**Q-Tune: Final Report**

Yenessa Avalos-Maldonado, Timothy Chiles, William Trigg, Giovanni Cornejo, and Dakota Cappel

Github Link: <https://github.com/avalosy8/qtune>

**Abstract**

The process of tuning a guitar can be a daunting task for guitarists, especially for beginner guitarists who have yet to develop the skill to tune a guitar by ear. Even experienced guitarists find it challenging to maintain their guitar in tune for extended periods. This project aims to tackle this problem by developing an automatic guitar tuner that can precisely and efficiently tune each string in a guitar. The tuner can analyze the frequency of the guitar string vibrations using a piezo sensor and provide instant adjustments using servo motors attached to each tuning peg of a guitar.

The tuner features a visual display using an LCD that helps guide the user throughout the tuning process by displaying the string selection, its targeted frequency, and overall tuning progress. Each tuning peg is attached to its own servo motor that twists the pegs. To maintain accuracy and reliability, the tuner utilizes a piezoelectric sensor to capture the vibration frequency from the string. Additionally, a rechargeable power bank is used to power the device via a USB type-C cable, which is connected to the Adafruit RP2040 microcontroller. In general, the tuner aims to develop a solution that can ease the process of tuning a guitar. The device is enclosed in a 3D-printed case that can clamp onto the head of the guitar. Ultimately, the project investigates how to enhance the overall guitar playing experience.

**Introduction**

The main goal of this project is to make the process of tuning a guitar easier, more accurate, and automated. Tuning an instrument is a universal struggle among musicians, and for guitarists this is no exception. Depending on how out of tune a guitar is, it can take as long as 15 to 20 minutes to get all strings back in tune. In addition, most guitarists must use a device that reads the note’s frequency and tells the guitarist whether it is still too high or low. While more experienced guitarists can memorize how each string sounds and tune it accurately, it still requires them to turn each peg an appropriate amount.

Our project intends to serve a need for all guitarists regardless of their skill level, which is to tune a guitar much faster by providing an easy-to-understand user interface. This user interface consists of a LCD screen and two buttons, and the latter allows the user to select which string they want to tune by cycling through the strings. Therefore, all the user must do is to attach the servos to the tuning pegs, select which string they want to tune, and strum the guitar until the LCD outputs a completion message. The screen can either tell the user to keep strumming, or that the tuning process is finished, while using a progression bar. The user can then move onto the next string, if needed. This overall improvement to the speed and reliability of the tuning process is something any guitarist will appreciate, and the stated elements above help achieve this improvement.

For this project, we largely focused on cultural, social and economic factors that could influence the development of our device. This includes the difficulty of upscaling production of our device because of its many separate parts, how the speed of setup would impact use among consumers, and the ability of novel guitarists to use the device to tune their guitar, to name a few. Considering our device involves moving servo parts, we would suggest that users refrain from putting their fingers near the tuning pegs as they turn. Otherwise, this project is largely disconnected from matters of public health, safety, or welfare.

**Background**

Analog guitar tuners were invented in 1936 and replaced with digital tuners in the 1980s. An issue that these original microphone-based guitar tuners exhibited was that they were much less effective in loud environments due to background noise. This issue was resolved in 1995 when the first clip-on guitar tuner was released that utilized a contact microphone that senses the physical vibrations of the guitar’s wood rather than the sound waves traveling through the air. Since this breakthrough, developments have been made towards automating the entirety of the tuning process. Automatic guitar tuners currently take two forms: the handheld motorized guitar tuners that the user must hold up to each string one by one (see Figure 1) and the built in guitar tuners that come installed in certain guitars (see Figure 2).



Figure 1: Handheld Automatic Guitar Tuner



Figure 2: Built-in Automatic Guitar Tuner

Guitars produce sound via the oscillations of the strings when the strings are plucked. These oscillations get transferred to the body of the guitar via vibrations. The length, thickness, and tension of each guitar string determine the frequency with which the string will vibrate and, in turn, the note that the string will produce. Before playing, the frequency of each full-length string needs to be set to a reference note. Because the length and thickness of an open guitar string are constant, setting the reference note is achieved by adjusting the tension of the string with the tuning pegs at the head of the guitar. The standard reference notes and frequencies are, in order from top to bottom, E (82.41 Hz), A (110.00 Hz), D (146.83 Hz), G (196.00 Hz), B (246.94 Hz), and E (329.63 Hz). Guitar tuners convert the sound of each open string into an equivalent sinusoidal voltage, determine the sinusoid’s frequency, and compare it to these reference values to determine if the string is in tune.

While each guitar string vibrates primarily with its fundamental frequency, it technically oscillates at a superposition of its fundamental frequency and its harmonics, which are integer multiples of the fundamental frequency. To account for these other frequencies, different techniques could be utilized. One such approach would be to filter out the higher harmonic frequencies via analog or digital filtering techniques. While this approach could be used to filter out many of the harmonic frequencies producible by the guitar strings, it would not filter out the multiples of the lower frequency strings that are still smaller than the higher strings. Because the harmonics are integer multiples of the fundamental, a simple digital approach to account for harmonics would be to check for not only the fundamental frequency being present but to also check for twice the frequency, three times the frequency, and onwards.

As mentioned previously, contact microphones have widely been utilized in clip-on guitar tuners. Contact microphones are implemented using piezoelectric transducers. These sensors produce charge on their nodes that is proportional to the mechanical stress applied to their surfaces. These sensors are sensitive enough to pick up the small vibrations produced at the head of a guitar; however, the charge produced by such vibrations is very small. The signal produced by piezo sensors is often driven through an amplifier circuit to increase its amplitude to an appropriate level for subsequent circuitry.

Amplifiers with high input impedance are often used to match the high source impedance of piezo sensors. While voltage-mode amplifiers are often used, charge amplifiers are more recommended for piezo sensors. While the charge produced by the sensor produces a voltage across the piezo sensor’s two plates, this voltage is not necessarily proportional to the charge due to the ability of the capacitance to change between the two nodes. The capacitance can change because the two nodes are often connected to the circuitry via a cable. Changing this length of cable or even adjusting the cable could alter the capacitance of the cable and corrupt the signal from the piezo. Utilizing a charge amplifier nulls this effect as the charge is forced onto one of the plates of the feedback capacitor in the circuit whose value does not change. Additionally, a charge amplifier’s input resistance forms a high-pass filter with the piezo’s capacitance, and the feedback capacitor and resistor form a low-pass filter. The charge amplifier can thus be used to achieve a bandpass response which is useful in many applications.

When it comes to motors, there are many types of motors that could be used to tune a guitar. Servo motors can offer a large amount of torque while limiting the size and the power needed. Using three wires, ground, power, and data, the servo motors can be controlled through pulses from a microcontroller pin. The data channel for the servo motor is constantly receiving pulses from the controller. When wanting to move the motor with a certain power, the width of this pulse changes to a certain value, indicating the motor to move. This type of communication is called Pulse-width modulation. Servos can come in limited rotation or full 360 degree rotation. Both operate in the same manner, however, the limited rotation only offers about 180 degrees of rotation. This can be useful in some cases, but for a guitar tuner, a full 360 rotation is required to ensure that the tuner can always turn each way no matter where it starts.

**Timeline**

We began the project with some research on how we might develop an automatic guitar tuner. This research led us towards a plan to use a piezoelectric transducer to sense the vibrations of the guitar strings in conjunction with servo motors to turn the motors based on a frequency calculation. To bootstrap the project for the pre-alpha build, we utilized parts we had on hand and began experimenting with calculating frequencies and turning servo motors. We decided that the simplest method would be to convert the sinusoids from a guitar into a square wave and calculate the frequency by timing the period between rising edges. To convert the sinusoids into square waves, we constructed a simple op-amp comparator circuit. Because we had not acquired a piezoelectric sensor yet, we fed sinusoids into the comparator using a waveform generator. We utilized an oscilloscope to verify that the comparator successfully converted the inputted sinusoid into a square wave with the same frequency. During this process, we utilized CircuitPython on our Adafruit RP2040 microcontroller to quickly get a simple micro servo motor to turn clockwise and counterclockwise. Because we planned to use interrupts to time the periods of the square waves from the comparator, we were led to utilize asyncio in CircuitPython to get interrupt-like functionality that could successfully calculate the frequency of our square waves.

When progressing towards our prototype build, we realized that the servo we were using could only turn 90 degrees in both directions which would not suit our needs if our guitar was more than a quarter-turn out of tune. We ordered continuous servo motors that could continuously turn clockwise or counterclockwise and rewrote our code to accommodate these new motors. In addition, we wrote functions that could rotate the servo motors in increments in either direction based on the frequency that another section of our code calculated. We introduced the UI during the prototype phase but did not connect them to the remainder of the tuner. We acquired a piezo sensor during this period and began testing our comparator circuit/firmware with it. We initially played pure tones on our laptop speakers and attempted to pick up the vibrations with our sensor. Testing the piezo with speakers and then a guitar introduced a lot of randomness and inconsistency with our analog signal and thus our frequency we calculated in our firmware. One of the causes of the inconsistency we discovered was that there was an input DC offset that our comparator would introduce to our piezo’s AC signal that would change over time and affect our outputted square waves. We implemented safety guards in code to ignore frequencies that were too far from the frequency of the current string being tuned to account for the noise. In addition, inaccuracies in our calculated frequencies would arise in the code due to how we were directly timing the periods of the square waves.

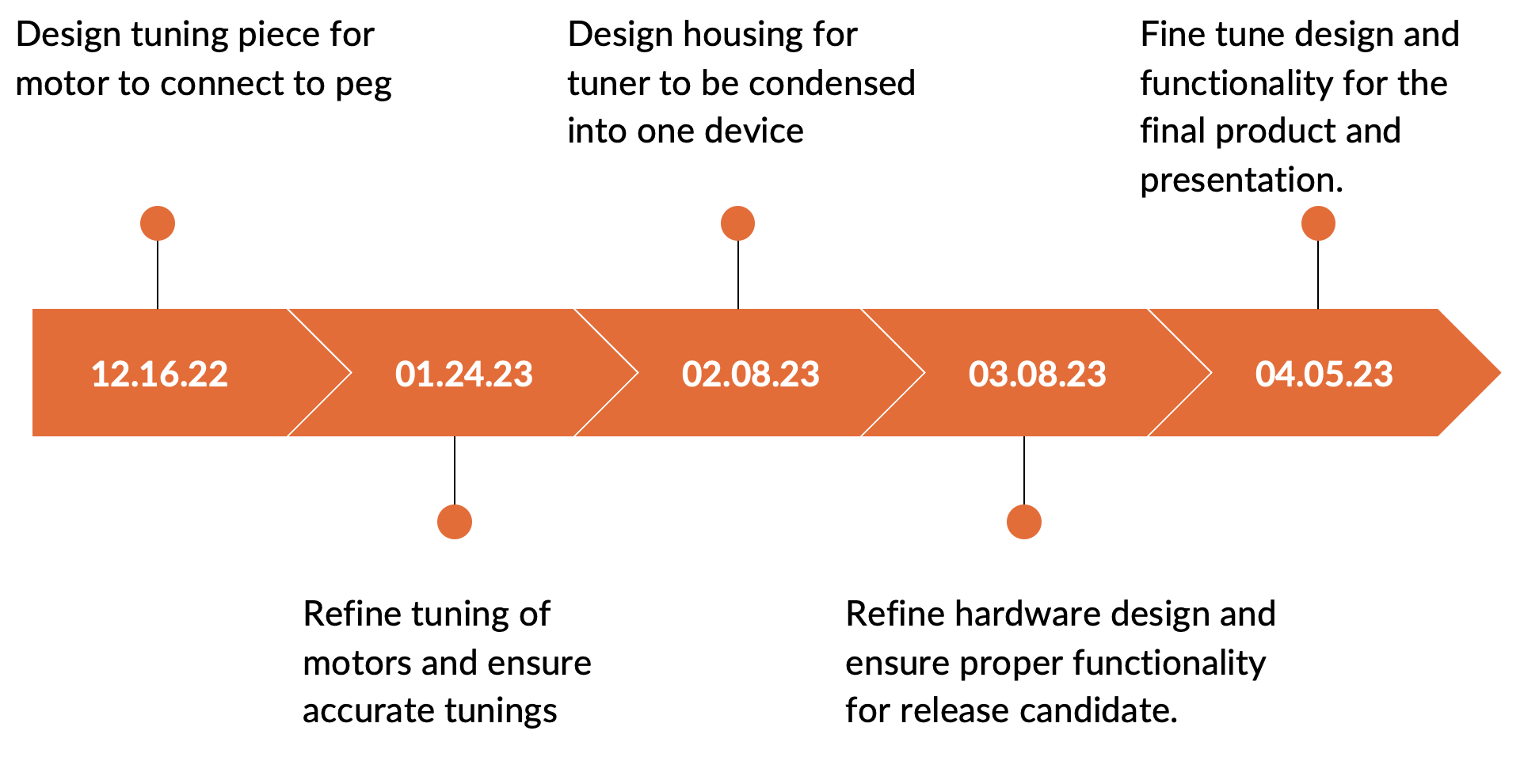


Figure 4: Prototype Project Timeline

For our alpha build, we primarily upgraded our motor turning and UI systems to better test our ability to tune a guitar. We upgraded our micro servos to much larger servos that could actually turn the tuning pegs of a guitar. In addition, we designed and 3D printed attachments that could attach the motors to the tuning pegs to allow them to turn the pegs. We integrated the UI with the rest of the design and began testing the whole tuning process. We altered sections of the code to get the frequency calculations to be more accurate despite the added complexity of the motors; however, we still had the same inconsistency problems as during the prototype phase.

We addressed the inconsistencies in the frequency calculations during the beta build phase of the project. After much research and experimentation, we designed a charge amplifier circuit to amplify our piezo signal. We discovered that very high frequencies were appearing at the output of our charge amplifier and designed a bandpass filter to filter these frequencies and undesired low frequencies out. To get rid of the changing DC offset that we were experiencing, we added an AC coupling capacitor to block an DC component from entering the comparator. After many bandpass filtering attempts, we realized that the large frequencies we were noticing were the 2nd harmonics of the strings we were strumming. To account for these 2nd harmonics, we added checks in code to check for double the desired frequencies. During this phase, we expanded to 6 motors with tuning attachments.

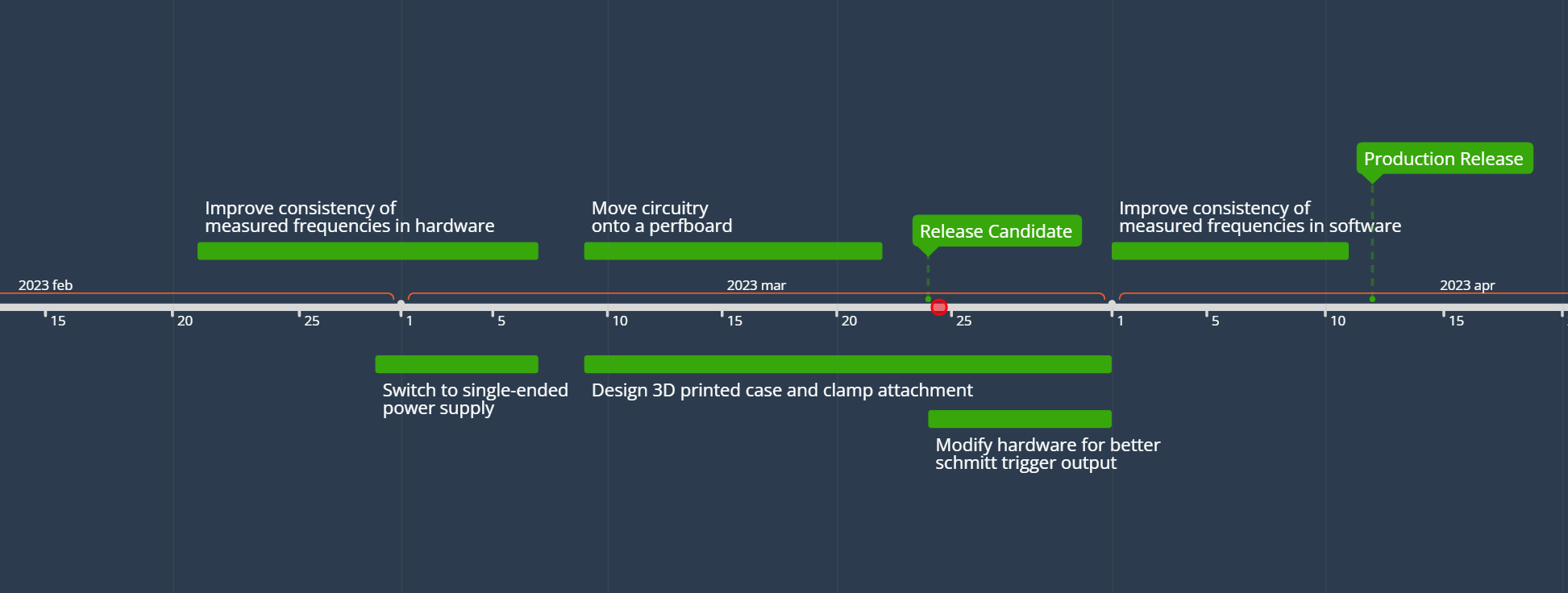


Figure 5: Project Timeline (Beta Build - Production Release)

Because most of the high frequencies were just 2nd harmonics, we removed the discrete bandpass filter for the release candidate. We also moved all of our circuitry to a single 3.3V supply during this phase. Despite all of the improvements to the signal conditioning circuitry, we still had a lot of inconsistency and inaccuracy in calculating frequencies. We shifted our circuitry to a perfboard and used a higher quality piezo cable to remove some randomness from the circuitry. We also shifted to an actual comparator based square wave generator rather than an op-amp based one. We added hysteresis to the comparator to prevent faulty triggering due to noisy signals. Due to the problems we were having discretely timing square wave periods while turning motors in the background, we shifted away from asyncio and found a synchronous way to calculate frequencies/turn the motors. We utilized the pulsein function, averaging, and other statistics to get a very accurate frequency that we tested against other commercial guitar tuners. We designed a case that could house the motors and all circuitry to mount to the headstock of a guitar and line up the motors, the piezo sensor. This case would attach to the guitar via clamps.

For the production release, we removed the clamps and utilized rubber bands to strap the tuner case to the guitar. We made aesthetic changes to our tuner case as well as some LCD messaging changes to enhance the user experience such as adding a tuning progress notification. We did further testing to enhance the performance of the tuner.

**Design**

The QTune automatic guitar tuner comprises three peripheral systems: the frequency measurement system, the tuning peg turning system, and the user interface system. The functionality of and communication between these peripheral systems is handled by an Adafruit RP2040 microcontroller whose firmware is written with CircuitPython. All these components physically interface with the guitar via a custom 3D printed case and are powered by a USB power bank.

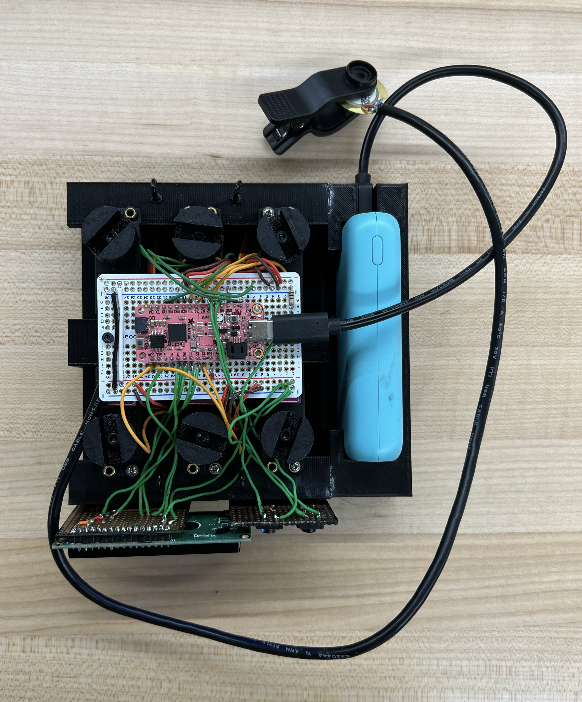


Figure 3: Top down view of the completed Release Candidate (left). Side view of the device clamped onto the guitar, with the motor tuning attachment properly fitting onto the guitar’s tuning pegs (right).

The frequency measurement system assesses the state of each guitar string as it is tuned by determining the frequency with which the string is vibrating after it is plucked by the user. The tuning peg turning system either turns the string in question clockwise, counterclockwise, or not at all depending on the state of the string reported by the frequency measurement system. The user interface allows users to choose the string that they wish to tune, informs them of when they need to pluck the string, and reports on the progress of the string being tuned with information from the frequency measurement system.

To determine the frequency of the guitar strings, our tuner utilizes a piezoelectric transducer sensor mounted to the headstock of the guitar. The piezoelectric sensor’s positive and negative nodes are wired through a shielded cable into a charge amplifier circuit that amplifies the sensor’s sinusoidal output. The sensor type, input resistor, feedback resistor, and feedback capacitor values were selected to create a passband of around 60 Hz - 360 Hz to filter out frequencies that the guitar could not produce. The sinusoidal output of the charge amplifier is then fed through a schmitt trigger circuit that converts the signal into a clean 0V - 3.3V square wave to be processed by the microcontroller.

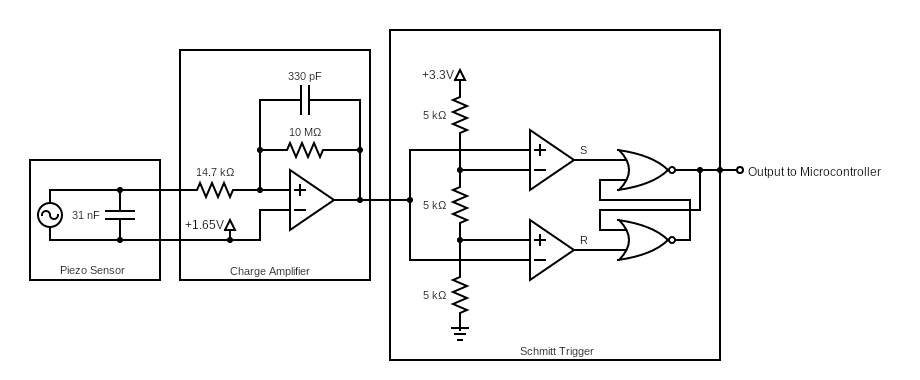


Figure 6: Piezoelectric Signal Conditioning Circuitry

The frequency is calculated based on the amount of time required for one rising edge and one falling edge - or one period - to be completed, and then taking the reciprocal. This is done via the pulseio module of CircuitPython, which measures a series of active and idle pulses that are read from the rx pin of the microcontroller and stores the times in an array. Accuracy is assured by using the average period time of the array before calculating the frequency. The length of each sample is determined by which note is being played, as higher frequencies require a larger series to maintain accuracy. Precision is assured by repeating this process and storing each measured frequency in another array until ten samples are reached. Only values around the interquartile range are used to remove outliers and then are averaged again to get the final frequency.

The previously calculated frequency will then be compared against the target frequency. The servo attached to the corresponding tuning peg will move either clockwise or counterclockwise depending on if it is higher or lower than the target frequency. If it is within ± 0.5 Hz of the actual value, then it is considered to be in tune and the tuning process is finished.

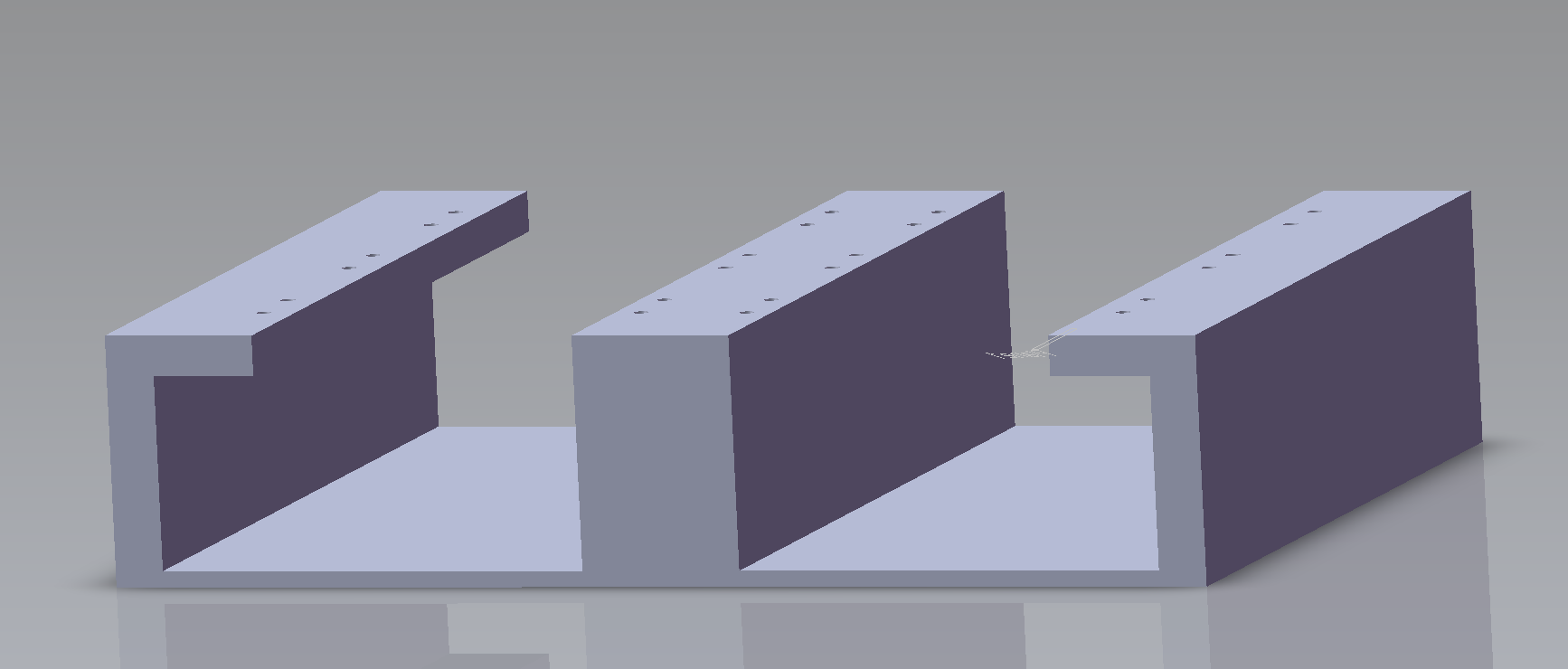


Figure 7: Finalized case for containing the servo motors, designed in Solidworks. The indicated holes align the motors to the correct tuning pegs.

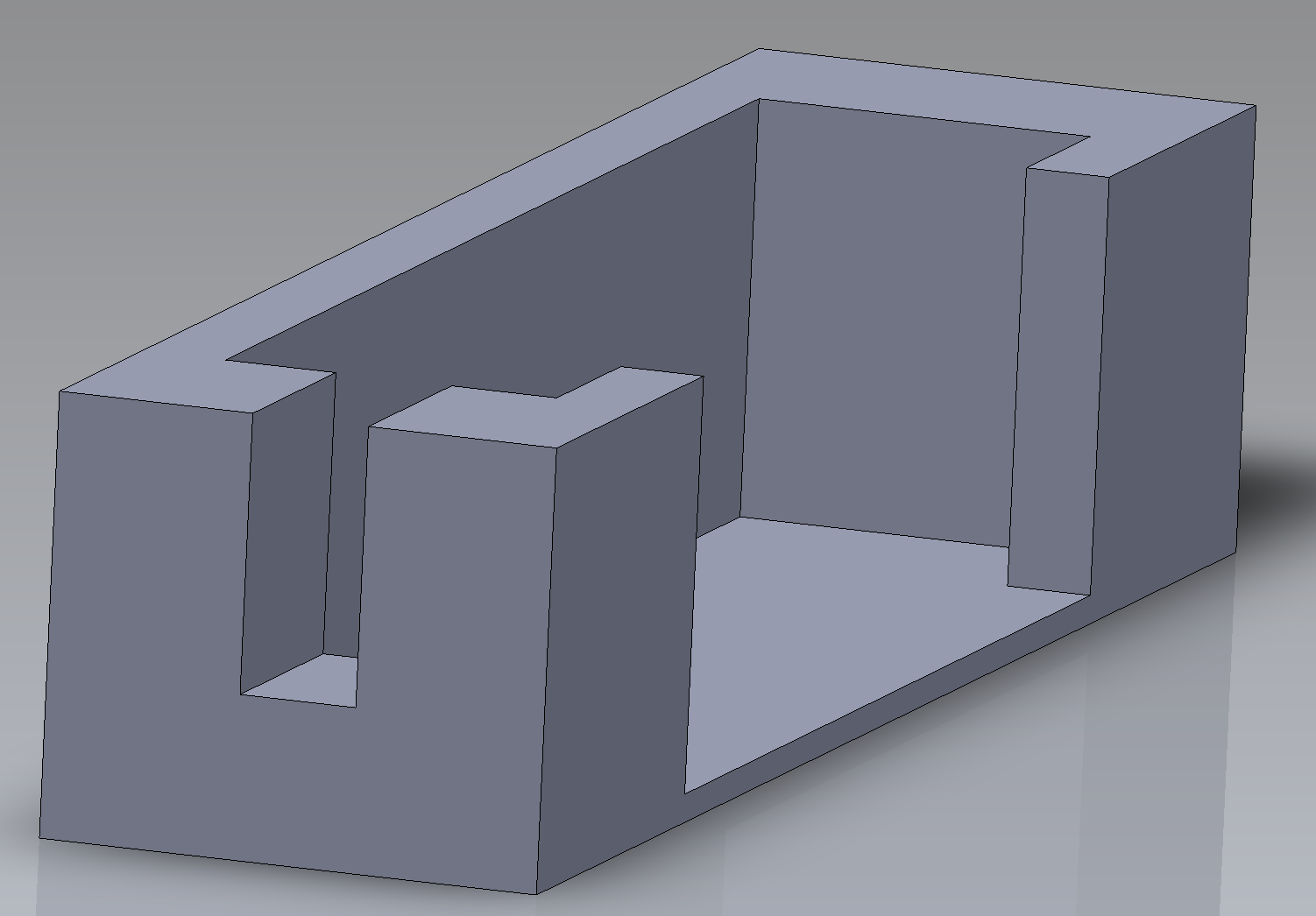


Figure 8: Finalized case for the power bank, designed in Solidworks.

**Impact**

To focus on the impact of this project, it’s important to focus on its positive impacts first, starting with its cultural and social impacts. Because this project intends to make the tuning process easier, it allows newer guitarists to continue developing their skills without having to worry about their guitar falling out of tune. This project also removes a learning barrier with people that struggle to get into playing guitar, so more people can actually become musicians. In general, this process should allow more people to become interested in making music, and since music is so intertwined with cultural developments, this project can help develop cultural trends if optimized to its fullest. The device’s compactness, speed of setup, and cost can be further improved with more development time and a generally better piezoelectric sensor. We believe that the device could get this introduction into the industry through smaller record companies or bands, as they could better compete with larger firms and their more advanced technology by speeding up the tuning process. The faster this goes, the easier for them to produce music, and change music and cultural trends sooner.

Considering its cultural and social impacts, one must also recognize its global impact. If this device proves to be a quick and effective tool for tuning guitars, then the technology should be able to spread globally for anyone to use. While English isn’t universally spoken, most countries around the world should be able to understand the technology due to its simplicity and the widespread use of English too. Because our coding is open for anyone to edit, anyone can edit the intended frequencies of each guitar string to tune the guitar to a different key, mode, etc.

In contrast, it’s important to highlight a drawback of the device- its environmental impact. This project requires an external battery source and multiple integrated circuits, and the byproducts of these electronics are often released in wastewater from PCB production facilities. These include chemical agents used to treat the perfboards, heavy metals such as copper and iron, and organic contaminants from the etching process. According to the study done by Geric, et. al, the results demonstrated that the concentration of contaminants of the wastewater was both cytotoxic (decreased viability) and genotoxic (can damage DNA) to human white blood cells. This leaves people more prone to getting sick, among other things. While treating the wastewater is highly effective at controlling this runoff, a lack of regulation can easily lead to contamination of surrounding environments.

Lastly, consider the economic impact of this project. Below is a running total of all the items used for creating the project and their associated costs. The 3D printed casing was done by the Marston Science Library printing lab. Do note that these costs could improve given ownership of a 3D printer and control of production of the servo motors.

*Cost Breakdown:*

Adafruit IC- $12

3D Printed Main-Body Casing - $15

3D-Printed Battery Case - $7

Piezoelectric Sensors (15 count) - $8

Servo Motors (2 for $16) - $48

Battery Pack- $10

Perfboards ($4.50 each) - $9

The whole cost of the device is approximately $110 dollars, and from a consumer aspect the economic impact becomes clear. Each device produced would cost at least that amount in order to break even, and even more to make a profit. Considering this and other options that consumers have at their disposal for tuning their guitars, they might forgo the automatic aspect of the project and opt for manual tuning with a listening device. Therefore, the economic impact of this project, if realistically marketed and sold in stores, would be limited. The focus would be on the automatic aspect of this device, and its ease of use would see a more niche group of consumers that would want to purchase it, such as beginning musicians.

Finally, in its current state, the device’s main limitation is the sensitivity of the piezo sensor. The comparator circuit and software can read and process all signals correctly, but at certain times, a few strings will have a very short time period where an accurate frequency can be read and processed. This is remedied through multiple strums, but the intended response is to have as few strums as possible. We have attempted to remedy this by shortening the pauses between data collection, with some success.

**Results**

Alpha Tests

1. Powering/Turning servos
   1. After determining that we had enough available pins on the Adafruit board, we connected all six servo motors and tested different voltages to find the proper voltage level needed to power the motors. After testing, we concluded that 5V was enough to power the motors. We no longer use the default 3.3V on the Adafruit board and instead use the USB pin to obtain 5V.
2. Determining total power required
   1. After testing, we concluded that all components of our guitar tuner could be powered simultaneously via the 5V and 3.3V pins on the Adafruit board. The Adafruit board could successfully power its internal components, the LCD display, the piezo sensor’s circuitry, and at least 1 servo motor at a time via a USB power bank plugged into its USB-C port. We determined that a USB power bank is a suitable power source for our final tuner design as it can make the whole tuner rechargeable and easily able to be turned on and off.
3. Testing 3D printed tuning attachment
   1. Once the tuning attachment design was finalized and printed, we screwed the attachments onto each servo motor and tested them against each tuning peg. We turned the motor in small increments and tested how much strength was needed to turn the tuning peg. After testing, we’ve concluded that the tuning attachment design was a suitable shape for turning the tuning pegs.
4. Testing tuner for each string
   1. As we began testing the accuracy of our tuner, we determined that more circuitry was required to condition the signal from our piezo sensor (amplification, filtering, noise reduction, removal of DC offset). After implementing this new functionality, we encountered an issue where the second harmonic of each string was appearing more so than the fundamental frequency we were looking for. We added to our code to account for this issue. When testing our tuner after these progressions, we achieved our first successful tuning of the low E string.

Beta Tests

1. Guitar string accuracy
   1. In order to ensure that the software was reading the frequencies correctly for each string, we performed a variety of tests for each string. First, we used the DAD and fed in the comparator output to the oscilloscope and had WaveForms calculate the frequency to make sure the comparator was accurate. We compared these results to a clip-on tuner. We then ran the program on the adafruit and had it calculate the frequency and compared it to both the DAD and the clip-on tuner to ensure accuracy. We then put each string out of tune and made sure that software continued to record accurate frequencies. All of these tests were successful, with the software recording nearly perfect readings each time.
2. Motor Tests
   1. We then used the frequency data to feed into the motors to turn the proper direction. We went through each string and put them out of tune and ran through each string and ensured that the motors could properly tune the string within a certain threshold. We first started with a large frequency threshold and slowly made the threshold smaller to make sure the motors could tune very accurately. These tests were very successful with the tuner being able to tune within +/-0.5 Hz.
3. Environmental Tests
   1. We then tested the guitar in different positions to make sure that this did not have an effect on the readings of the guitar and motors tuning. Upon moving the guitar to different positions, such as being held normally instead of flat on the table, the tuner continued to work as expected.
4. Durability Tests
   1. To test durability, we attached the tuner to the end of the guitar and moved it as a normal guitarist might and made sure the tuner did not move much. The tuner would stay on with our current clamp design, however with too much movement, the tuner may fall off. This is expected though, as the design is supposed to be removable and therefore too much movement could potentially make it fall off. With normal, non rigorous movements, the tuner is very stable.

The completed hardware elements of the project include a 3D printed case enclosing all components. Measurements for component lengths and the guitar were taken using a caliper for construction of the case in SolidWorks. This contains the six motors that are each attached to a tuning peg and a small hub that holds a rechargeable power bank to power the device and the LCD and push button user interface. The interface is attached to the outside of the case and painted to match. The circuitry itself has been condensed onto a perfboard to fit neatly between the motors. Consisting of the charge amplifier and schmitt trigger circuits, the output is sent to a secondary perfboard that holds the Adafruit RP2040 microcontroller. All wired connections are also hidden within the case. The enclosure is attached to the guitar with elastic bands that attach to hooks on opposite sides of the device. The tuning pegs of the guitar then fit precisely into the motors of the device allowing the motors to efficiently and accurately tune the guitar.

As for software, the frequency measurements have been cleaned up to produce very accurate results, with frequency readings being almost perfect (within +/- 0.1Hz) in the best case. This allows the device to tune the guitar to within +/- 0.5Hz of the desired frequency. The LCD display continuously shows the current progress of the tuning, notifying users when the tuning is done. The software has also been modified to turn the motors for longer the more out of tune the string is, and shorter the closer it gets, producing much quicker and more accurate results. The users are now also able to exit a tuning and return to the initialization screen if they select the wrong string.

**Conclusion**

In conclusion, the automatic guitar tuner project aims to develop an innovative solution that can reliably automate the process of tuning a guitar. With features such as an LCD interface with buttons and accurate frequency measurements, the tuner will provide reliable and consistent tuning for both beginners and experienced guitarists. To ensure the device tunes as precisely and accurately as possible, multiple tests and adjustments were made to the design of the tuner throughout the whole process while following proper engineering practices.

Ultimately, the project seeks to enhance the quality and enjoyment of the guitar playing experience, which is done using the device’s intuitive and responsive interface. This device will enhance the guitar playing experience by minimizing the frustrations and time it takes to tune a guitar. Especially for beginner guitarists who have yet developed good intonation, this device will help aspiring guitarists jumpstart their musical journey without having to worry about knowing how to tune.

**References**

1. Colin, “Guitar tuners 101: The Ultimate Buyer's Guide,” E-Home Recording Studio, 02-Mar-2023. [Online]. Available: https://ehomerecordingstudio.com/best-guitar-tuners/. [Accessed: 24-Mar-2023].
2. E. Moffatt, “Nashville-based Gibson says self-tuning guitars are here to stay,” 90.3 WPLN News, 29-Jun-2015. [Online]. Available: https://wpln.org/post/nashville-based-gibson-says-self-tuning-guitars-are-here-to-stay/. [Accessed: 24-Mar-2023].
3. Evstark and Evstark, “Modeling and experimental tools with prof. magnes,” Modeling and Experimental Tools with Prof Magnes, 12-May-2019. [Online]. Available: https://pages.vassar.edu/magnes/2019/05/12/studying-and-modeling-guitar-harmonics-using-fourier-analysis/. [Accessed: 24-Mar-2023].
4. J. Karki, “Signal conditioning piezoelectric sensors - Texas Instruments,” TI, Sep-2000. [Online]. Available: https://www.ti.com/lit/an/sloa033a/sloa033a.pdf. [Accessed: 25-Mar-2023].
5. M. Gerić, G. Gajski, V. Oreščanin, A.-M. Domijan, R. Kollar, and V. Garaj-Vrhovac, “Environmental risk assessment of wastewaters from printed circuit board production: A multibiomarker approach using human cells Author links open overlay panel,” ScienceDirect, Feb-2017. [Online]. Available: https://doi.org/10.1016/j.chemosphere.2016.10.101. [Accessed: 24-Mar-2023].
6. “Roadie Tuner Automatic guitar tuner,” Agazoo, 10-Jun-2015. [Online]. Available: https://agazoo.com/offers/unique-gadgets/roadie-tuner-automatic-guitar-tuner/. [Accessed: 24-Mar-2023].
7. “Piezo tuners - the clip on Tuner,” SWAMP Industries, 26-Oct-2022. [Online]. Available: https://www.swamp.net.au/articles/piezo-tuners.html. [Accessed: 24-Mar-2023].