Fully Automated Guitar Tuner

ECE 445 Design Document

Benjamin Wang, Brandon Ramos, Cooper Ge
Team 71
TA: Vassily Petrov
4/17/20

0 Table of Contents

1 Introduction

1.1 Problem and Solution Overview

It's an unfortunate fact of life that guitars fall out of tune over time. Playing on pitch is hugely important, but is impossible to do if the instrument itself is out of tune. Those with perfect pitch may do it by ear, but experts estimate only 0.01% to 0.05% of the population has that ability [1]. Everybody else must rely on a tuner. The most common tuners will tell you whether a specific string is sharp or flat, however players must still physically adjust the tuning knobs and manually pluck the guitar string each adjustment.

We envision a complete package that will tune 3 strings of a guitar at a time with minimal user input. The device will be battery powered and compact, making it easily portable. The user would attach it to the head of the guitar and adjust the motor arms so they sit on the tuning pegs. The user would then attach the strumming motor and allow the system to tune the guitar completely hands free. Once finished, the process would repeat on the other side's three strings, making tuning as automatic and convenient as possible.

1.2 Visual Aid

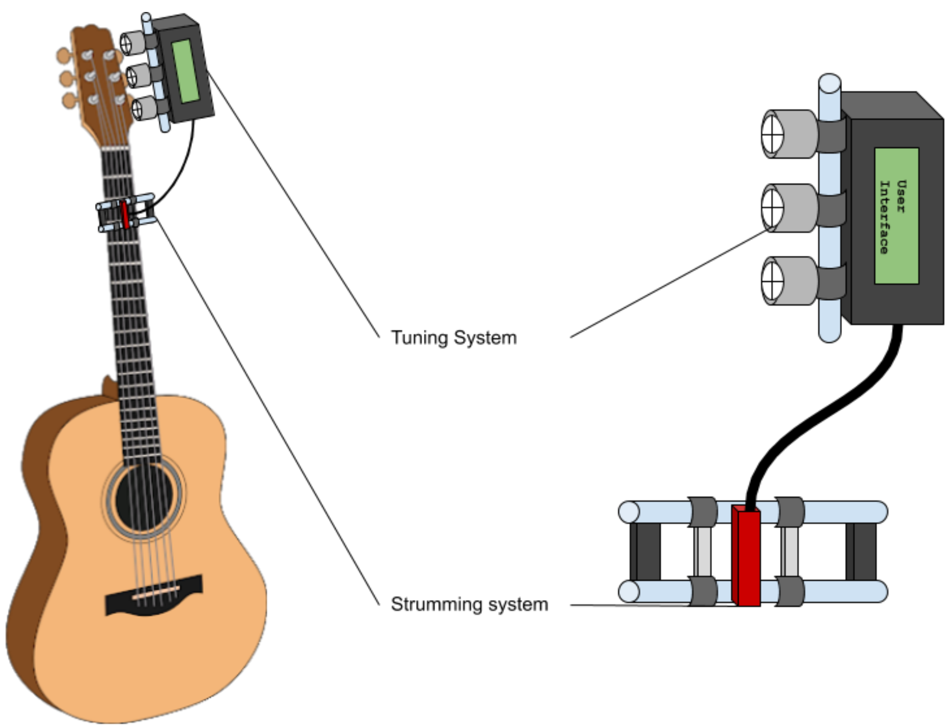


Figure 1.2 Visual Aid

1.3 High-Level Requirements

- Identify triple pitches to within 20 cents each
- Tune guitar within 40 cents
- Tune within 1 minute

2 Design

2.1 Block Diagram

We have divided our automatic guitar tuning system into three primary subsystems: the audio recording system to collect the audio information, the mechanical systems to interface with the guitar, and the processing system to both control the mechanical systems and determine how they should be controlled. All block connections are wired, with power lines and data lines being differentiated by color. The audio line connection is indirect as it has to interact with the guitar for sound to be produced for the microphone to record.

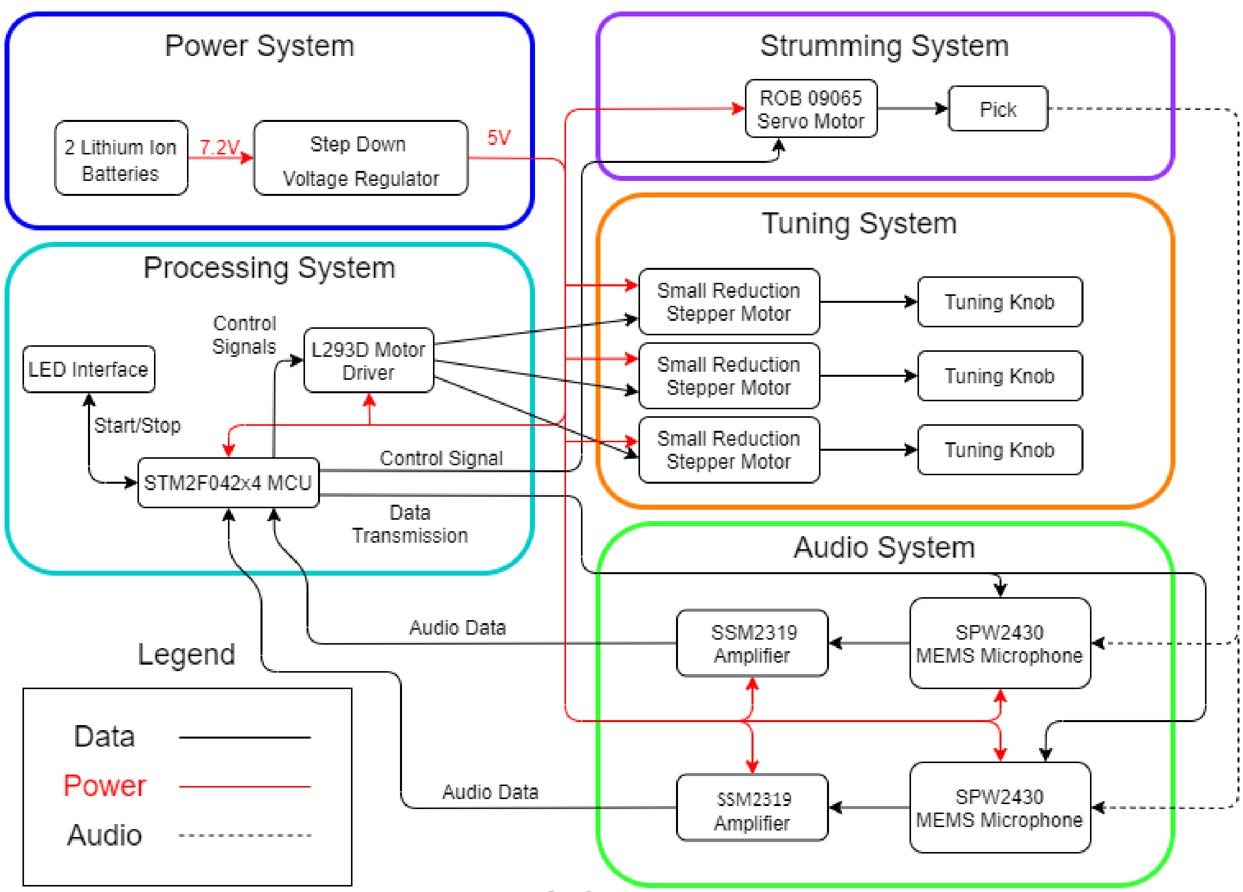


Figure 2.1 Block Diagram

2.2 Physical Design

The functionality of our design depends on being able to interface with a variety of guitars. Unfortunately, there does not exist a standard shape nor size for guitars across brands [9]. The shape of the headstock, the part of the guitar where the tuning knobs are located, similarly have great variation in size and shape. Tuning knobs can be configured all six lined up on one side or three on each side. In the interest of universal guitar compatibility, the tuning motors are adjustable from 1" to 3.5" and the strumming motor adjustably clips to the neck at a generous 2.5" rail length for the strumming range to the standard 1.75"-2" guitar neck width.

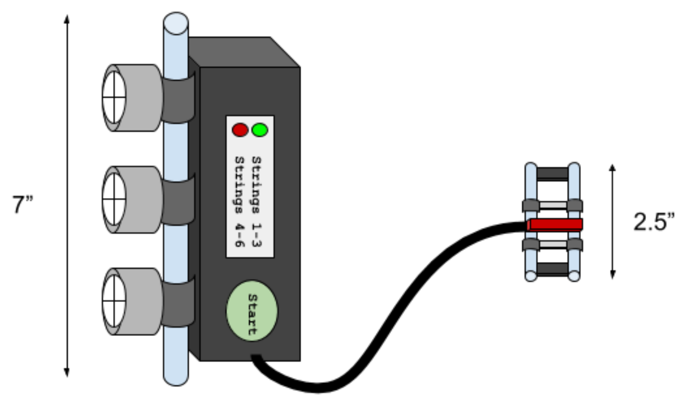


Figure 2.2.1 Physical Design Sizing

2.3 Mechanical Systems

The mechanical systems are the most novel aspects of our design intended to automate the traditional human components of tuning. The design utilizes four individually addressable motors with custom fittings to interact with the guitar that is being tuned. A combination of stepping motors for tuning and a continuous servo motor for strumming are used. Control of the motors are dictated by the control logic and computations of the processing system and the mechanical systems here most largely rely on the mechanical design of the fittings to work in our design.

2.3.1 Tuning Motor System

For the tuning motors, we have chosen to use the Adafruit 858 stepper motors in order to have fine control over the degree of rotation. Our design necessitates three bidirectional stepping motors and each of three motors need their own driving controller. The controller we chose for our motors was the Dual H-Bridge Motor Driver L293D.

The Adafruit 858 stepper motors will be able to programmatically adjust the guitar tuning knobs by passing the appropriate control bits to the motor controllers from the processing system. Each of the controllers will be connected to the processing system for both feedback and control. The motors will interact with the tuning knobs using custom fittings to attach to the knobs.

2.3.2 Strumming Motor System

The strumming system will be composed of a single servo with a pick attachment mounted on its horn. The servo motor we selected is capable of running off 5V, and has a rated torque of 22.22 oz*in, is more than the estimated 1.4 oz*in required to strum a guitar string [14]. The motor model is the ROB-09065 by SparkFun.

The ROB-09065 will be able to programmatically strum the guitar strings by utilizing the control logic from the processing system to determine when to run the motor across the strings, forward or backward. The strings that are plucked by the system will be controlled by adjusting the mechanical stops on the rail that the servo will move along. Only a PWM output from the processing system is needed to control the servo of this system.

2.3.3 Mechanical Systems Requirements and Verifications

Table 1. Mechanical Systems Requirements and Verifications

Requirements	Verifications
Tuning Motor System 1. The tuning system should begin tuning the guitar knobs within 2 seconds of receiving the command from the microcontroller	**Tuning Motor System** 1. Verification process for Item 1: a. Isolate the tuning system and simulate signals to the tuning system b. Measure the time it takes for the system to tune begin tuning the knob after receiving the signal with a stopwatch
Strumming Motor System 1. The strumming system should strum the guitar string within 2 seconds of receiving the command from the microcontroller	**Strumming Motor System** 2. Verification process for Item 1: a. Isolate the strumming system and simulate signals to the strumming system b. Measure the time it takes for the system to strum a string after receiving the signal with a stopwatch

2.4 Audio System

The audio system establishes a connection between the strumming motor system and the processing system, sending the correct audio data from the string strummed by the motor system to the processing system. This would then enable the processing system to analyze the frequency and figure out how much torque is required for the tuning motor system. Since a cardioid microphone configuration is implemented in this system, the two microphone readings will have to be separately read in with the processing system and averaged together.

2.4.1 Microphone

The microphone in the audio system allows the communication and recording between the guitar and our microprocessor. Our design could be greatly bottlenecked by the use of an inaccurate or imprecise microphone so a good deal of care was taken in the selection of the SPW2430 MEMS microphone. We chose the SPW2430 because it uses favorable MEMS technology for digital noise tolerance and package size [4] and 5V power usage.

In order to maximize microphone clarity of the guitar sound, we plan on placing two SPW2430 microphones in a cardioid configuration inside the enclosed clamped to the headstock of the guitar with the processing unit. The cardioid configuration is a directional microphone configuration that is superior in the suppression of background noise and room reflection [12]. Directional miking makes sense in our design because we are only interested in measuring sound from the strings which will not move relative to the tuning enclosure with minimal background noise. Shown in Figure 2.4.1 is the cardioid configuration of two microphones.

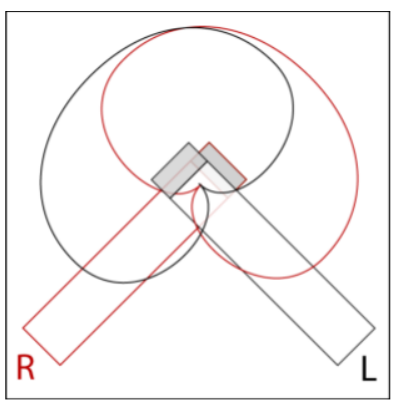


Figure 2.4.1 Cardioid Microphone Configuration

2.4.3 Amplifier

Due to the cardioid configuration using two microphones, there must similarly be two amplifiers. Our amplifier choice was dictated by compatibility with both our microphone choice and processing system choice. In order to most effectively leverage the ADC capabilities of the processing system, the microphone system must output as close to the full range of the ADC as possible with minimal amplifier overhead and distortion. For these reasons and the chips favorable 5V power supply and consumption, we elected to implement the SSM2319 audio amplifier into our design. The output of the two audio amplifiers will go directly to their respective pins in the processing system's ADC.

2.4.4 Audio System Requirements and Verification

Table 2. Audio System Requirements and Verifications

Requirements	**Verifications**
Microphone 1. Must be able to accurately detect the tune of the guitar string being strummed by the strumming system	**Microphone** 1. Verification process for Item 1: a. Connect microphone to microcontroller b. Start with the microphone directly next to the guitar string being strummed by the strumming system c. Slowly move the microphone further away while looking at the audio data

2.5 Processing Subsystem

The processing system is the brains of our design. It will take in audio data transmitted by the audio recording system, process the difference between the current pitch and the ideal pitch, then send that data to our tuning system where we tune the guitar knobs. This system also includes the user interface system and the power supply since it will be physically housed in the same enclosure.

2.5.1 Algorithm

The algorithm is responsible for identifying the pitches being played, which will allow the microcontroller to compare the detected pitch with the desired pitch, and tune the string accordingly.

2.5.1.1 Background

Musical notes are the fundamental building blocks of music, and is commonly notated with the first seven letters of the alphabet - A, B, C, D, E, F, G. These can be further modified by accidentals, which are called sharps and flats, and denoted by #/b respectively, making 12 total notes - C, C#, D, D#, E, F, F#, G, G#, A, A#, B. Each note has a fundamental frequency; for example, middle C is approximately 261.6 Hz. A lower frequency corresponds to a lower pitch, and vice versa. Doubling the fundamental frequency will generate the same note, just an octave higher. For example, 440 Hz is an A, and 880 Hz is also an A. Each note will also have harmonics at integer multiples of its fundamental frequency. For example, a 220 Hz A will have harmonics at 440Hz, 880 Hz, 1320 Hz, 1760 Hz and so on, as shown in the figure below.

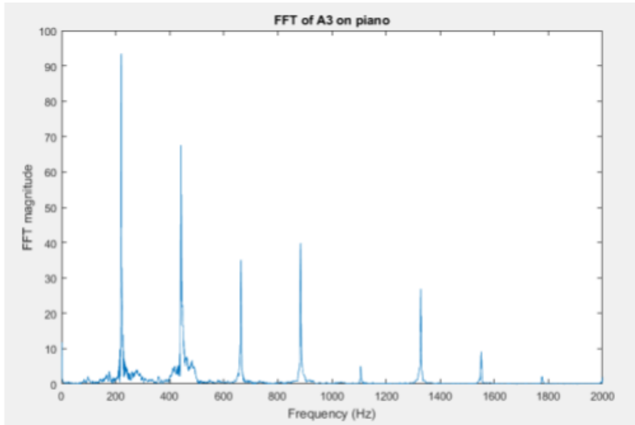


Figure 2.5.1.1 Frequency Composition on Piano

A chord is simply two or more notes played at once, and is how our project sets itself apart. Common pitch detectors, including the ones in commercial tuners, cannot handle multiple inputs at once.

2.5.1.2 Algorithm Summary

Our algorithm will leverage Fujishima's 1999 paper [5]. His algorithm involves taking the DFT of the signal, generating a spectrum bin table, and using that to sort the DFT into a pitch class profile. He then proceeds to use pattern matching to identify the chord, which will not be necessary for our purposes. His graphic from his paper is reproduced below.

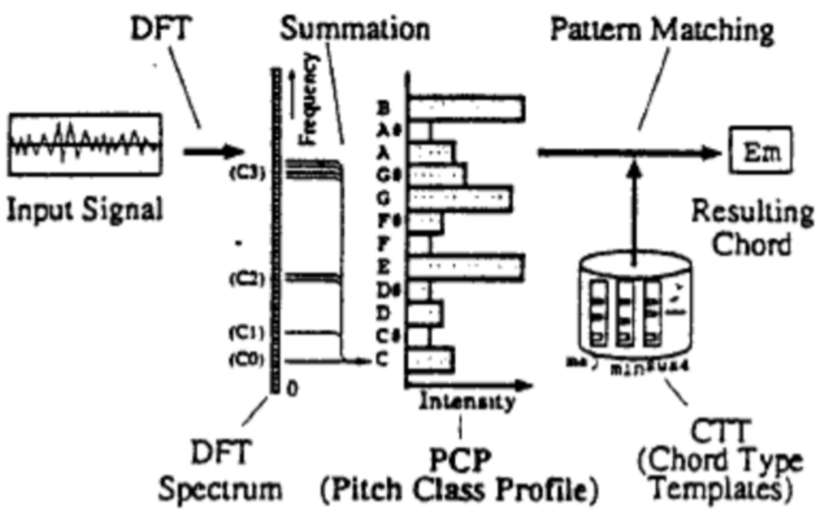


Figure 2.5.1.2.1 Algorithm Graphic

Our algorithm flow chart is pictured below.

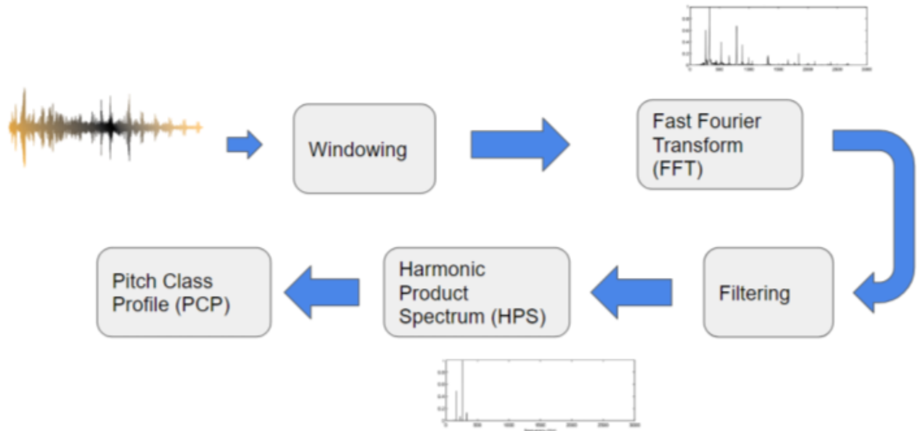


Figure 2.5.1.2.2 Algorithm Flow Chart

Since we cannot feasibly process the entire length of the signal, we must first window the incoming data. In our prototype testing, we found no significant difference between different windows, and chose the hamming window for its side lobe attenuation. We then take the fourier transform of the windowed signal to convert the signal into the frequency domain, and pass it through a bandpass filter to attenuate the frequencies that are not associated with our desired notes or chords. The narrower the band, the less intensive the algorithm would have to be, but we had to make sure we weren't filtering out data that we wanted. We found that for guitar, we could filter out frequencies outside 60 - 2000 Hz. As mentioned earlier, notes have harmonics at integer multiples of their frequency. These harmonics can often overlap with other notes in the chord, making it hard to distinguish what is a fundamental frequency and what is a harmonic. This is where the harmonic product spectrum comes in. The HPS cuts off higher order harmonics, leaving either just the fundamental frequency, or a lower number of harmonics.

From that, we generate our pitch class profile, a twelve dimension array, which contains the intensity of each pitch with the following equation. $PCP(p) = \sum_{l\ s.t\ M(l)=p} ||X(l)||^2$

where $M(l)$ is the spectrum bin table. $M(l)$ is defined according to the equation below, where f_s is the sampling frequency, N is the length of the FFT, and f_{ref} is the reference frequency desired for $PCP(0)$.

$$M(l) = -1\ for\ l = 0$$

$$M(l) = round(12 * log_2(\frac{f_s * \frac{1}{N}}{f_{ref}}))mod12\ for\ l = 1, 2, ..., \frac{N}{2} - 1$$

2.5.2 Control Logic

The control logic is critical in making sure the mechanical systems both correctly create audio input for the algorithm and correctly implement the tuning adjustments to the strings. This system is created in software on the processing core according to the diagram shown in Figure 2.5.2. The control logic design specifies the order in which mechanical events occur and implements the tuning calibration of the tuning knob adjustments. Tuning calibration is important because not all guitar tuning knobs have consistent change in string tension per tuning knob rotation.

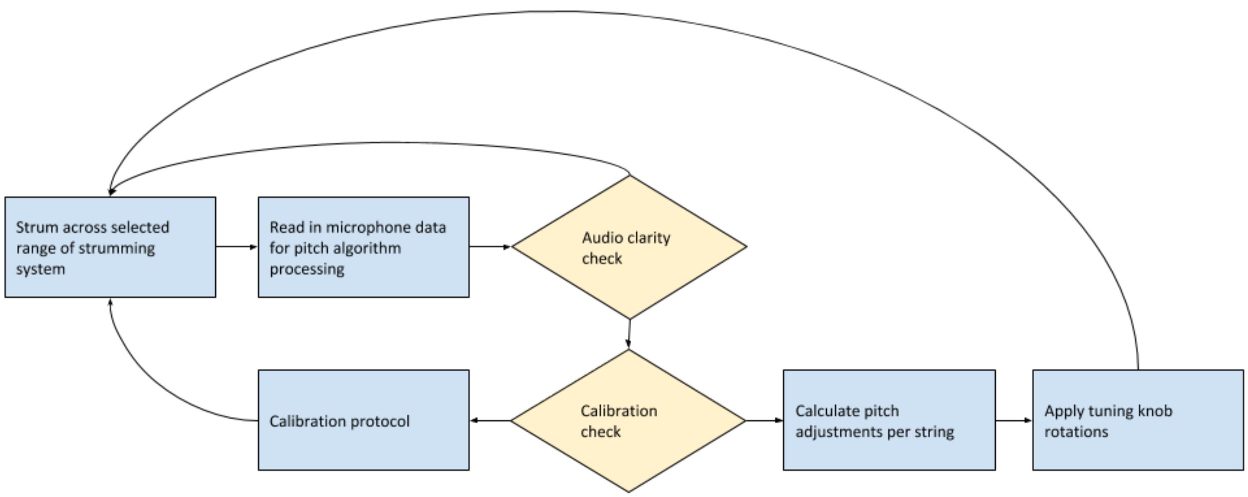
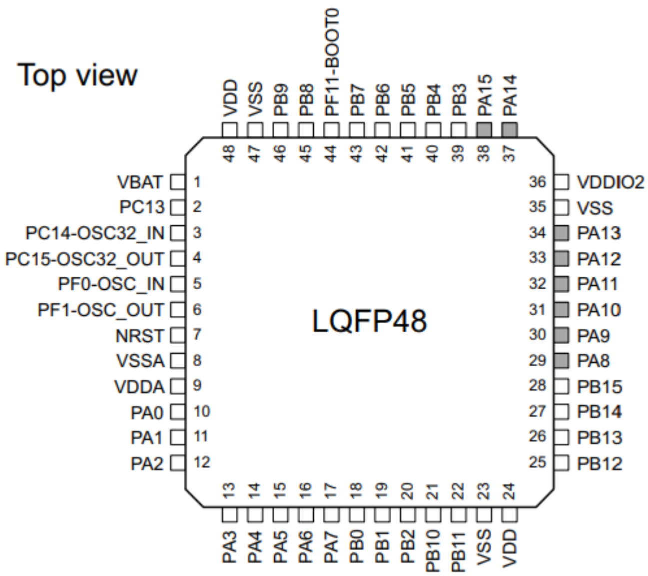


Figure 2.5.2 Control Logic Flow Diagram

2.5.3 Microcontroller Unit

Due to the computation-heavy nature of our pitch detection algorithm across three strings, we have elected to use the STM32F042x4 MCU in the LQFP48 package. We chose this microcontroller because of it's 5V power compatibility with our mechanical systems, 0-3.6V 12-bit ADC, I2C support for our microphone, 32-bit ARM architecture for advanced computing, and prevalence in development boards for prototyping. The microcontroller will implement both the pitch detection algorithm and the control logic. Shown below is the pinout for the surface-mount package and the relevant pin mappings for our design.



Relevant Pin Mapping

Pin	Connection
PA0, PA1	Microphone amplifiers out
TIM2, TIM3, TIM14, TIM16	Servo modulation signals
PA9, PA10	I2C USB programmer
VDDIO1, VDDIO2	Interface LEDs
PA7	Start button

Figure 2.5.2 Microcontroller Pinout [10]

2.5.4 Processing System Requirements and Verifications

Table 3. Processing System Requirements and Verifications

Requirements	**Verifications**
Processing System 1. Processing system must be able to communicate simultaneously with all the mechanical systems 2. Tuning of the guitar should take no longer than a minute to complete	**Processing System** 1. Verification process for Item 1 a. Connect all the parts and attach to guitar b. The strumming and tuning sequence should continue until the guitar is correctly tuned 2. Verification process for Item 2 a. Connect all the parts and attach to guitar b. Measure the time it takes for the for the tuning process to complete with a stopwatch

2.6 Power System

To power our device, we plan on using two lithium ion batteries in series, with a nominal voltage of 7.4V. Since our electronics all require 5V or less, this means that we only need a step-down converter instead of a step-up/step-down converter, slightly simplifying our design.

The amplifiers draw up to 4 mA each, the microcontroller draws up to 500 mA, the stepper motors can draw up to 600 mA, the strum motor will draw an estimated 165 mA, giving us a total of 2.5 A. This leads us to choose the D24V22F5, which is capable of taking our input voltage and delivering 5V at currents between 1.9 and 2.5 A. It is also a switching regulator, and has an estimated efficiency of above 90% based on our input voltage and current needs. Lithium Ion batteries can have a capacity of up to 3500 mAh, meaning our device could run for over one hour straight at maximum power. We think this is acceptable given that the device will very rarely be at maximum power, and that it will be used sporadically. If the user were to use it for 5 minutes each week, it would go almost 4 months without a charge.

2.7 Tolerance Analysis

Manually tuning a guitar may take five minutes for a beginner guitarist but, generally, experienced guitarists can tune a guitar in around 30 seconds [11]. In order for the automatic guitar tuner to be worth using, it is essential that we, at minimum, meet our high level requirements of tuning accurately within 1 minute. By our estimation, meeting those requirements rely on our design's ability to accurately identify pitch and reliably interact with the guitar.

2.7.1 Pitch Detection

Identification of the pitch of each guitar string is reliant on the audio information we are able to collect from the microphones. Theoretically, according to the algorithm we are using outlined in section 2.5.1, the only bottleneck on the frequency information we can recover is the sampling rate and number of FFT points used. The DFT formula is given by

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi nk}{N}} ,$$

where N is the number of FFT points. The frequency resolution is given by

$$\frac{f_s}{N} .$$

f_s is dictated by how fast our microcontroller can sample our microphone. In this case, we can read our two microphones at a sampling rate of up to 1MHz. However, our microphones are only capable of detecting up to 10 kHz, and therefore by the Nyquist theorem, we should sample at a minimum of 20 kHz. We chose an FFT length of 1024, giving us a frequency resolution of 19.53 Hz.

The standard tuning for the six strings on the guitar E2, A2, D3, G3, B3, and E4 have frequencies 82.41 Hz, 110 Hz, 146.8 Hz, 196 Hz, 247.9 Hz, 329.6 Hz. The minimum frequency difference between the strings we must then be able to resolve is between E2 and A2, 27.59 Hz.

Additionally, we must be able to resolve pitches within 40 cents. This criteria is less consequential than the 27.59 Hz baseline, however, because the pitches will have harmonics that allow the relative cent scale to be seen on the higher order harmonics which double in frequency every order. Since our frequency resolution is less than the minimum frequency distance between pitches, we will be able to successfully detect pitch for our design.

2.7.2 Mechanical Calibration

It is worth considering whether or not our design has enough mechanical precision to accurately perform the gestures needed for tuning efficiently-- namely, strumming and turning the tuning knobs. Strumming is a simple enough problem to solve. Designing a servo motor to run along a rail across guitar strings is simply a matter of ensuring the servo has enough torque. Similarly, the stepping motors for tuning must also have a sufficient maximum torque to apply tension to the guitar strings. The torque ratings for both types of motors are guaranteed by the manufacturer to be sufficient for our use case. A much more serious consideration, however, must be given to how calculated pitch differences for each string will be translated into tuning knob adjustments.

Figure 2.7.2 Tuning Knob Mechanism [13]

Tuning knobs adjust string tension by driving a gear in the headstock of the guitar attached to a post that the guitar string gets continually wrapped around. Shown in Figure 2.7.2 is the tuning knob mechanism. Just as there does not exist a standard guitar size and shape, there does not exist a standard post size to mount the strings onto nor a standard gear size to rotate that post. The formula for standing waves on strings is as follows:

$$f = \frac{1}{2L}\sqrt{\frac{T}{\mu}}$$

where f is the frequency of vibration, L is the length of the string, T is the tension of the string, and μ is the constant linear density of the string. The number of rotations required to make a specific pitch adjustment on a tuning knob not only on how much the string length is adjusted by the tuning knob, but also how much each degree of rotation adjusts the length of the string. Luckily, guitar strings are designed to linearly change tension with the square of string length in their respective frequency ranges. By leveraging this fact, we can simplify the calculation of the first harmonic frequency of standing waves to be the following:

$$f = \frac{1}{2}\sqrt{\frac{x}{\mu}}$$

where *x* is a parameter directly proportional to knob rotations. This means that the relationship of knob rotations to string frequency can be calculated with just two data points. This means that every string can be tuned in approximately three iterations by our design, well under the 1 minute time requirement.

3 Project Differences

3.1 Overview

The original project did not physically tune the guitar. It was a tuner in the commercial sense of the word, and merely measured and displayed the pitch being played. They utilized a DSP processor to perform a length 1024 FFT and peak detection on the signal sampled at 8 kHz to determine the fundamental frequency. The key differences between ours and the original are the fact that ours physically tunes the guitar, and handles more than one pitch at a time. The tradeoff is that it sacrifices portability and efficiency for minimal user input.

3.2 Analysis

The design is not meant to improve on current tuners. The market has been saturated, and there is not much improvement left to be had; current tuners are already accurate to ±9 cents, with more quality examples accurate up to ±3 cents, in comparison to our tuner, which we aim to be accurate to ±20 cents. Rather, our device is targeted at user convenience. A shop owner or a music teacher who is responsible for multiple guitars could use this and simply hit a button instead of spending valuable time at the end of the day to tune their stock.

4 Cost and Schedule

4.1 Cost

Table 6. Parts Cost Summary

Description	**Quantity**	**Vendor**	**Total Cost**
Small Reduction Stepper Motor	3	Adafruit	$14.85
Dual H-Bridge Motor Driver L293D	3	Adafruit	$8.95
Servo - Generic (Sub-Micro Size)	1	Sparkfun	$8.95
SSM2319CBZ-REEL	2	Mouser	$1.80
STM32 F0 ARM Development Board	1	Banggoood	$3.94

MEMS Microphone Breakout - SPW2430	2	Adafruit	$9.90
Capacitor and Resistor Budget	x	x	$20
Mechanical Build Budget	x	Machine Shop	$200
Pinout/PCB Interface Budget	x	x	$30
PCB Printing(General)	1	x	$35
			Total: $333.39

Table 7. Labor Cost Summary

Engineer	**Hours Expected**	**Hourly Rate**	**Cost + Overhead**
Ben	156	$50	$19,500.00
Brandon	156	$50	$19,500.00
Cooper	156	$50	$19,500.00
(Machinist)	20	$40	$2,000.00
			Total: $60,500.00

Table 8. Grand Total Cost

Section	**Total**
Materials	$333.39
Labor	$60,500.00
Grand Total	$60,833.39

4.2 Schedule

Table 9. Project Schedule

Week	**Tasks**	**Responsibility**
03/27/20	**1. Project Approval**	1. All
04/03/20	**1. Project Proposal** 2. Complete general idea of prototype design	1. All 2. All
04/10/20	1. Research and select hardware components 2. Initial design of mechanical layout and tolerance analysis 3. Rough design of electrical schematic 4. Rough idea of algorithm	1. All 2. All 3. All 4. All
04/17/20	**1. Complete Design Document** 2. Order parts for prototyping 3. Ensure compatibility of proposed hardware components	1. All 2. Brandon 3. Cooper
04/24/20	**1. Conversation with machine shop** **2. Design Review**	1. All 2. All

	3. Prototype tuning system 4. Prototype audio recording system 5. Prototype strumming system	3. Brandon 4. Ben 5. Cooper
05/01/20	**1. Teamwork evaluation I** 2. Design and test algorithm	1. All 2. All
05/08/20	**1. Submit final mechanical design to machine shop** **2. Submit first central processing unit PCB design**	1. Brandon 2. Cooper
05/15/20	**1. Individual progress report** 2. Unit test strumming system 3. Unit test audio recording system 4. Unit test tuning system	1. All 2. Cooper 3. Ben 4. Brandon
05/22/20	**1. Final central processing unit PCB design** 2. Ensure high level requirements are met	1. Cooper 2. All
05/29/20	**1. Mock project demonstration** 2. Implement suggestions from mock demonstration	1. All 2. All
06/05/20	**1. Project demonstration**	1. All
06/12/20	**1. Complete final paper** **2. Lab checkout** **3. Teamwork Evaluation II**	1. All 2. All 3. All

5 Safety and Ethics

Following the IEEE Code of Ethics, our only issue in regard to Code of Ethics #1, that our design holds the safety of the public as its first priority and informs the public of any endangering aspects [6]. Our project utilizes rechargeable lithium ion batteries, which is our main source of power for our project. Lithium ion batteries have the potential to cause fire hazards from overheating [7] or environmental hazards if disposed of incorrectly [8]. We will monitor the battery temperature using a semiconductor temperature sensor and a LED to notify users when battery temperature goes above 55 degrees Celsius. We will also make it easy to deconstruct our system so that the lithium ion battery can be disposed of properly. Although we are using a microphone to take in audio input, we are directly processing the audio data and not saving it in any way, so this does not violate the code of ethics.

6 References

[1] *PubMed,* "Autism-related language, personality, and cognition in people with absolute pitch", 2020. [Online] Available: https://www.ncbi.nlm.nih.gov/pubmed/?term=pitch+autism+brown+folstein. [Accessed: 3-April-2020].

[2] *Roadie,* "Press", 2020. [Online] Available: https://www.roadiemusic.com/press. [Accessed: 3-April-2020].

[3]*Wikipedia,* "Electronic Tuner", 2020. [Online] Available: https://en.wikipedia.org/wiki/Electronic_tuner. [Accessed: 3-April-2020].

[4] *DigiKey,* "Electret Condenser (ECM) vs MEMS Microphone", 2017. [Online] Available: https://forum.digikey.com/t/electret-condenser-ecm-vs-mems-microphone/447. [Accessed: 3-April-2020].

[5] *ICMA,* "Realtime Chord Recognition of Musical Sound", 1999. [Online] Available: https://quod.lib.umich.edu/i/icmc/bbp2372.1999.446/1/--realtime-chord-recognition-of-musical-sound-a-system-using [Accessed: 3-April-2020].

[6] IEEE,"IEEE Code of Ethics", 2020. [Online] Available: http://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 3-April-2020].

[7] *ViceProvost,* "The Hazards of Lithium Batteries", 2020. [Online] Available: https://viceprovost.tufts.edu//ehs/files/The-Hazards-of-Lithium-Batteries.pdf. [Accessed: 3-April-2020].

[8] *ABRI,* "Environmental Impacts of Lithium Ion Batteries", 2020. [Online] Available: https://batteryrecycling.org.au/environmental-impact-of-lithium-ion-batteries/.[Accessed: 3-April-2020].

[9] *Guitar Gear Finder* " Everything You Need to Know About Guitar Sizes", 2018. [Online] Available: https://guitargearfinder.com/guides/everything-you-need-to-know-about-guitar-sizes/. [Accessed: 16-April-2020]

[10] *STMicroelectronics* "STM32F042x4 STM32F042x6", 2017. [Online] Available: https://www.st.com/resource/en/datasheet/stm32f042c4.pdf. [Accessed: 16-April-2020]

[11] *Guitar Tuning* “Guitar Tuning”, 2011. [Online]
Available: http://www.guitar-tuning.net/. [Accessed: 16-April-2020]

[12] *AZDEN* “Understanding Microphone Polar Patterns”, 2018. [Online] Available:
http://www.azden.com/blog/understanding-microphone-polar-patterns/.
[Accessed: 17-April-2020]

[13] *Wikipedia* “Machine head”, 2017. [Online]
Available: https://en.wikipedia.org/wiki/Machine_head. [Accessed: 17-April-2020]

[14] *WPI* “Assistive Guitar Plucking Device and User Interface”, 2019. [Online] Available:
https://web.wpi.edu/Pubs/E-project/Available/E-project-042419-130612/unrestricted/PAM_Band_MQP_Final_Paper.pdf. [Accessed: 17-April-2020]