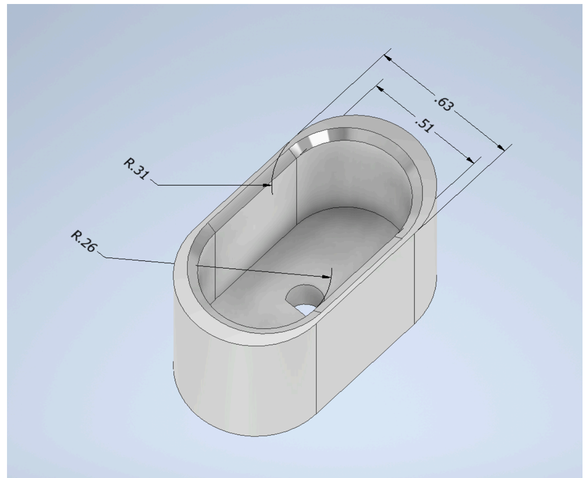




*Figure 9: Sensor Mount*

*Figure 10: Motor to Peg Connector*

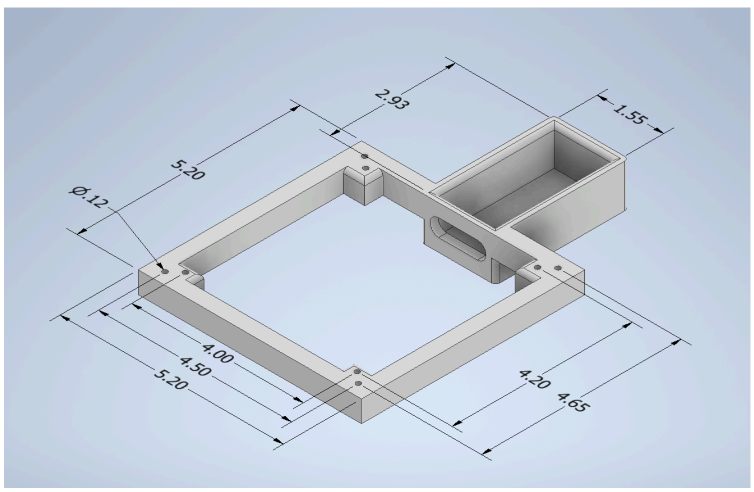
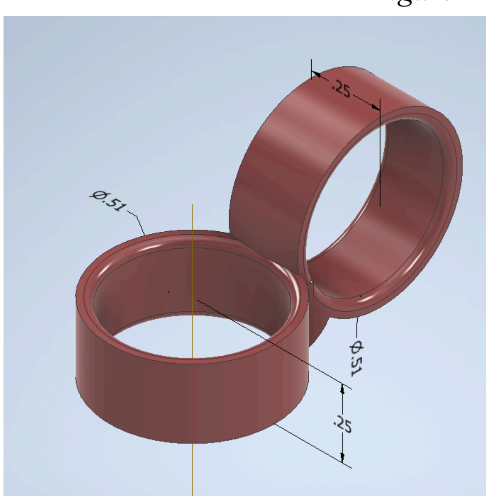



*Figure 11: PCB Mount*



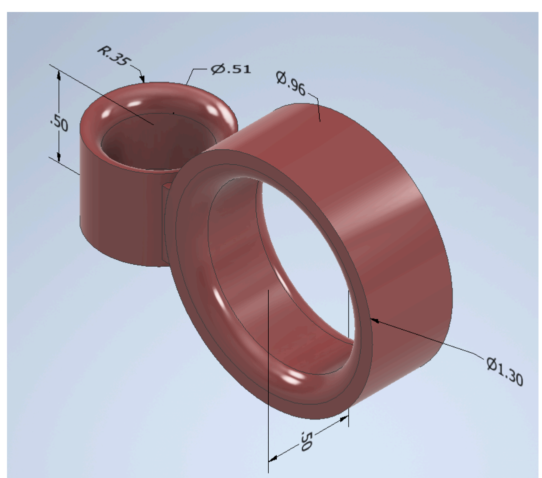




*Figure 12: Dowel to Dowel Connector*

*Figure 13: Motor to Dowel Connector*

# 3. Design Verification

The design verification section details how each subsystem was individually tested and the success or failure of certain requirements that were established. The full requirements and verification table can be found in Appendix B.

## 3.1 Power Subsystem

### 3.1.1 Stable Power Supply

The first requirement was that the power subsystem must be able to supply a continuous 5V±0.25V and 3.3V±0.2V for all system components. After a 60 second test was performed the minimum and maximum voltages for the 5V and 3.3V converters were 4.928V, 4.944V and 3.241V, 3.257V

### 3.1.2 Load Test

The second requirement ensured that output voltage did not decrease by more than 5% when motors are under load. To test this, a 60 second test was performed and the minimum and maximum voltages for the 5V converter were 4.877V, and 4.944V. This test was done with four GA12-N20 placeholder motors and two high torque worm gear motors.

## 3.2 Motor Subsystem

### 3.2.1 Motor Timing

The first requirement ensures the motor can adjust the pitch of the guitar within 15 seconds per strum of the guitar. This is to ensure that the overall 1 minute tuning time constraint is met. Through testing we verified the 0.5 RPM speed of the motors and found that the motors started affecting the pitch of the guitar within 3 seconds of enabling. This lag time is due to the inherent space in the connector between the motor and tuning peg to accommodate differently sized tuning pegs on different guitars. This lag time is only present once in the tuning process, allowing for adequate time overall for the pitch adjustment. This is due to the out-of-tune restrictions of one whole step in each direction. Though slightly different for each string, a 180 degree rotation is sufficient to bring the guitar in tune.

### 3.2.2 Torque Limits

The torque requirement for the motor is $\leq 0.5$ N·m to prevent damage to the guitar. When tested experimentally, the motors at full duty cycle and 5V produced 0.42 N·m of torque, which meets this limit. Additionally, in order to ensure safety of all equipment, software limits are put into place through using preset motor timings and out-of-tune limits. This ensures that the motor will never overtension a string, resulting in damage to the guitar, string, or user.

## 3.3 Vibration-Sensing Subsystem

### 3.3.1 Sensitivity

The main requirement for the vibration-sensing subsystem is for the piezoelectric sensor to be able to sense the vibrational frequency of the guitar within ±12 cents of the true frequency. In order to verify this requirement, the sensor output was fed into the ADALM 2000's spectrum analyzer and then attached to the guitar bridge to best pick up the vibrations. The table below shows the experimental results from this test with the sensor accurately detecting the frequency of all 6 strings within ±12 cents of the original frequency.

| Guitar String Note | Exact Tuning Frequency | Tuning Frequencies ±12 cents | Obtained Tuning Frequencies |
|---|---|---|---|
| E2 | 82.41 Hz | 81.84 Hz - 82.98 Hz | 82.8125 Hz |
| A2 | 110 Hz | 109.23 Hz - 110.77 Hz | 109.375 Hz |
| D3 | 146.83 Hz | 145.81 Hz - 147.85 Hz | 146.875 Hz |
| G3 | 196 Hz | 194.65 Hz - 197.36 Hz | 195.313 Hz |
| B3 | 246.94 Hz | 245.22 Hz - 248.66 Hz | 246.875 Hz |
| E4 | 329.63 Hz | 327.34 Hz - 331.92 Hz | 331.25 Hz |

*Table 1: Vibration-Sensing Experimental Frequency Results*

### 3.3.2 Signal to Noise Ratio

The second requirement for the sensing subsystem is to ensure that the Signal to Noise Ratio (SNR) is above 20dB in an indoor environment. The SNR can be found using Equation (4).

$$SNR\ (dB) = 20\ log_{10}[A_{signal}/A_{noise}] \quad (4)$$

This ratio can be affected by the mechanical connection between the sensor and its mount, or between the sensor and the guitar's bridge, or the low pass filter cutoff frequency. Through testing each string individually and all six strings at once, the minimum SNR was found to be 20.4 dB over three trials.

| | Trail 1 | Trial 2 | Trial 3 |
|---|---|---|---|
| **Minimum SNR (dB)** | 21.0 | 20.4 | 20.7 |

*Table 2: Minimum SNR over Three Trials*

## 3.4 Processing Subsystem

### 3.4.1 Multiple String Detection

The first requirement for our processing subsystem was the ability to detect multiple strings frequencies when strummed at once. To verify this, a baseline test case was run where two of the guitar strings were strummed at the same time. We can see from Figure 14 (shown in Hz) that the it is able to correctly detect both strings' frequencies when simultaneously strummed.

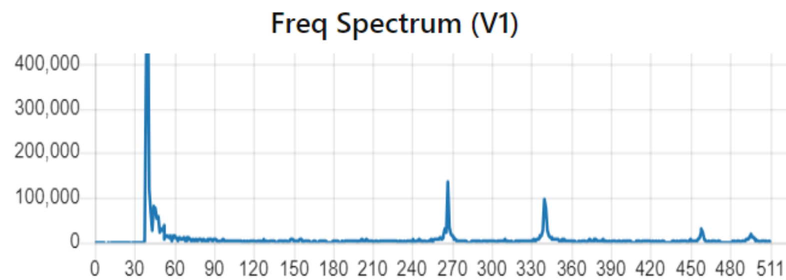



*Figure 14: Frequency Spectrum Output of Two Strings Strummed at Once*

### 3.4.2 Processing Accuracy

The second requirement is that the processing subsystem must analyze frequencies with at least ±3 Hz of precision in the 80 Hz-350 Hz tuning frequency range. This ensures that frequency detection precision is not lost after taking the FFT via the microcontroller. In order to verify this, the sensor is attached to the guitar and all 6 strings are strummed at once. The output from the sensor is then fed to the microcontroller where it detects each string's frequency via the tuning algorithm process described in section 2.3 Tuning Algorithm. Thus, Table 2 shows that it both meets the requirement of ±3 Hz of precision and also fulfills the ±12 cents of precision as well.

| Guitar String Note | Exact Tuning Frequency | Tuning Frequencies ±3 Hz | Obtained Tuning Frequencies |
|---|---|---|---|
| E2 | 82.41 Hz | 79.41 Hz - 85.41 Hz | 82 Hz |
| A2 | 110 Hz | 107 Hz - 113 Hz | 110 Hz |
| D3 | 146.83 Hz | 143.83 Hz - 149.83 Hz | 147 Hz |
| G3 | 196 Hz | 193 Hz - 199 Hz | 195 Hz |
| B3 | 246.94 Hz | 243.94 Hz - 249.94 | 248 Hz |
| E4 | 329.63 Hz | 326.63 Hz - 332.63 Hz | 329 Hz |

*Table 3: Processing Accuracy Experimental Frequency Results*

# 4. Costs

## 4.1 Parts

| Part Description | Part Number | Vendor | Quantity | Total Cost |
|---|---|---|---|---|
| 3.3V Voltage Regulator | LM2937-3.3 | DigiKey | 1 | $1.81 |
| 5V Voltage Regulator | µA7805 | DigiKey | 1 | $1.16 |
| H-Bridge | L298N | E-shop | 3 | - |
| Motors | Turbo Worm Motor | Amazon | 6 | $44.94 |
| Microcontroller | STM32H7B0RBT6 | DigiKey | 1 | $9.31 |
| Piezo Disc Transducer | TXJ-055-US | Amazon | 15 | $6.99 |
| Guitar | First Act Guitar 222 | Facebook | 1 | $55 |
| 9V Battery System | Energizer MAX | Amazon | 4 | $17.87 |
| | | | | **Total: $137.08** |

*Table 4: Parts/Materials Cost Breakdown*

## 4.2 Labor

| Engineer | Circuit Design | Soldering | Tuning Algorithm | Prototype and Testing | Documentation and Write-Up | Total Hours |
|---|---|---|---|---|---|---|
| Ethan Lin | 15 | 15 | 30 | 50 | 30 | 140 |
| Nathan Kim | 10 | 10 | 40 | 50 | 30 | 140 |

*Table 5: Labor Breakdown by Hour*

## 4.3 Total Cost

Given that an average ECE graduate makes a starting yearly salary of $98,472.50 [6], we can use an estimated hourly rate of $47.34 to calculate the labor costs of this project.

| Overall Total Hours | Hourly Rate | Overhead Multiplier | Total Cost |
|---|---|---|---|
| 280 | $47.34 | 2.5x | **$33,275.08** |

*Table 6: Total Cost Breakdown*

# 5. Conclusion

## 5.1 Accomplishments

Our project can successfully fulfill all the high level requirements, and is almost capable of performing its intended function in a real-world setting, delivering a fully automated guitar tuning system that is both precise and user-friendly. The system currently accurately detects string frequencies via the piezoelectric sensor, processes them in real time using the STM32's FFT algorithm, and sends the correct control signals to drive six independent motors to adjust tuning pegs to within ±12 cents of the target pitch, meeting professional tuning standards. The entire process completes in under one minute, fulfilling our speed criterion while ensuring no damage to the guitar through torque-limited motor control.

## 5.2 Uncertainties

A key challenge this project faced was the inability to establish communication between the microcontroller and H-bridge motor drivers, despite extensive testing of current draw, pin connections, and PCB layout. As a workaround, we validated motor control functionality using the ADALM Scopy. For the rest of the project we utilized a development board with a similar STM32H7 microcontroller. While this approach confirmed our tuning algorithm's functionality and feasibility, the slight differences between the development board's chip and our specific STM32H7B0 model may require additional firmware adjustments. These unresolved integration issues point to potential areas for future investigation, including more detailed signal analysis or driver firmware modifications.

## 5.3 Ethical Considerations

One potential ethical or safety issue that would arise from this project would be potential harm to people's property [7]. Automatic guitar tuners are not a new idea, but consumers are generally skeptical about them due to their history of damaging the guitars they tune. In order to prevent this, our design limits the power and torque the motors can produce, removing the possibility of damage. Another possible issue would be to respect the work required to produce new ideas [8]. Previous ECE 445 groups have created automatic guitar tuners, and our work aims to build on their designs. To prevent issues, we will clearly document our ideation and creation process to clarify our sources and references.

## 5.4 Future Work

Future work for this project includes full PCB integration, wireless communication between the sensor and processing subsystems, miniaturization, and improving the user feedback system. Additionally, functionality to tune to alternate tunings like drop-D and open-G can be explored.

# Appendix A: PCB Layout and Board

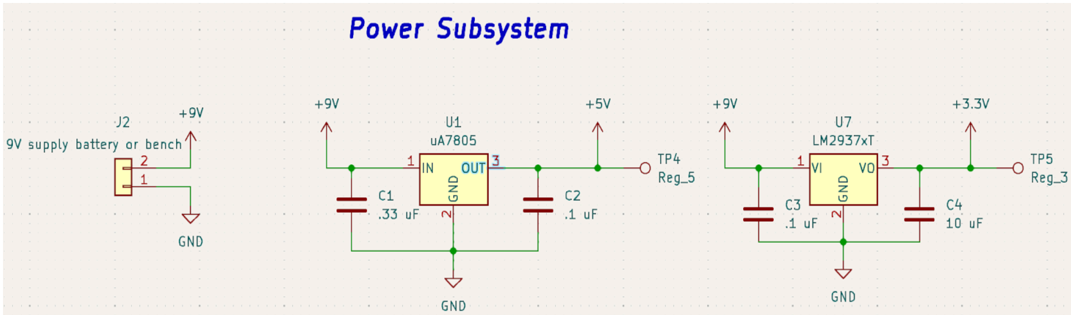



*Figure 14: Power Subsystem Layout*

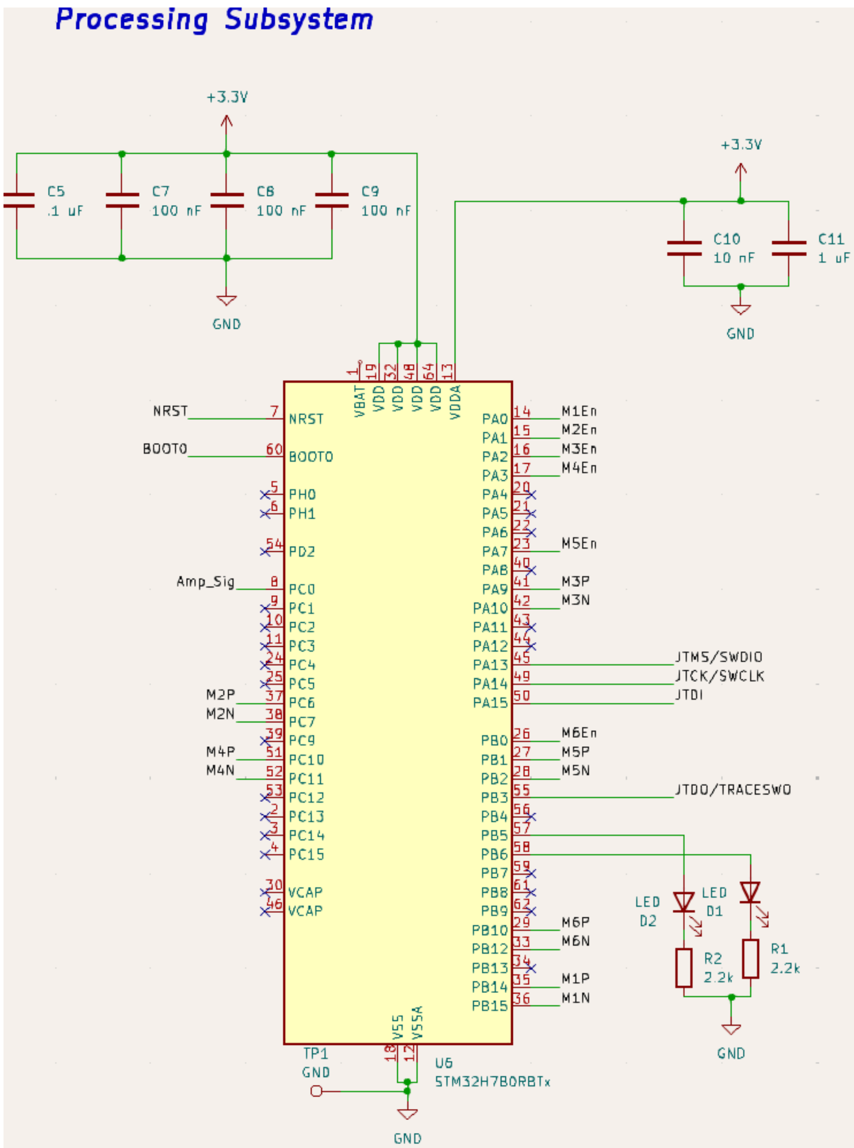



*Figure 15: Processing Subsystem Layout*

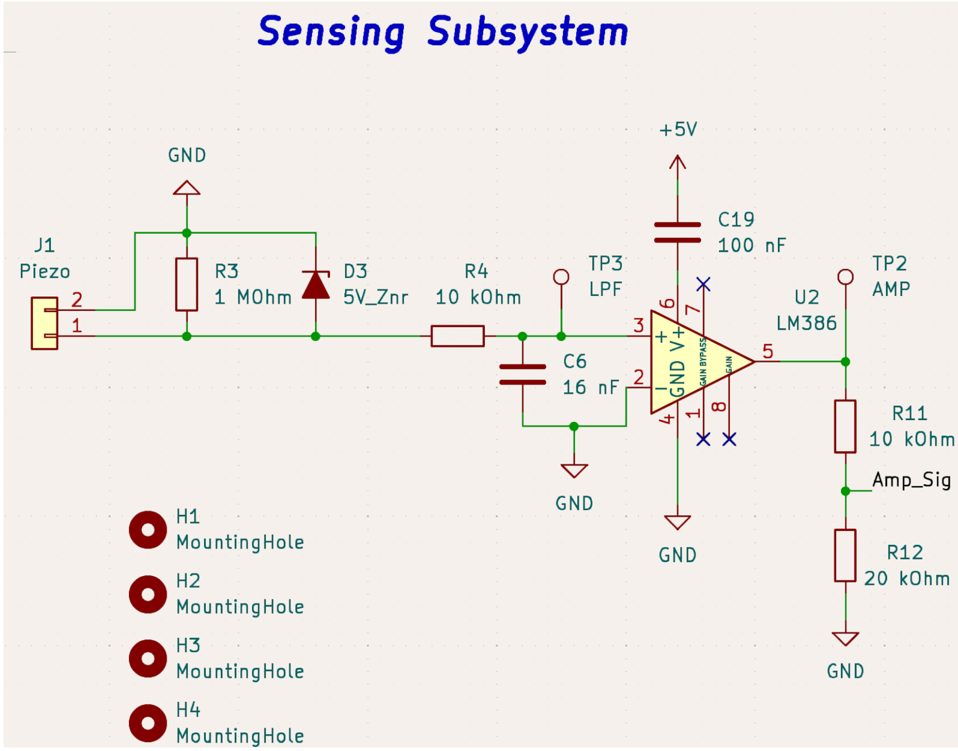



*Figure 16: Sensing Subsystem Layout*

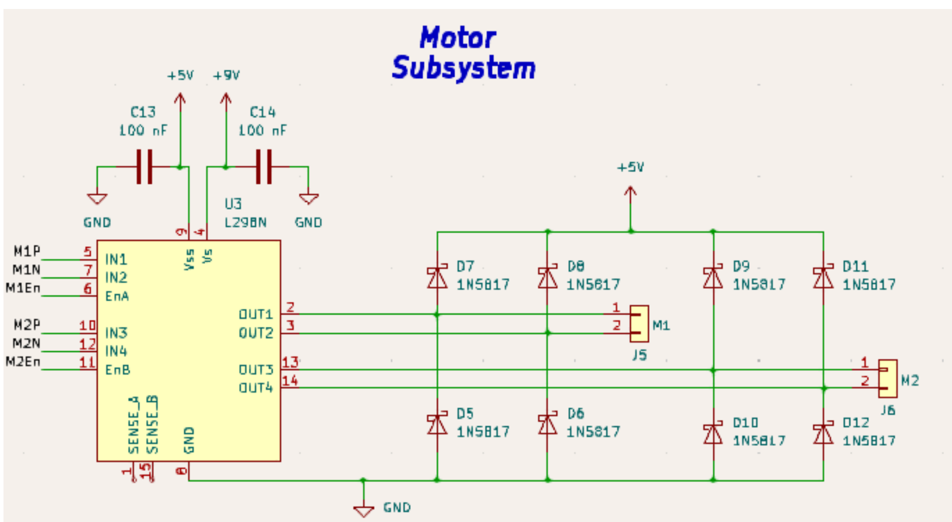



*Figure 17: Motor Subsystem Layout*

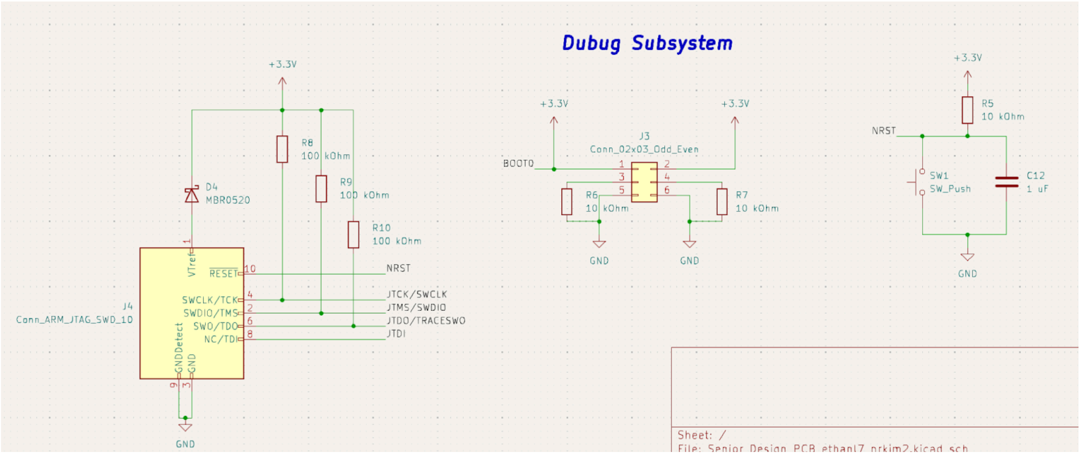


Figure 18: Debug Subsystem Layout

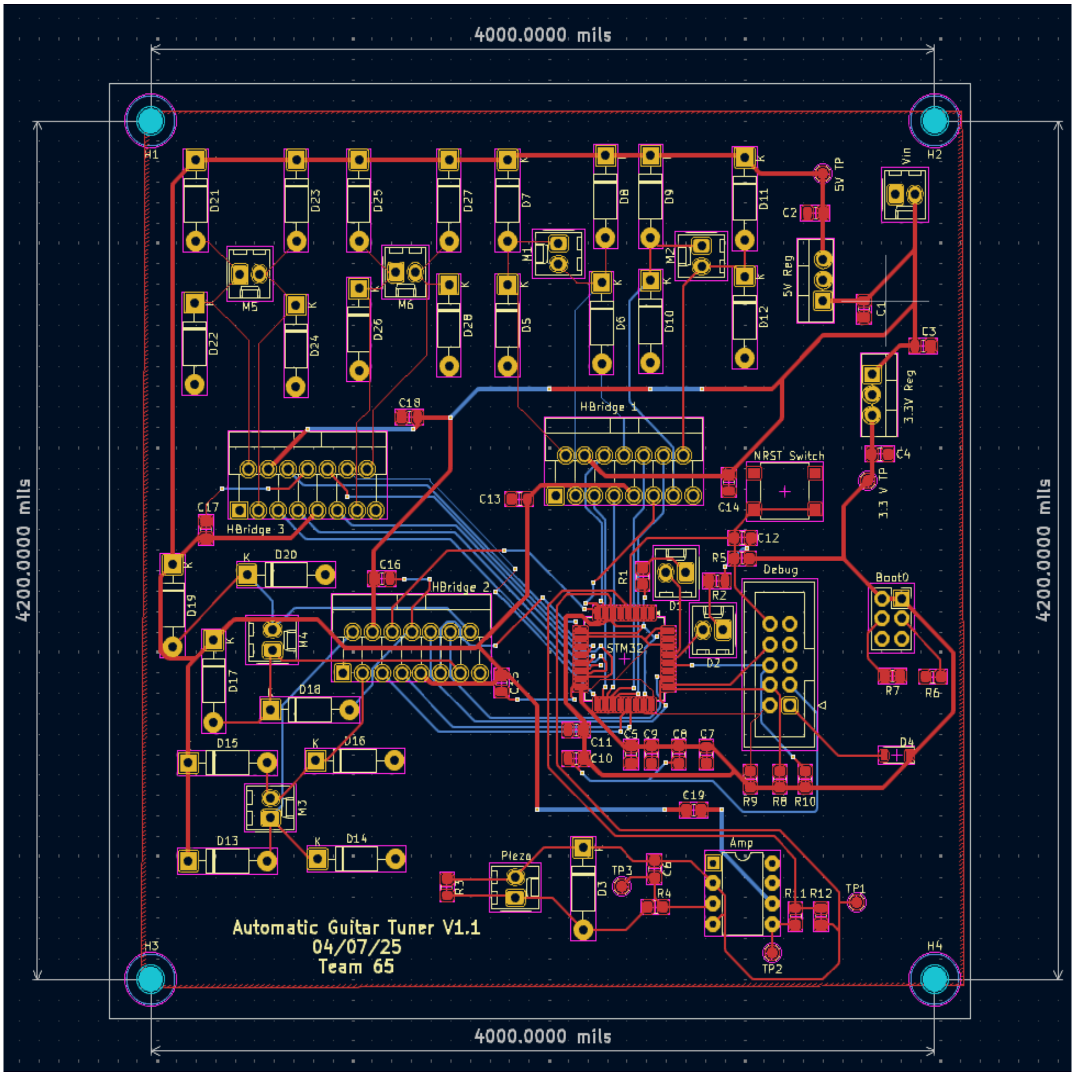


Figure 19: PCB Layout

# Appendix B: Requirements and Verification Tables

## Power System Requirements and Verifications

| Requirements | Verifications | Verification Status (Y/N) |
|---|---|---|
| Must be able to supply a continuous 5V±0.25V and 3.3V±0.2V for all system components | 1. Connect 9V battery to battery clip and ensure power switch is turned on<br>2. Connect jumper wire to ground, and another to either the 3.3V or 5V power rail<br>3. Measure voltage with a voltage sensor and verify that it is within 3.1V-3.5V and 4.75V-5.25V respectively | Y |
| Output voltage should not decrease by more than 5% when motors are under load | 1. Connect 9V battery to battery clip and ensure power switch is turned on<br>2. Connect jumper wire to ground, and another to the 5V power rail<br>3. Attach the piezoelectric sensor to the guitar without connecting the motors to the tuning pegs<br>4. Ensure at least one string is out of tune on the guitar, then strum the guitar<br>5. Verify the motors are on, measure the voltage, and verify it is above 4.5125V | Y |

## Motor System Requirements and Verifications

| Requirements | Verifications | Verification Status (Y/N) |
|---|---|---|
| The motor system must be able to adjust pitch within 15 seconds per strum to meet the overall tuning time requirement | 1. Assemble automatic guitar tuner, with all subsystems connected and power on<br>2. Ensure at least one string is out of tune on the guitar, then use an external tuner to note the pitch of each string<br>3. Strum the guitar and start a timer for 15 seconds<br>4. At 15 seconds, or when tuning completes, turn off the automatic guitar tuner and note the pitch of each string again<br>5. Verify all strings have adjusted pitch by at least 1 Hz. | Y |
| The motor system must rotate tuning pegs with a torque limit of $\leq 0.5$ N·m to prevent damage to the guitar | 1. Attach arm of known length to motor<br>2. Power motor from H-bridge at full operating conditions<br>3. Use a scale to measure the force at the end of the arm<br>4. Calculate the torque using the measured values and verify that it is within the stated limit | Y |

## Vibration-Sensing System Requirements and Verifications

| Requirements | Verifications | Verification Status (Y/N) |
|---|---|---|
| Must be able sense to vibrational frequency of the guitar and accurately output a signal within ±12 cents of the original frequency | 1. Attach the piezo disc transducer to the guitar bridge<br>2. Connect an oscilloscope to the output of the sensor<br>3. Strum the low E string and measure the outputted frequency from the transducer<br>4. Use an external tuner to also measure the frequency of the low E string<br>5. Repeat for all six strings<br>6. Verify that the outputted frequency from the tuner is within ±12 cents of the frequency measured by the external tuner | Y |
| The subsystem must reject ambient vibrations by maintaining a signal-to-noise ratio (SNR) ≥20 dB for string frequencies (75–370 Hz) when tested in a typical indoor environment | 1. Attach the piezo disc transducer to the guitar bridge<br>2. Play recording of typical indoor environment’s noise (foot taps, room noise)<br>3. Strum the low E string and measure the outputted frequency from the transducer<br>4. Measure the peak amplitude of the target frequency and the average noise floor amplitude in adjacent frequency bins<br>5. Verify that the SNR is at least 20 dB for all six strings | Y |

## Processing System Requirements and Verifications

| Requirements | Verifications | Verification Status (Y/N) |
|---|---|---|
| The processing system is able to detect multiple string frequencies at once | 1. Assemble automatic guitar tuner, with all subsystems connected and power on<br>2. Simultaneously strum two strings of the guitar<br>3. Obtain a frequency spectrum via FFT from the microcontroller<br>4. Verify that the two frequency peaks are distinct | Y |
| The processing system must analyze frequencies with at least ±3 Hz precision in the 75 Hz – 370 Hz range | 1. Assemble automatic guitar tuner, with all subsystems connected and power on<br>2. Strum all six strings of the guitar and obtain the 6 transformed frequency outputs from the microcontroller (one for each string)<br>3. Verify that the frequency output for each string is within ±3 Hz of the standard guitar tuning (82.41 Hz - E, 110 Hz - A, 146.83 Hz - D, 196 Hz - G, 246.94 Hz - B, 329.63 Hz - E) | Y |

## References

[1] “TronicalTune PLUS - Guitar Tuner AI full automatic for almost any Guitar,” Guitar tuner professional. full automatic guitar tuner that fits almost any guitar like Fender, Gibson, Ibanez Yamaha, Oct. 14, 2024. https://www.tronicaltune.net/tronicaltune-plus/

[2] J. M. Geringer and M. D. Worthy, “Effects of Tone-Quality Changes on Intonation and Tone-Quality Ratings of High School and College Instrumentalists,” Journal of Research in Music Education, vol. 47, no. 2, pp. 135–149, Jul. 1999, doi: https://doi.org/10.2307/3345719.

[3] Phil’s Lab, “Mini 6-Layer Mixed-Signal Hardware Design Walkthrough - Phil’s Lab #78,” YouTube, Oct. 08, 2022. https://www.youtube.com/watch?v=AxEL3eESL9I (accessed Apr. 02, 2025).

[4] Phil’s Lab, “Mixed-Signal Hardware/PCB Design Tips - Phil’s Lab #88,” YouTube, Dec. 22, 2022. https://www.youtube.com/watch?v=v6fTa6LRJLI (accessed Apr. 02, 2025).

[5] L298 Dual full-bridge driver, datasheet, STMicroelectronics, 2023. Available at: https://www.st.com/resource/en/datasheet/l298.pdf

[6] G. E. O. of M. and Communications, “Salary Averages,” ece.illinois.edu. https://ece.illinois.edu/admissions/why-ece/salary-averages

[7] “IEEE Code of Ethics,” @IEEEorg, 2019. http://www.ieee.org/about/corporate/governance/p7-8.html

[8] Association for Computing Machinery, “ACM code of ethics and professional conduct,” Association for Computing Machinery, 2018. https://www.acm.org/code-of-ethics