Automatic Guitar Tuner

UNIVERSITY OF CENTRAL FLORIDA

Bryan Casey CpE, John Geiger EE, Adama Harmon EE, Jason Lupo EE
GROUP F | 04/27/2017

Table of Contents

List of Figures

List of Tables

1. Executive Summary

Tuning a guitar has been a difficult process for beginning guitarists since the creation of the instrument, and it has driven many potential musicians to different instruments, or even to quit music altogether. Only recently, the technology required to implement a more nuanced solution to this problem has become common. The continuity of Moore's Law and increasing access to powerful, yet cheap integrated circuits and microprocessors has made the dream of an automatic guitar tuner a possibility instead of just a dream. An automatic guitar tuner allows new musicians to tune their guitars with computer numeric control (CNC) precision, while an application allows them to learn what each note should sound like. The device also has applications for professionals like teachers who would like a faster way to tune multiple guitars for their students.

There are very few integrated solutions to automatic guitar tuning in the market as of today. Most solutions found in research are merely proof of concept and were never marketed. The main solution found on the market is called the Roadie Tuner which has moderate reviews on retail websites. The device mainly functions through use of an external application which requires a smart phone to operate. This device is also expensive and has a learning curve to use. We would like to create a system which allows the user to use all the main operating modes without an external application while also providing more functionality through an application if desired. The system will have a display to provide menus and basic feedback, while providing multiple buttons with intuitive control directions given on screen.

One of the most difficult aspects of this project is the integration of frequency sensing and physical tuning. There are many ways to extract the core frequency of a musical instrument which range in cost of application. The control system behind the tuning servo also needs to be as fast as possible, while still retaining accuracy to ensure ease of use. A microcontroller will be used in conjunction with sensors, ADC converters, motor drivers, and the OLED display to provide end results to the user. The device will also feature a rechargeable battery system which will cut costs in batteries when the device is used heavily. The device will charge off a simple 5V power supply and handle all battery management internally.

This system will be low cost, accurate, and easy to use. It will provide features to be used by beginners and professionals alike. Along with its utility in quickly tuning guitars, the device will also be useful as a learning aid to people who want to improve their understanding of each note and how to manually hear the small differences in the sound.

2. Project Description

This project is complicated and requires a large number of subsystems to work together in order to achieve the end goal. With this in mind, the project is described in a broad sense in the following sections in order to better narrow down the goals, design requirements, and break down the overall system into separate subsystems.

2.1. Project Background

In the musical industry, manually tuning an instrument is a large hurdle to the mastery of the instrument. Gaining the proverbial "ear" requires references and training to achieve. Most references offer an example of frequency to the beginner, but do not account for the characteristic timbre of the specific instrument. Timbre can be explained by observing the latent frequencies which vary on an instrument to instrument basis. While the carrier frequency of the string might be spot on, the additional vibrations and reverberations in other parts the instrument combine to produce the unique tone and timbre.

Another one of main difficulties faced by a beginner is the lingo that goes along with the subject. A new musician might have difficulty telling certain tones apart and just listening to the raw frequency does not correlate due to the previously described timbre. This device would give these users a precise and reliable way to tune their own instrument and therefore learn according to the real thing instead of pre-recorded examples. Professionals, teachers, students, and casual guitar players could all benefit from this device and through different features. The low cost of entry should make it a no-brainer to purchase this product and any future iterations.

2.2. Objectives

This project has two main purposes: It aims to give experience in the full design process of a product, as well as produce a working design which can be utilized by other people. This project takes motivation and dedication to achieve and the progress made in this Spring 2017 semester will be important for a final design in the Fall 2017 semester.

2.2.1. Motivation

The motivation for this project is to apply the skills learned throughout the collective group's stay at the University of Central Florida (UCF). This project combines technical knowledge and the ability to research new material towards rapid application in a design project. This project is a requirement for graduation and is strictly graded according to a given rubric. Successful completion of this class and project proves to future employers that the individual members of this group can successfully work together to produce an end product with strict milestones and schedule.

The Automatic Guitar Tuner will be a handheld device with the ability to tune a guitar. This product will aid people who are unable to or have difficulty tuning a guitar by hand. It will also help the musician calibrate their "ear" to hear if the guitar is in tune or not. It will be easy and intuitive to use, no matter the skill level of the user.

The tuner will have a mechanism that allows it to pick up the frequency produced when the guitar string is plucked. The mechanism will then send this information to the microcontroller unit (MCU). The MCU will communicate with a phone application designed for this purpose. The application will provide the MCU with the desired in-tune note. The MCU would then use this information and the frequency feedback to spin the servo and in turn tune the guitar.
The application will be the user interface for the device itself. It will allow the user to designate which note they want tuned, while also providing the user with a history of previous tuning measurements. The measurement history could be used to train the human ear into hearing if a note is out of tune or not. The application will also allow the user to completely unstring or restring the guitar without hassle, by using a series of button presses. The guitar tuner will also have an organic light emitting diode display (OLED) which will allow the user to forgo the phone application and perform all basic functions.

An example usage case would be this: The musician decides to restring their guitar. They open the application on their phone and set the device to the "Remove String" option. The device is held against the tuning knob and a button on the tuner will be pressed and held down. The device will continuously rotate the knob, loosening the string until it can be removed. The musician then changes the setting in the application to "Install String" option. The musician places the new string in the correct position and presses the button again. This time, however, it will spin only a certain number of rotations. This will ensure that the device does not over tighten the string and damage the guitar. Once it stops rotating, the application will automatically set itself into tuning mode and request input for which setting the guitar should be tuned to. Once chosen, the guitarist will then pluck the string and the Automatic Guitar Tuner will tune the guitar.

The device will be handheld due to the multiple configurations of guitar headstocks. It will decrease the risk of damage to the guitar and the risk of affecting the sound quality of the guitar itself. This also allows the device to be smaller and lighter.

This device is essentially 3 devices in one: a tuner, a guitar peg winder, and a teaching tool. It will allow people of all ages to continue enjoying their hobby without the hassle and pain of restringing and tuning their guitars. It will also help musicians who are interested in developing their "ear" to hear if their guitar is in tune or not.

2.2.2. Design Goals

The goal of this project is to design a device which can automatically tune any stringed instrument. The device has three main areas of focus which require electrical and computer engineering ingenuity to complete:

- The device will actively process the audio signal input to extract the carrier frequency of the musical instrument string.
- The device will have a fully enclosed battery and recharging system which can be charged off the commonly found micro-USB cell phone charger.
- The device will be capable of operating as a standalone unit while also providing expanded functionality when paired to an Android smartphone over a wireless communication protocol.

2.3. Requirements Specifications

The requirement specifications are a physical or functional need that our design must be able to perform. The requirement specifications below is the list of all of the technical and physical aspects that our project needs to meet for ultimate functionality.

Weight: The device will weigh no more than 0.5 kg.
Durability: The device will be capable of surviving a 1 m drop.
Packaging: Device will house the electronics in a purpose-built ABS or Nylon plastic housing.
Reliability: The device will have a service life ≥ 1 year
Affordability: The device will cost no more than 75 dollars
Torque: The device will use a continuous rotation electric motor with enough torque to tune ≥ 95% of all guitars.
Low Noise: The device will have a low Signal-to-noise-Ratio (SnR) microphone
Accuracy: The device will have a precision of tune of ±2 percent frequency.
Thermal Management: The tuner will be capable of continuous operation without risk of overheating at room temperature (25℃)
Calibration: The tuner will maintain calibration for duration of ≥ 1 year.
Speed: The tuner will complete tuning cycle in ≤ 10 sec.
Dimensions: The tuner shall fit in a package ≤15x7.5x5cm.
Memory: The tuner will have onboard flash storage to save previous tuning frequencies.
Charge Time: The battery will be capable of charging from 10% to 90% in three hours when using an adequate power supply input.
Battery Protection: The battery charging circuit will prevent overvoltage, undervoltage, overcurrent, and overtemperature conditions of battery cells.
Battery Management: The battery circuit will implement a battery management system to mitigate damage to battery cells.
Power Distribution: The regulation circuit will provide adequate power rails for motor, display, microcontroller, amplifier, and ADC circuitry from the same battery.
Voltage Ripple: The regulator circuit will provide less than 100 mV ripple to the motor, and less than 20 mV ripple to the logic components.
Frequency Processing: The microcontroller will be capable extracting the carrier frequency of the musical note in < 200ms.
User Input: The device will provide buttons and adequate software to allow basic functionality in the standalone package.

Power Source Standard: The battery will be capable of charging off a common 5V micro USB power supply.
Display Brightness: The LED display will have a brightness of greater than or equal to 200 nits.
Audio Detection Bandwidth: The system will be able to detect sound waves with frequencies from 30 Hz to 4400 Hz at minimum.
LED Power Consumption: The LED display will consume no more than 0.5W.
LED Resolution: The LED display resolution will be no less than 50 pixels per inch (ppi).
Leakage Current: The tuner will produce a leakage current of no more than 0.75mA.
Range: The tuner will be wireless accessible via Bluetooth to within 15 meters.
Latency: Communication between the tuner and Android application will have an average latency of no more than 225ms.
Application Response Time: The application's response time to touch input will be no longer than 200ms.
Motor Drive: The microcontroller will be capable of communicating with a motor driver circuit to perform tuning.
Logic Communication: The microcontroller will use serial data lines to communicate with external logic circuits.

2.4. System Block Diagram

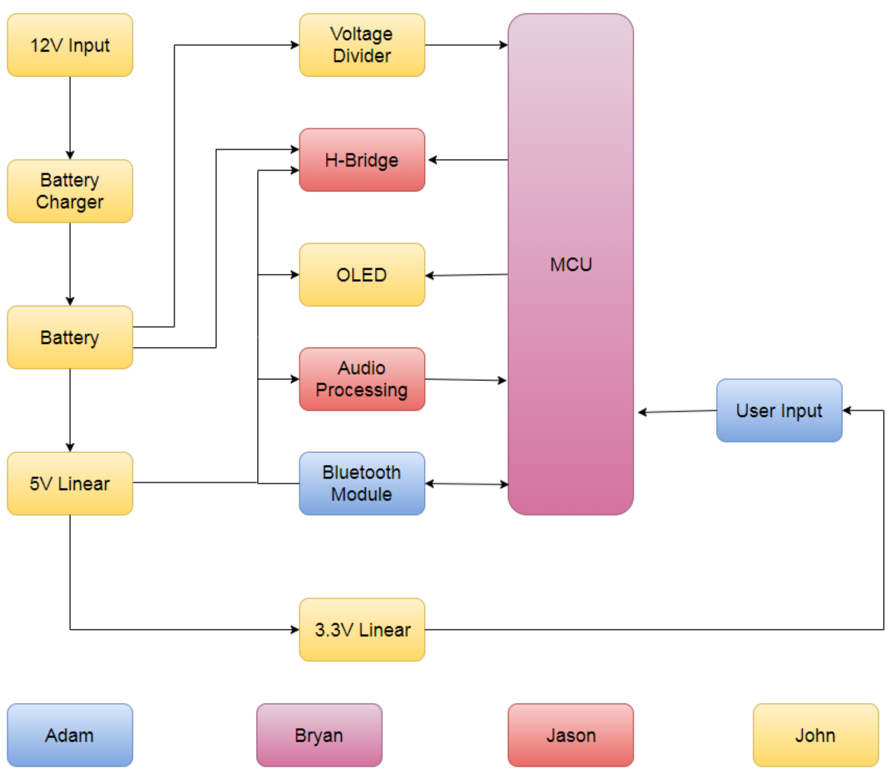


Figure 1: System Block Diagram

With the above requirements listed out, a system block diagram was created to organize the overall system into individual subsystems which could then be designed piecemeal. These systems all connect to form a cohesive system which will achieve all the goals set forth. The block diagram is also color coded to credit the person responsible for the bulk of the subsystem development. The overall device is split into seven individual subsystems, each with their own unique challenges and design requirements. They are as follows; Master Control Unit, Bluetooth Communication, Sound Feedback, Servo and Driver, Rechargeable Battery and Power Regulation, Display, and User Input.

The master control unit is the microcontroller which ties the entire system together. All subsystems interact or communicate with the MCU in some way, which is why it is placed in the center of the figure. The lone CpE in the group is responsible for selection of microcontroller along with software/firmware which handles all peripheral communication, signal processing, and control signaling.

The Bluetooth communication module handles all wireless communication with the external application. This subsystem is important for providing extended functionality through interaction with the user's phone along with giving better feedback when the device is used for training.

The sound feedback module gives sound feedback to the MCU when the tuning is in progress. This subsystem includes a condenser microphone, active filter, amplifier, and Analog to Digital Converter, which sends a digital sample of the guitar sound back to the MCU for processing. The design of this system needs to be sufficient to provide a clean enough signal for frequency extraction and is a challenging analog circuit to implement.

The servo system is the only moving part in this device. The servo handles physical turning of guitar knobs to bring each individual string into tune. The servo system needs to output sufficient torque to turn the hardest knob, while also being fast enough to provide a tuning cycle time less than listed in the requirements.

The rechargeable battery and power regulation module provides lasting power to all components within the device. This system consists of a single lithium ion battery cell, a linear constant current/constant voltage charging circuit, battery fuel gauge, a short circuit protection circuit, a buck converter, and a pair of linear regulators. This system provides an efficient, reliable, and integrated power delivery scheme which is important for the end usage of the device. The user can simply plug in a common micro-USB phone charger which will keep the device going up for up to 250 charging cycles.

The display module of this device provides a clear and bright picture with good contrast. A display is important for offering standalone capability to the user in the situation where a phone is unavailable. Without the display, the device would be unusable without an expensive smartphone. The display is sufficiently sized to provide adequately sized menus while being small enough to fit on the body of the device.

The final section, user input, works in tandem with the display to offer standalone functionality. These buttons are tactile and clicky, offering a reassurance to the user that the signal is sent to the microcontroller. The buttons also offer an intelligent menu navigation system which is quickly learned and utilized. There is also a switch to allow easy power down of the system.

Overall, this project offers many unique challenges in the areas of electrical and computer engineering. The challenges range over analog filtering, digital sampling and processing, power systems, mechatronic control, and wireless communication. Each of the systems designed in this Spring 2017 semester will be further integrated together in the Fall 2017 semester to form a completed Automatic Guitar Tuning device.

2.5. Marketing and Engineering Requirements

This section provides a figure detailing the project's requirements. Many requirements are dependent on one another and defined by a negative or positive correlation with each other. There were chosen four particular marketing requirements and five engineering requirements for comparison. Marketing requirements are generalized features of the device which appeal to the average consumer. Engineering requirements are technical specifications which are necessary for an adequately performing system. Each marketing requirement is rated against each engineering requirement, and then each engineering requirement is rated against the other engineering requirements. After rating each requirement against the next, five technical specifications were chosen which should meet the need of the product. For more information, see the key below the figure.

The Marketing requirements cover four different areas as follows: Ease of use denotes the simplicity of using the device for the first time for beginners. Functionality is the practical features the device offers to the user. Battery life limits how long the user can tune with the device and needs to be high. Retail cost is the barrier to entry for most people, since automatic guitar tuners are not the standard in the industry. The device needs to be good enough to warrant the cost.

The engineering requirements cover the five different areas as follows: Accuracy is the most important detail and needs to provide better results than possible by manual methods. Speed needs to be high to negate user error and patience. The size of the package should be small enough to fit the average human hand to reach the largest segment of the market. Weight should be kept as low as possible to make it easier to hold the device for up to a minute at a time while tuning multiple strings. Cost should be kept as low as possible to minimize retail costs and reach the broadest market possible.

Up arrow = Positive Correlation
Double Up arrow = Strong Positive Correlation
Down arrow = Negative Correlation
Double Down arrow = Strong Negative Correlation
+ = Increase requirement
- = Decrease requirement

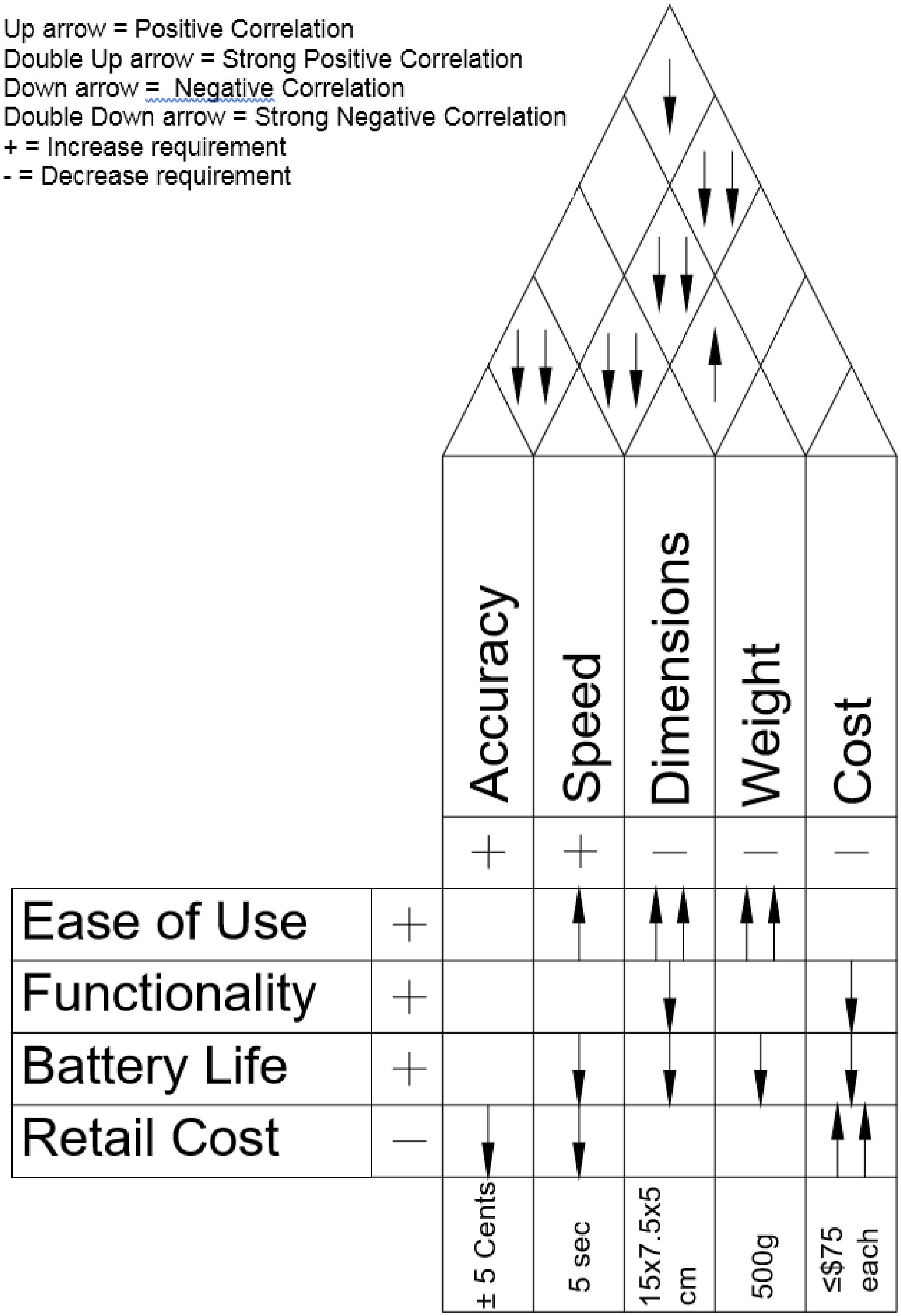


Figure 2: House of Quality

3. Research Related to Project Definition

The market for guitar tuners was carefully examined and documented in the sections below. The underlying technology needed for creating an automatic guitar tuner was also examined and compared.

3.1. Existing Projects and Products

Engineering is built on modifying existing products and projects to create a better product. The section below will discuss existing products and projects that influenced our design.

3.1.1. Korg CA-40 Electronic Chromatic Tuner

Figure 3: Korg CA-40

The Korg CA-40 Electronic Chromatic Tuner is a very popular product with musicians. It has two input methods, a port for an electric guitar or bass and a microphone for all the other instruments. Another feature this tuner has is that with electric guitars and basses, it has an output jack that allows the signal to continue to the amplifier. This allows the user to constantly tune while still performing.

This tuner can detect notes A0-C8 which translates to 27.5 Hz - 4186 Hz. This range allows it to be used not just for stringed instruments, but almost every single instrument that is played. The musician plays a note on their instrument and the tuner picks it up and displays what note was played and how far sharp or flat that note was. The musician can manipulate their instrument in real time to see the effect on the note to get in "tune". This tuner also allows for experienced musicians the ability to calibrate the tuner to be slightly out of “tune” but sound better with the entire piece. This is because music is not mathematically perfect and being slightly out of tune can have a positive affect.

The tuner design is also significant. It is designed in a way that allows for 85 hours of continuous use. However it also has the ability to autonomously turn itself off if it has performed an operation in over 20 minutes. This will extend the battery life quite a bit.

This tuner is a fantastic option for any musician, however it does not autonomously tune the instruments themselves. This means that for some musicians, the act of tuning their guitars could be painful or very confusing. The musician needs to know how the turn the tuning knobs when the tuner indicates the note being sharp or flat. The musician also needs to know what the standard tunings are. All these issues could be solved with a combination of an autonomous tuner with an application that provides some basic information about the instrument being played.

3.1.2. Roadie Tuner

Figure 4: Roadie Tuner

The Roadie Tuner is the only competing design currently on the market. It features a handheld design which attaches to each individual knob and tunes it according to the user chosen frequency and feedback from strumming the string.

The Roadie Tuner moves all user interface to its phone application which is available for both IOS and Android. This makes it accessible to almost all customers since the target demographic overlaps with the people who own IOS and Android smartphones.

The Roadie tuner has a detection range from 55 Hz to 880 Hz which allows it to reliably tune any stringed instrument with notes in that range including 6-7-12 string guitars, ukulele, and banjos, as long as the clip fits on the peg. The Roadie Tuner application also allows for electric guitars to be plugged directly into the smartphone, so the sound feedback can avoid external noise. The included microphone on this device implements active noise canceling to allow tuning in noisy environments.

The Roadie Tuner appears to do all sound processing on the smartphone by transmitting the raw signal sample to the phone over Bluetooth. The application then crunches the numbers and sends the required turning output to tune to the correct frequency. This system has shown to have a precision of 0.1 Hz and can tune to +/- 1 cent. The application allows for custom profiles to be saved which gives the user a great amount of freedom in how their instrument is tuned.

This device was funded through a Kickstarter crowdsourcing campaign. $178,613 was raised by 2,002 contributors to see this device built and brought to market. The device is currently available on Amazon and has four out of five stars as its average review. People appear to be very satisfied with how the device works. They state that the learning curve is a little sharp, but once mastered, the device works as intended and can be used over and over with the same amount of precision.

3.1.3. Automatic Guitar Tuner - UCF EECS Senior Design Fall 2014

The project performed in Fall 2014 is much more of a proof of concept design than the Roadie Tuner discussed previously. This project aims to create a system which automatically tunes the five strings of a bass guitar. The project attaches to the neck of the guitar where the tuning motors are permanently bound to the tuning knobs. The user is still required to pluck the string of the desired tuning knob.

The Automatic Guitar Tuner had an Android application which allowed the user to select the desired tuning preference. The application then sent this data to the microcontroller via Bluetooth, then the microcontroller took over. The user plucks the string as commanded by the application, the microcontroller uses pickups to sample the sound, extract the underlying frequency, and command the motor to turn until the string is calibrated to the set note. The application allows for storage of commonly used profiles for each string and can tune to each of them with the press of a button.

This project utilizes the MSP430F5529 MCU to perform most of its signal processing and motor control. The MSP430 uses a separate Bluetooth IC to perform the communication between application and device. The user can select frequencies for each string either from a set of pre-stored values, or they can enter the desired value in manually. The system uses magnetic pickups which were salvaged from an old electric guitar. These magnetic pickups work by detecting the changing magnetic field around a vibrating string and translating it into an electric signal. The signal is then amplified and converted to a digital signal by an analog to digital converter (ADC). After taking a sufficient number of samples, a Fast Fourier Transform (FFT) is performed and the carrier frequency extracted. The error is calculated and sent through a motor controller to turn the peg in the correct direction until the error falls below a chosen threshold.

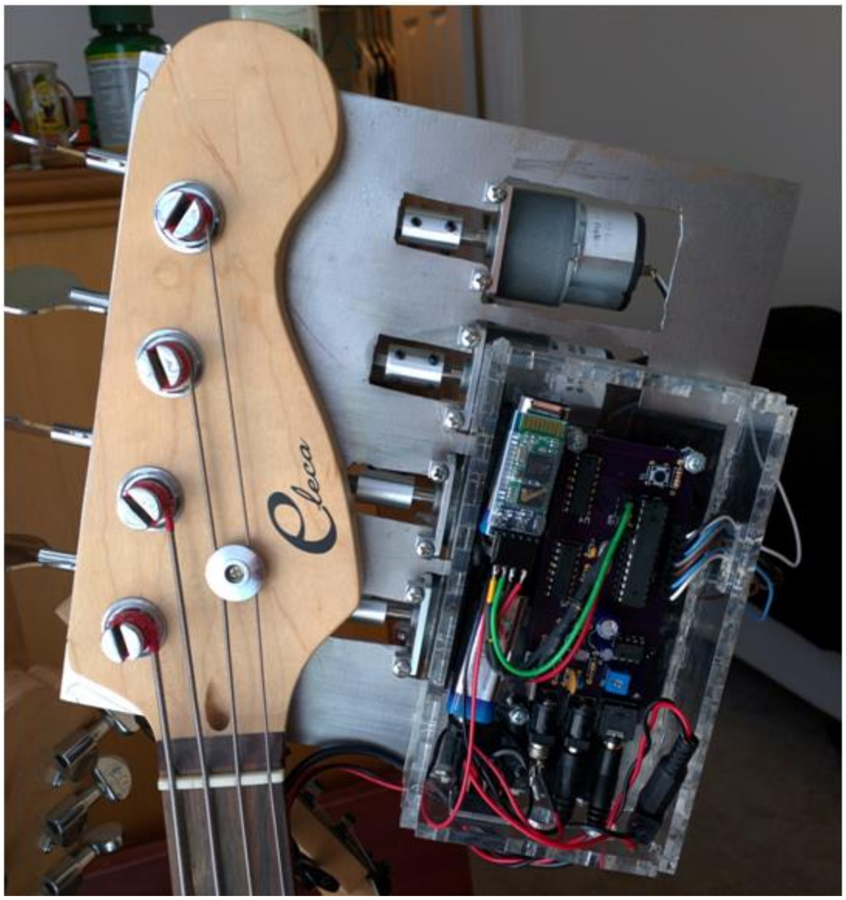


Figure 5: Automatic Guitar Tuner

While this project seems to perform adequately and provides a good amount of usability, the overall implementation of the project could be more streamlined. The project is permanently attached to the neck of the guitar, which would impede actual playing of the instrument after tuning. The project also lacks a battery charging option on board which would make it a hassle if the battery runs out during a critical stage. This project provides great reference points, but the end implementation is vastly different from the goal of our product.

3.2. Relevant Technologies

The following section is a discussion of all relevant technologies involved in this project. Detailed definitions and descriptions of each technology are provided. In cases where a technology may cover a multitude of categories, each category is compared with each other to provide the advantages and disadvantages of each.

3.2.1. Battery

Batteries are devices which use chemicals to store energy for later use. Most batteries have two terminals, the cathode and anode, which carry a potential voltage across them. When a circuit is created between these two terminals, electrons tend to flow from the negative terminal to the positive terminal. This movement of charge is known as current, measured in Amperes, and the product of current and voltage is known as power, measured in Watts. There are various configurations of battery cells to be used depending on the requirements of the device being powered. Batteries are typically split into two groups: disposable batteries, and rechargeable batteries. These two technologies are compared in the table below.

Table 1: Battery Technology Comparison

Feature	Disposable Battery	Rechargeable Battery
Technology	Alkaline	Lithium Ion
Reusable	No	Yes
Safety	High	Moderate
Minimum Voltage	Low	Medium
Shelf Life	3 Years	2 Years
Initial Cost	Low	High
Cost Over Time	High	Low
Environmentally Friendly	No	Yes

Disposable batteries are typically used in devices which have low power draw and do not need to be replaced nearly as often, like a TV remote. A common disposable battery chemistry is known on the market as “Alkaline”, which has a magnesium dioxide cathode and zinc anode with a potassium hydroxide electrolyte. These cells can be drained to very low voltages with no worry of damage because they are disposable. Oftentimes after discharging to low values, the voltage will recover if left alone for a short period of time. This configuration has a small internal resistance and a long shelf life, upwards of three years. The only downside is the batteries must be disposed of after running out of energy, which adds waste to the environment and costs money to replace.

Rechargeable batteries are more commonly used in devices with shorter run cycles and higher power draw, like an autonomous vehicle. A common rechargeable battery chemistry is known as “Lithium Ion” technology which comes in a few different configurations of cathode and anode. Rechargeable cells usually have a cutoff voltage where permanent damage can be done if drained any lower. Lithium Ion cells have a small risk of catastrophic failure when strict operating parameters are not maintained. Catastrophic failure includes rapid temperature rise, fire, and even explosions, which is dangerous for the end user. A proper battery charging circuit and battery management system can reduce this risk and best implemented with safety in mind. These cells have long shelf lives and can maintain a charge for months at a time. Cells can commonly be charged up to 1000 times, which saves money and wastes less resources.

3.2.2. Battery Charging Integrated Circuit (IC)

Lithium Ion batteries require controlled parameters for recharging safely and efficiently. The commonly accepted method involves a constant current segment and a constant voltage segment. The constant current segment charges the battery at a set number of amperes according to the datasheet of the cell. As the battery voltage rises and reaches a certain limit, the charger switches to a constant voltage mode which tops off the battery until a chosen cutoff current is reached.

Most battery charging ICs contain various safety circuits built in. They will protect against overvoltage, under voltage, overcurrent, and high temperature breakdown. Some battery charging circuits will even detect an over-drained cell use proven methods to revive the cell with a minimum amount of damage. These features are all important parts of creating a robust battery configuration and play heavily into the selection of charging IC.

3.2.3. Battery Management System (BMS)

When Lithium Ion cells are used, it is good practice to implement a BMS. This is a circuit which monitors the state of the cells and ensures the harmony of the cell pack. The BMS protects against short circuit, under voltage, overvoltage, overcurrent, and overtemperature conditions. Each of these protections are important to protect the integrity of the battery cell, which also protects the user from the dangers of cell failure.

A battery fuel gauge IC can also be used to monitor the current output and voltage of the battery cell, and is an important part of a battery management system. Using this data, it can return a 0-100% battery value to the MCU which can then show this on the display. This value can also be used to protect from overdrain scenarios, so the MCU can shut down before overdraining the cell.

3.2.4. Display

A display is used to convey visual information to the user of the device. The two most common technologies for this task are Liquid Crystal Displays (LCD) and Organic Light Emitting Diode (OLED) displays.

LCD displays use a matrix of crystals suspended in liquid which can be manipulated using electronic control. The most common crystals used are called Twisted Nematic (TN) and these crystals either let light through, or hinder light depending on how they are electrically stimulated. Using this in conjunction with an array of LED backlights, characters and figures can be displayed. TN panels usually have decent contrast and good response times while maintaining low cost.

	LCD/LED	OLED
Lighting Uniformity		X
Brightness	X	
Local Dimming & Contrast		X
Burn-In	Tie	Tie
Resolution	Tie	Tie
Expanded Color Gamut		X
Viewing Angle		X
Energy Consumption		X
High Dynamic Range (HDR)	Tie	Tie
Refresh Rate / Motion Blur		X
Lifespan	X	
Price	X	

Figure 6: LCD/LED vs OLED

An OLED display uses a matrix of light emitting diodes as pixels to form text and figures in a viewable form. OLED displays have the advantage of being high contrast and energy efficient. These features are due to the characteristic method in which the OLEDs work. When the pixel is black, the OLED is off. When this creates near infinite white to black contrast and uses no power when a black background is used.

3.2.5. Microphone

Microphones are devices which translate sound into an electrical signal. These components are used in everything from cellphones to deep sea observatory projects and help to record sound as a picture records a sight.

Table 2: Microphone Comparison

Feature	Condensor Mic	Dynamic Mic
Diaphragm Technology	Capacitor	Inductor
Active Power	Yes	No
Frequency Response	Flat	Quadratic
Size	Small	Medium

One of the most common microphones is the condenser microphone. When the sound wave hits the diaphragm it begins to oscillate with the same frequency as the sound wave. This diaphragm is apart of a capacitor, so as it oscillates it changes the capacitance. This change in capacitance affects the electrical signal, allowing it to have the same frequency as the sound wave that originally hit the diaphragm.

The other common type of microphones is the dynamic microphone. This type of microphone works in a similar manner to the common loudspeaker, except in reverse. Like the condensing microphone, it has a diaphragm that moves when hit with soundwaves. However the difference is that in this microphone, the diaphragm moves a coil of wire around a magnet. This movement causes an induced current through the current. One of the benefits of this type of microphone is that it does not need to be powered. The movement from the soundwave produces the signal.

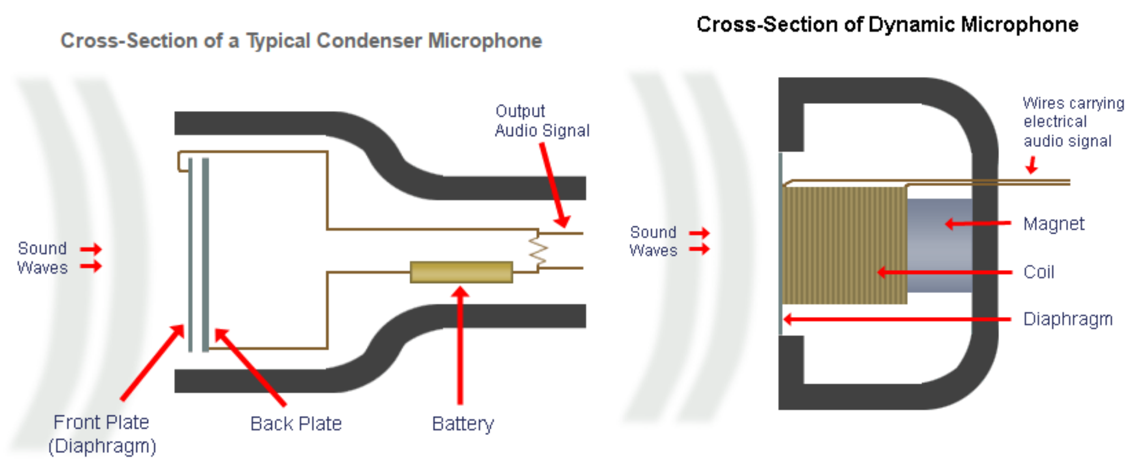
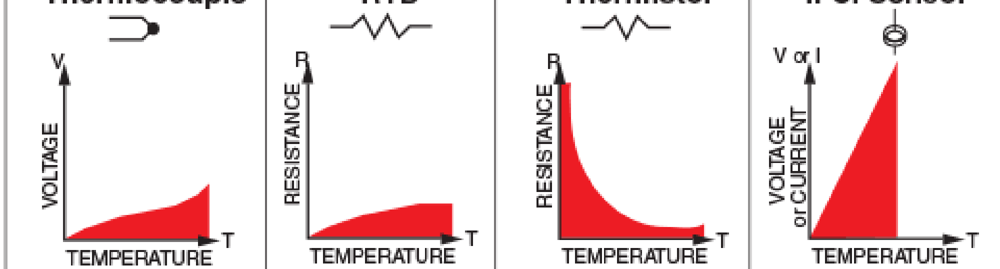


Figure 7: Condenser vs. Dynamic Microphones [15]

Condensing microphones tend to have a more flat frequency response when compared to dynamic microphones. This flat frequency response is desirable for our system in order to ensure that all the output from the microphone is predictable. This predictability allows for easier filter and amplifier designs.

3.2.6. Temperature Sensor

A thermistor is a resistor that varies predictably with temperature. Its resistance value also swings over a wide value, which makes sensing the resistance much easier than typical materials. Using the measured voltage drop and current through a thermistor, the resistance can be calculated. This calculated resistance can then be used to tell the temperature which is used for cold and hot temperature cutoff conditions.



	Thermocouple	RTD	Thermistor	I. C. Sensor
Advantages	Self-powered Simple Rugged Inexpensive Wide variety Wide temperature range	Most stable Most accurate More linear than thermocouple	High output Fast Two-wire ohms measurement	Most linear Highest output Inexpensive
Disadvantages	Non-linear Low voltage Reference required Least stable Least sensitive	Expensive Current source required Small Δ R Low absolute resistance Self-heating	Non-linear Limited temperature range Fragile Current source required Self-heating	T<200°C Power supply required Slow Self-heating Limited configurations

Figure 8: Temperature Sensor Comparison [16]

Another device which can be used to sense temperature is a thermocouple. Thermocouples use dissimilar metals to produce a small voltage which varies with temperature at the junction. This small voltage can be amplified and correlated to an output temperature.

Thermocouples are usually more accurate than thermistors due to their reliance on the physical interaction between materials, rather than the size and shape of the material used in a thermistor. Thermocouples, however, require more complicated circuitry to amplify the low voltage signal to something readable. A thermistor circuit can be as simple as a voltage divider being read by a ADC and often can be used in a standalone configuration as in input to an integrated circuit.

3.2.7. Switches

Ever since the discovery of electricity, there has been a need to control it. Switches are used to manually create and break connections in a circuit. These can be used to switch power lines on/off, or communicate with a microcontroller to provide user input to an intelligent device. As such, there are many different features of switches which need to be chosen carefully for the specific application.

One of the main characteristics to decide between is whether the switch needs to be momentary or toggle. A momentary switch has automatically resets to its default state after being activated and is useful for communication with a microcontroller. A toggle switch will remain in the selected state and does not have an intrinsic default state. This is useful as a power switch which can be used to select on and off states of the device.

Another characteristic to pay attention to when using a momentary switch is its default state and active state. Momentary switches can either be normally open (infinite series resistance) or normally closed (zero series resistance). Depending on the application, both of these switches are commonly used in handheld devices. A normally open switch signaling circuit has the benefit of using zero power in its default state, so it is commonly used in efficient devices. The only thing to keep track of is the microcontroller will read high as a default, and low when the switch is activated.

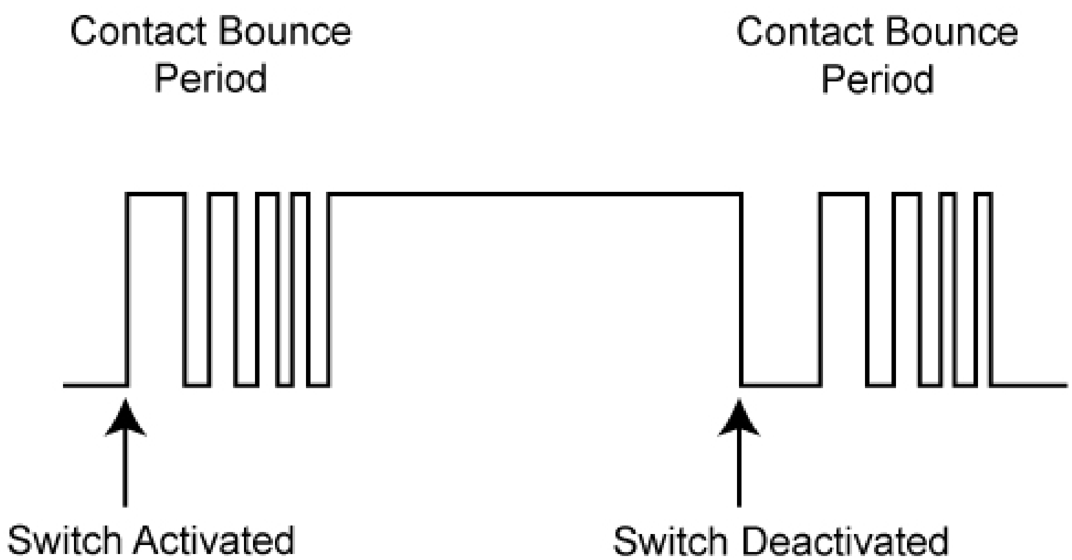


Figure 9: Switch Bounce [17]

While ideal switches are assumed to instantly change state, the real application sees a certain amount of noise known as "bounce". Bounce is what happens when a spring-loaded contact settles down and can flick on and off very fast. Due to the very fast operating speed of modern microcontrollers, these pulses can be read as individual button presses, so a certain amount of care needs to be taken to "debounce" the input circuit. This can be done in a variety of ways, but one of the more simple methods is to use a combination of capacitor and resistor to smooth the voltage at the microcontroller sense pin. There is also a method of using high and low voltage thresholds called a “Schmitt Trigger” which allows a microcontroller to put separate limits on high-to-low and low-to-high transitions.

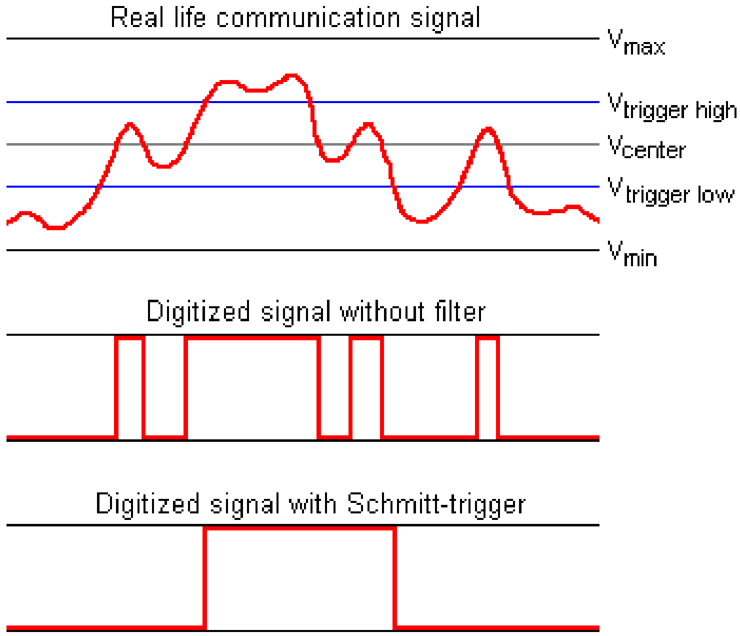


Figure 10: Schmitt Trigger [18]

3.2.8. Transistors

Transistors are the most influential invention of the 20th century. These are clever circuits which allow for power to be switched electronically, rather than manually. These transistors can also be used to create logic circuits and have evolved into the driving force behind modern computers. Transistors can be found in all modern technology, and as such, there are a few different technologies to discuss. There are two main transistors which have been proven useful in modern technology; the BJT, and MOSFET.

Table 3: Transistor Comparison

Features	BJT	MOSFET
Amplification	Good	Moderate
Switching Speed	High	High
Efficiency	Moderate	High
Rd(on)	Medium	Low
Switching Characteristic	Current Controlled	Voltage Controlled

BJT stands for Bipolar Junction Transistor, and can be used in either active mode, or saturation mode depending on its desired function. Active mode allows for amplification of a DC signal and can be used in high frequency amplifier circuits. Saturation mode allows the BJT to act as a digital switch to be on or off. BJTs are often more useful for higher frequency amplification circuits, where an Operational Amplifier could not keep up.

MOSFET stands for Metal Oxide Semiconductor Field Effect Transistor. MOSFETs are typically used in saturation mode, where they can be turned on and off by application of voltage from the gate to the drain. MOSFETs can also be used to switch very quickly, which can be used to control high power things from a low power control circuit.

MOSFETs have become the default choice for many circuits where a fast switching speed and high power switch is needed. MOSFETs can also be used to simulate analog signals by using pulse width modulation (PWM) to control the average power being delivered to a circuit.

3.2.9. H-Bridge

One specific circuit which can be created using MOSFETs is an H bridge. This is a circuit used to selectively drive DC motors in forward or reverse directions. It does this by switching the polarity of applied voltage on the DC motor terminals, which in turn, controls the direction of magnetic field which drives the rotor. The MOSFETs each have drop some voltage which lowers the output voltage a small amount. This should be accounted for by running the input voltage higher than the motor needs. Most common H-bridge ICs include helpful features like freewheeling diodes to ease the transition from switching directions. Freewheeling diodes work to prevent damage to the internal MOSFETs by allowing current from motor inductors to gradually change rather than forcing a high voltage switch. This protection is vital when controlling a motor with the signals of a sensitive microcontroller.

A more complicated H-bridge circuit is used to control stepper motors. This circuit includes two H-bridges which can be used in conjunction together. A two-pole stepper motor needs simultaneous control of two poles in order to properly traverse between steps, which are measured discrete amounts of rotation according to the number of coils in the motor. A dedicated stepper motor controller simplifies the control signal input by requiring only three digital signals; enable, direction, and step. The enable pin turns on the stepper driver and locks the rotor in place. The direction pin chooses either clockwise or counterclockwise stepping motion. The step pin chooses when steps are performed. If more precision is needed, the direction and step pin can be cycled very quickly to perform something called microsteps, which bounces the rotor between two discrete steps.

3.2.10. Amplifier and Active Filter

The amplifier will increase the voltage of the microphone signal to the active filter. Most commonly used amplifier circuits use operational amplifiers. Operational amplifiers are a classification of circuit which offers high input impedance, low output impedance, and can output very closely to their input rails. An operational amplifier works by taking the voltage difference between its noninverting input and inverting input, multiplying it by its open loop gain, and sending that value to its output. Operational amplifiers can also be used in a feedback mode, where the output voltage can be routed back to one of the inputs in order to provide a controlled amplification factor. The ability to provide feedback from output to input greatly eased the difficulty factor of many commonly used circuits, such as active filters.

An active filter is an electrical circuit that filters signals within desired range. This filter also allows the signal to be amplified as it goes through the filter eliminating the need to have an amplifier later in the device. The filter is constructed with simple electrical devices such as resistors and capacitors as long with operational amplifiers. Another useful feature of certain operational amplifiers is the ability to operate from a single supply. This allows an AC signal input to be boosted to a purely DC voltage, which is necessary for an analog to digital converter to work.

3.2.11. Analog to Digital Converter (ADC)

An analog to digital converter can be used to convert an analog signal to a digital series of digital numbers. The ADC needs to have a sampling rate that is at least twice as fast as the highest input frequency. This will ensure that the waveform will be preserved in the conversion. An ADC works by sensing the input voltage in reference to ground. It then converts this analog number into a binary digit which is sent to the MCU via serial protocol. ADCs can only separate the sensed voltage values into discrete steps due to limited number of binary combinations. More expensive ADCs will send longer binary numbers and, therefore, offer more precision.

3.2.12. DC Voltage Regulator/Converter

DC-DC voltage conversion is required for most products which run on battery power because battery cells have a limited voltage output which varies around the specified nominal voltage. DC converters are often modeled as two port networks, where there is an input voltage node and an output voltage node. The voltage at the input is taken from the positive terminal of the battery cell, sent through a complex circuit, which varies per the desired output. The output can be converted to higher or lower voltages depending on the circuit used. The three main types of DC to DC conversion commonly used are outlined in the table below.

A linear regulator can be used to create a controlled voltage bus usually for sensitive components. Linear regulators are low noise and can be used for sensitive components with narrow input voltages. While providing the previously mentioned benefits, linear regulators are inefficient compared to switching regulators. The best practice for efficiency with linear regulators is minimize the difference between input and output voltage according to the limits of the data sheet. Another downfall of linear regulators is that they can't increase their output voltage higher than that of the input. A boost converter is needed for this function.

A boost converter is used to increase the voltage of a DC source. It does this with a clever arrangement of a power source, MOSFET, and an inductor. A capacitor is also usually put in parallel with the load in order to smooth out the signal. Boost converters are also very efficient, providing values often greater than 80% efficiency when designed correctly. While a boost converter provides the ability to increase voltage, it is still noisier than a typical linear regulator. This makes it a poor choice for powering sensitive components which have narrow input voltage ranges. This switching circuit works such that Vout > Vin.

A buck converter, also known as a switching regulator is very similar to a boost converter, but it is used to decrease the voltage of a DC source. It can be used to regulate any voltage such that Vout < Vin. Buck circuits are very useful for stepping down voltages as efficiently as possible, often obtaining values greater than 90% efficiency. Another useful function of a buck converter is that it can create a negative rail from a positive source with the proper arrangement of components. These three DC-DC voltage regulators provide great flexibility in designing the necessary power lines for each component of this project from a single power source.

Table 4: DC Voltage Regulator Comparison

Converter Type	Linear Regulator	Boost Converter	Buck Converter
Switching or Linear	Linear	Switching	Switching
Efficiency	>40%	>80%	>90%
External Components	None-Few	Few-Medium	Few-medium
Voltage Ripple	Low	High	Medium
Vin Vs Vout	Vin > Vout	Vin < Vout	Vin > Vout

3.2.13. Communication

Electronic communication is split into two main categories; Serial communication, and Parallel communication. Both of these formats have varying levels of complexity and implementation when used in real world devices. The application of communication within this device needs to be analyzed to narrow down the best format of communication.

Serial communication in its base form can communicate in one direction, one bit at a time, thus it only needs one wire for data stream. This communication can be done synchronously, while synched to a clock line, or asynchronously, using start and stop bits to denote beginning and end of data streams. When using a synchronous communication, such as I2C, there is usually a master device and a slave device. The master requests data from the slave address, and the slave responds with the data requested, sending bits at the same time as clock pulses. Universal Asynchronous Receiver/Transmitter (UART) devices are configured to a set clock rate and sample the data line at a higher frequency to discern between bits and eventually form the byte of data. The higher sampling rate negates the need for a clock line, as long as both devices are configured for the same baud rate.

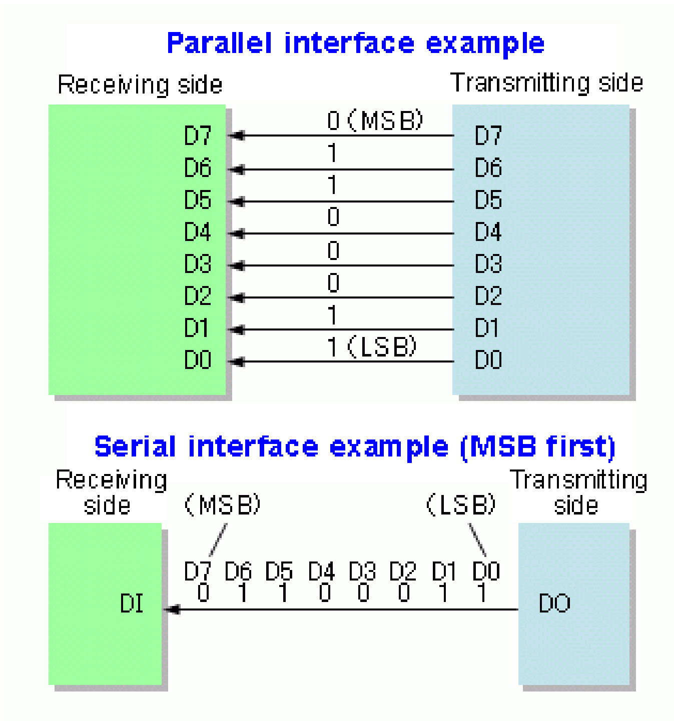


Figure 11: Parallel and Serial Interfacing

Parallel communication works by sending data over multiple data lines at the same time. This is much more complicated than serial communication and requires a much more complicated circuit to ensure all data lines can be read properly. Parallel communication is most often used when single lanes of communication can not provide the required amount of bandwidth for the system. This was much more common in older systems where clock rates were typically much lower. Modern systems are less likely to use parallel communication due to the speed at which newer devices can communicate.

3.2.14. Motors

A motor is an electromechanical device which converts electrical power into mechanical motion. Most motors work by using a combination of inductors and magnets to create an attracting and repelling force. When current is sent through an inductor, it induces a magnetic field perpendicular to the motion of the current, which can be multiplied by wrapping the wire around itself into a coil. Current can be sent through this coil to create a strong magnetic field which interacts with permanent magnets on a rotor. The changing magnetic field can repel or attract the magnet depending on the polarity of the magnet, thus inducing motion of the central rotor. There are many kinds of motors which are considered for this device to work.

Stepper Motor Versus Servo Motor	
Stepper Motor	Servo Motor
Stepper Motor Advantages	**Servo Motor Advantages**
Stability: A stepper motor can drive a wide range of frictional and inertial loads	High output: Power in relation to the servo motor size and weight
Does Not Require Feedback: The stepper motor also acts as the position transducer	Encoder: Determines the accuracy and resolution of the servo motor
Price: Relatively inexpensive	High-efficiency: The servo motor can approach 90% efficiency at light loads
Standardized performance and frame size NEMA standard	High Torque to Inertia Ratio: The servo motor can rapidly accelerate loads
Plug and Play features: Easy to set up and use	A servo motor has 2-3 times continuous power for short periods
Safety: The stepper motor stops if there is a malfunction or interferance	A servo motor has 5-10 times rated torque for short periods
Excellent Low Speed Torque: The stepper motor has the ability to drive several loads without gearing	A servo motor stays cool because the current draw is proportional to load
Repeatability: Accurately returns to the same location - open loop system	A servo motor maintains usable high speed torque of 90% of NL RPM
Overload safe: A mechanical overload cannot damage the stepper motor	A servo motor performs quietly at high speeds silently
Longevity: If the specifications of the motor are not exceeded, the stepper is good for 10,000 hours of operation	A servo motor has a resonance-free and vibration-free operation
Stepper Motor Disadvantages	**Servo Motor Disadvantages**
Low efficiency: A stepper motor draws enough power regardless of the load	Higher relative cost
Torque drops quickly with speed	The servo motor requires "tuning" to stabilize feedback loop
Requires microstepping to move smoothly; prone to resonance problems	Safety circuits are required; motor "runs away" when something breaks
There is no feedback to indicate missed steps	A servo motor is more complex and requires an encoder
Low torque to inertia ratio; cannot accelerate loads very quickly	The life of a servo motor is only 2,000 hours, after that, service is required
In high performance configurations, the stepper motor can get extremely hot (requires heat sink or fan-cooled installation	Peak torque is limited o a 1% duty cycle
Motor does not "pick up" after momentary overload	Sustained overload can damage the servo motor
In moderate to high speeds, the stepper motor can be noisy, which limits its applications	Overwhelming choice of motors, encoders, and servo drivers
Low output power for size and weight	Power supply current 10 times average to use peak torque
	The servo motor develops peak power at higher speeds; gearing is often needed
	Poor Motor Cooling: Ventilated servo motors are easily contaminated

Figure 12: Stepper Motor vs. Servo Motor

A continuous rotation servo is the first consideration. This is a servo motor with the feedback disabled, allowing it to continuously rotate. These motors are small and lightweight, but provide high torque due to their gearing ratio. The servo is controlled using pulse width modulation. The servo usually has a central, “off”, PWM value which denotes no motion. PWM values lower than this value will rotate clockwise, while PWM values higher will rotate counter-clockwise. The difference between “off” and the input PWM signal also control the speed of rotation. This means that the duration of the control pulse and value of PWM determines the angle of rotation over time.

A secondary option is to use a DC motor with a planetary gearbox to lower the speed and increase the torque. This is similar to the servo motor, except an H bridge circuit is required to control direction of rotation, while an encoder could be used to keep track of rotational positioning. An encoder could possibly be negated by tracking the frequency output in real time and only turning the motor while there is error in the signal.

The third option would be to use a stepper motor with a planetary gearbox to increase step precision and torque. Bipolar stepper motors operate using two poles to precisely control the rotation of the rotor in discrete steps. These steps are often as small as 1.8° and when geared down, they can achieve much greater precision without the need of an encoder, assuming no skipped steps. When the stepper motor is enabled, a constant current is being pushed through both coils. This leads to excess power usage when the motor is locked into its specific step. The stepper motor would also require a more specialized driver circuit than an H-bridge, since both coils need to be powered at the same time to maintain step position.

3.2.15. PID Controller

Along with the invention of computers, there has been a need for mathematical control systems to keep a system stable. The most simple system is a proportional system. This type of control works for very simple tasks, but is not sufficient for error-prone systems where accuracy is needed. A proportional-derivative controller can be used to correct for some error, but a proportional-integral-derivative (PID) controller has been tried and true since its implementation in the industrial age.

The proportion component of a PID controller works to eliminate error. The error between current output and goal output is calculated by subtraction, and the output is changed proportionally according to this error. Proportional control can be used for simple systems where there is little hysteresis, but as with most things, real world implementations are usually not simple.

The integral component of a PID controller calculates the accumulated error over time. The longer error exists, the more affect the integral component has on the total output. In this way, the integral component can help mitigate steady state error because the longer the steady state error exists, the bigger the integral component of feedback gets.

The derivative component of a PID controller calculates the error in the rate of change of the output. It does this by calculating the current slope of the output and subtracting it from the expected slope of the output. In a way, this helps to prevent future error. The derivative component helps to reduce overshoot and adds a damping component to the system.

The values for P, I, and D in a PID system need to be tuned together, and are best tuned via specialized algorithms. These algorithms work recursively to find the best combination of PID weighting in order to achieve a stable system.

There is no magical combination of PID numbers which works for all scenarios, and the system is considered tuned when it presents an adequate level of performance while not improving upon further iterations. An example of a recursive table for a PID loop tuning algorithm is seen below.

Table 5: PID Tuning Algorithm

Process	Action
Step 1	Set P, I, and D values to zero.
Step 2	Increase P value until output steadily oscillates around goal
Step 3	Increase D until oscillations decrease to zero, showcasing a critically damped system
Step 4	Repeat steps 2 and 3, increasing the value of P to increase reaction time of system
Step 5	After sufficiently tuning P and D values, increase I value until overshoot is adequate for system requirements

For the purpose of this tuner, the use of PID tuner will be important for controlling the output of the tuning motor. The error in frequency will be devised utilizing a combination of analog and digital processing, and this error will be fed into the PID controller. The PID controller will send output values to the tuning motor which will turn in the proper direction and ideally settle the sound error to zero with minimal overshoot and oscillation. The P, I, and D gain errors will need to be properly tuned to eliminate any ringing or overshoot.

3.2.16. Packaging and Casing

Packaging of a handheld product plays a large part in its usability and desirability to the common consumer. Plastic is the standard for handheld devices due to its durability, impact resistance, and ease of mass production with injection molding. Plastic has also been a staple of the prototyping industry as well with the greater availability of computer numeric control (CNC) machines like 2D laser cutters and 3D fused deposition modeling (FDM) printers.

Almost all plastic packages for mass produced handheld devices are produced through injection molding. The mechanical design is produced in a 3D computer-aided-design (CAD) program and turned into a negative mold. Most of these molds are machined out of steel for its resistance to wear and temperature change. With this in mind, molds commonly cost tens of thousands of dollars, which means the batch size needs to be adequately sized to meet economies of scale. Economies of scale usually mean producing more 10,000+ molds, meaning a customer base needs to be established beforehand.

For the purposes of prototyping, laser cutters provide a very fast turnaround from CAD design to processed parts. For a small project, a laser cutter can process a 2D CAD file in under a minute, after which it falls to the technician to assemble the full model. While laser cutters provide a fast turnaround for CAD to implementation, they are limited in capability of model which can be produced due to the unique thickness of the material and limitation to 2D cuts. This limitation also makes securing edges together difficult, especially without experience in mechanical design.

In the past 5 years, consumer 3D printers have come down in price to where it's finally a feasible option for rapid prototyping on a budget. The plastic is stored as a filament wrapped around a spool, commonly found in 1.75mm and 2.85mm diameters. The plastic is pushed down through the "hot end" where it is heated in excess of 200 ℃, melting it into a thick fluid. These FDM printers work by laying down layer after layer of plastic. The plastic is deposited in slices which are generated by feeding a 3D model into a special "slicer" program. Each of these slices stacks on top of the next to generate a nearly exact replica of the 3D CAD model. This type of 1:1 creation is great for its ability to produce complex curves and 3D contours which would be difficult through any other fashion. The only real downside of 3D printing is the length of time it takes to print. A small print can take upwards of several hours to complete, which makes turnaround time for design to testing much longer than with the laser cutter.

Table 6: Rapid Prototyping Comparison

Features	3D Printing	Laser Cutting	Injection Molding
Material Cost	$15/kg	$15/kg	$4/kg
Design Time	Moderate	Moderate	Long
Dimensional Capability	3D	2D	3D
Machine Time	< 1 Day	< 10 Minutes	< 30 Seconds
Setup Time	< 1 Hour	< 30 Minutes	< 1 Week
Economy of Scale	Small	Medium	High
Startup Cost	Low	Moderate	High

3.2.17. Microcontrollers

A Microcontroller Unit (MCU) is an embedded chip comprised of a microcomputer and an integrated circuit. The microcomputer contains a CPU with one or more cores and is capable of I/O control. Most MCUs come equipped with both RAM and ROM ranging from a single Byte to upwards of 32KB or more. It is common for an MCU to operate within a frequency range of 1-100MHz to limit power consumption but some MCUs approach frequencies rivaling and even surpassing normal CPUs in the GHz range.

MCUs are generally used in embedded systems where small physical size and low power consumption are a necessity. Such devices include home appliances, medical technologies, military weapons such as ballistic missiles, and automobiles. The increasing processing speed and decreasing costs of low power MCUs has made them appropriate for a wide range of needs, and as such, will be used in this project to perform most control and communication tasks.

3.2.18. Musical Instruments

The device will be using the sound waves produced by stringed instruments like guitars and bass guitars to tune the instruments. There are two main types of these stringed instruments, electric and acoustic. The electric instruments have pickups that pick up on the vibrations from the strings and converts it into electrical signals. These signals then need to be amplified from a separate amplifier. This type of instrument is very common for both guitars and bass guitars. The other type of instrument is acoustic. Acoustic instruments are hollow bodied. The chamber allows the vibrations to bounce around and amplify themselves. While resonating in the hollow body, it produces overtones that give the sound certain timbre. There are also some instruments that have the electrical pickups and have a hollow body, making them a hybrid.

Table 7: Standard Guitar Tuning Chart

String (Highest to Lowest)	Pitch Notation	Frequency
1 (E)	E4	329.63 Hz
2 (B)	B3	246.94 Hz
3 (G)	G3	196.00 Hz
4 (D)	D3	146.83 Hz
5 (A)	A2	110.00 Hz
6 (E)	E2	82.41 Hz

The below table lists the standard tuning frequencies of a bass guitar. Note that the frequencies are very low, which adds difficulty to the active amplifier circuit to extract the sound signal from the background noise.

Table 8: Standard Bass Guitar Tuning Chart

String (Highest to Lowest)	Pitch Notation	Frequency
1 (G)	G2	98.00 Hz
2 (D)	D2	73.42 Hz
3 (A)	A1	55.00 Hz
4 (E)	E1	41.20 Hz

Both types of instruments will need to be considered while designing the system. Each one provides a different challenge. With the electric type, since there is no passive amplification, the sound is going to be much more quiet than the acoustic instruments. This can be mitigated with proper amplification of the microphone output signal. With the acoustic and hybrid instruments, while they are much louder, the timbre of the instrument could cause issues when trying to detect the frequency of the sound waves. This issue could be mitigated with proper filtering of the signal.

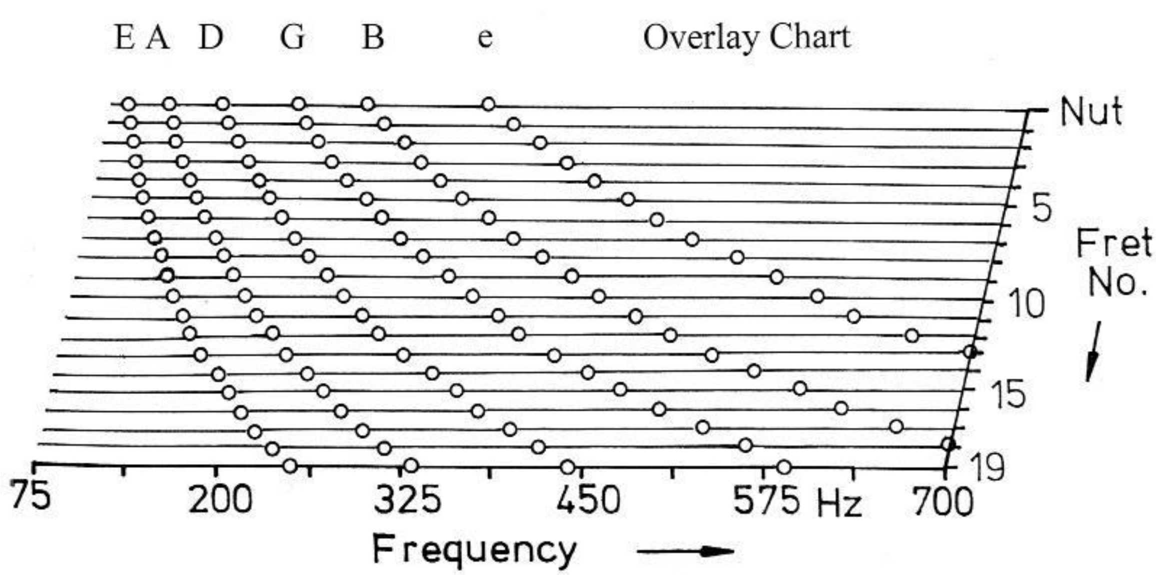


Figure 13: Acoustic Frequency vs Fret Number

3.3. Strategic Components and Part Selections

Table 9: Part Reference Table

Reference	Part
1	Parallax Continuous Rotation Servo
2	LG - Lithium Ion INR18650HG2 3000mAh
2	LG INR18650-HG2 Battery Cell
3	Texas Instruments - LM2700QMT-ADJ/NOPB-600kHz/1.25MHz, 2.5A, Step-up PWM DC/DC Converter
4	STMicroelectronics - LD33V - 3.3V Linear Regulator
5	STMicroelectronics - L7805CV - 5V Linear Regulator
6	Adafruit Monochrome OLED Display
8	Texas Instruments - MSP430G2553 Mixed Signal Microcontroller
9	Mallory Sonalert Products Inc. PMOF-9745W-42UQ Microphone
10	STMicroelectronics TL084CN Operational Amplifier
11	Microchip Technology 579-MCP3221A5T-I/OT ADC
12	Texas Instruments - BQ2057CSN-Low Dropout Linear 1-cell Li-Ion Charge Controller, 4.2V
13	MAXIM - MAX17043G+U Fuel Gauge
14	Adafruit BLE Friend

The below picture lays out all components which were purchased after a rigorous comparison shown in the sections below. The above table names each part according to the numbered index seen in the left column and on the picture. These parts were all tested for their vital parameters and are confirmed in working condition as this group enters Senior Design II in the Fall 2017 semester.

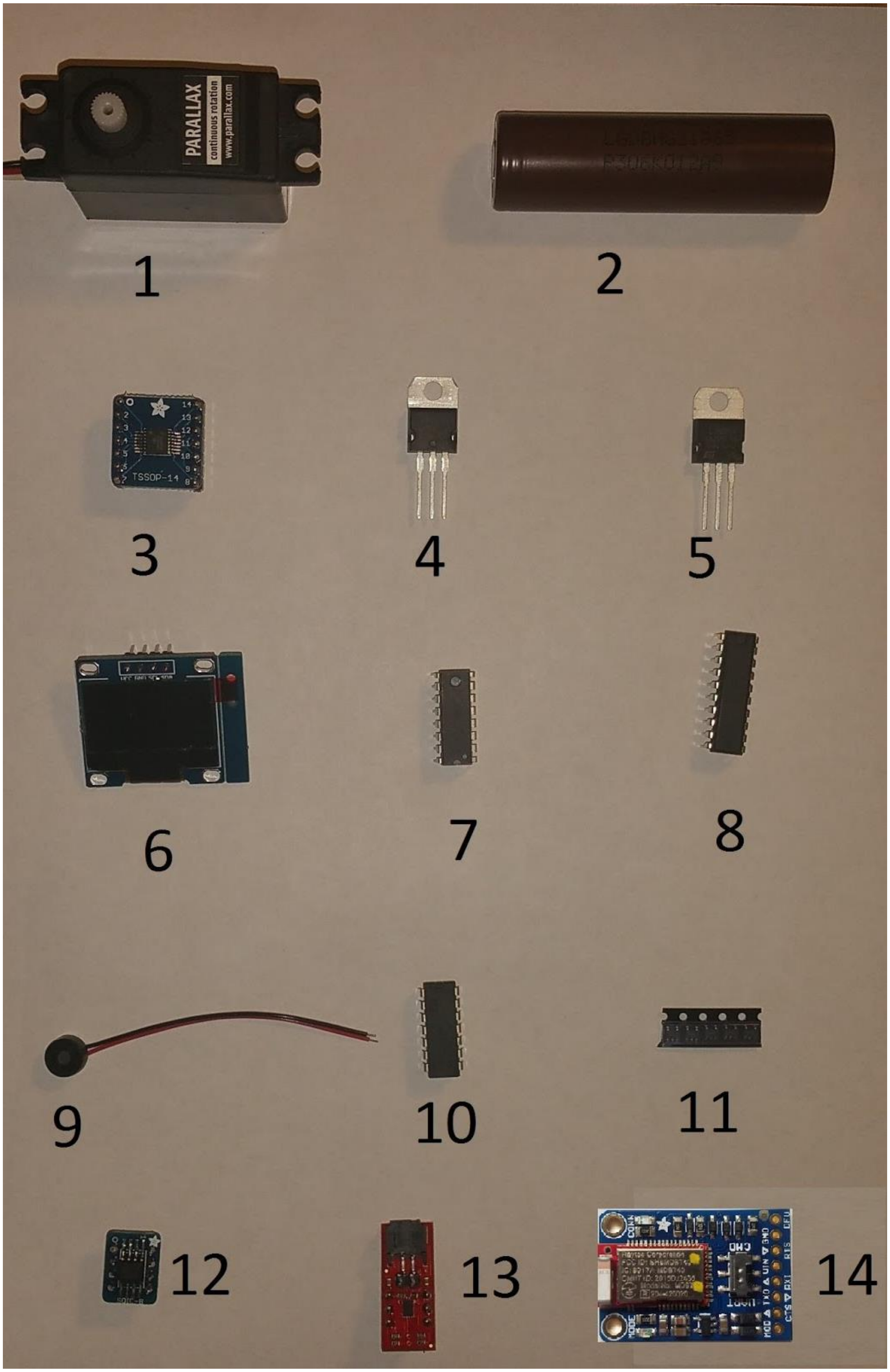


Figure 14: Part Selection

3.3.1. Battery Charging IC

The main requirements for this section are adequate charge current, USB standard input, and temperature monitoring. As such, the two chips were compared.

After comparing these two circuits, the BQ2057CSN chip was chosen for its more feature-rich specification. This chip is a Texas Instruments component which is used to charge a 1S Li-Ion configuration. This chip combines high accuracy current and voltage control, battery conditioning to improve life, temperature monitoring, charge status and termination, and AutoComp charge rate compensation.

This chip, like many others, charges a Lithium Ion battery in two modes; constant current, then constant voltage. The chip also has a third mode called "conditioning" which kicks in when the battery voltage has fallen below a certain threshold. It charges the battery at a very low constant current until the threshold voltage is reached, where it can then continue charging as normal.

Table 10: Battery Charger IC Comparison

Part Number	BQ2057CSW	MCP73831
Manufacturer	Texas Instruments	Microchip Technology
Charge Current	2000mA	500mA
Temperature Monitoring	External Thermistor	Internal Thermistor
Cell Conditioning	Yes	No
Charge Status LED	Yes	No
Operating Supply Voltage	4.5V - 15V	3.75V - 6V
Single Purchase Price	$2.10	$0.58
Bulk Purchase Price	$0.85	$0.42

The IC also has a thermistor input which can be placed near the battery cell. This thermistor is used to constantly monitor the temperature of the battery. If the temperature goes too high, the IC will throttle the charge rate until an acceptable temperature is reached. There is a sense resistor input used for current monitoring.

This IC also has a status LED which can provide feedback for the status of the cell. It can stay lit up while the battery has a working charge and will turn off when the battery drops below the low-battery threshold voltage.

3.3.2. Microphone

When selecting a microphone to use there was a few design parameters to look for. We needed to ensure that the microphone would be able to pick up the most common string instruments. We determined that we needed to collect frequencies from 30 Hz to 4400 Hz. We also realized we should look for a condenser microphone, as they typically have much flatter frequency response graphs than dynamic microphones. This flat frequency response graph will provide a more predictable response and ensure we are measure the tone from the instrument and not amplifying any noise.

Table 11: Microphone Comparison

Part Number	CME-1538-100LB	ECM-60P	PMO-6027P-40KDQ
Manufacturer	CUI Inc.	Jameco Valuepro	Mallory Sonalert Products Inc.
Type of Microphone	Condenser	Condenser	Condenser
Pass Band	20 Hz - 20 kHz	20 Hz - 12 kHz	20 Hz - 16 kHz
Sensitivity	-38dB	-54dB	-40dB
Signal to Noise Ratio	58dB	40dB	56dB
Diameter	4mm	9.8mm	6mm
Depth	2.9mm	12.8mm	2.95mm
Termination Type	Termination Wires	Termination Pins	Termination Wires
Unit Price	$3.16	$0.99	$2.08
Bulk Price	$1.896	$0.55	$.8736

The microphone also needed to be sensitive enough to pick up on the quiet tones from electric guitars and bass guitars that are not being amplified. It also needs to have a high signal to noise ratio. This will ensure that the signal is clear. The microphone can be either have an analog or digital output, depending on which is used it will affect other systems.

The physical characteristics also are very important. The size of the microphone should be as small as possible to keep the overall size of the device as small as possible. It is also important to consider the terminal of the microphone. The microphone may not be near the PCB, so we may need a breakout board or another solution to ensure the microphone is integrated into the system.We narrowed the microphone selection down to three options by three different manufacturers.

The ECM-60P was determined to be too large and would need additional parts, because of the termination pins, to be integrated into the rest of the system. We ultimately selected the PMO-6027P-40KDQ microphone manufactured by Mallory Sonalert Products Inc. This microphone was a condenser microphone as desired as well as fulfilling the pass band requirement necessary to pick up the most common string instruments. The selected microphone had comparable signal to noise ratio, diameter, and depth to the CME-1538-100LB microphone however it cost almost 50% cheaper at bulk prices. While the CME-1538-100LB would be the best microphone for performance, marginal performance gain is not worth paying twice as much.

3.3.3. Analog to Digital Converters

The design is going to need an analog to digital converter. When selecting the best chip, there are a few design characteristics that need to be considered. To ensure that the waveform is preserved, the sampling rate needs to be at least twice the input frequency. The bit resolution also needs to be considered. The more bits the more precise the converter will be. The interface also needs to be considered to ensure it can work with the microprocessor that is selected. Supply voltage should also be considered to ensure it will be easily integratable into the system.

We decided to go with the 579-MCP3221A5T-I/OT analog to digital converter manufactured by Microchip Technology. The two different analog to digital converters have the same resolution and interface options. The operating supply voltage is comparable between the two as well and will allow it to be integrated into the system easily. The Texas Instruments analog to digital convert has a lot more channels than necessary and just adds unnecessary complexity. The Texas Instruments chip also would need to be ran in Fast Mode as well in order to capture all the desired frequencies from the stringed instruments. The Microchip Technology chip has a sampling rate that while it won't be able to convert all the frequencies that the microphone picks up, those frequencies will have been filtered out before being converted. Additionally, the Microchip Technology analog to digital converter is much cheaper than the Texas Instruments chip. The built in analog to digital converter was not chosen due to the lower resolution.

Table 12: ADC Comparison

Part Number	ADC128D818CIMTX/NOPB	579-MCP3221A5T-I/OT	MK20DX256VLH7
Manufacturer	Texas Instruments	Microchip Technology	NXP/Freescale
Resolution	12 Bit	12 Bit	16 Bit
Number of Channels	8	1	2
Sampling Rate	8.33 kS/s in Standard Mode 33.3 kS/S in Fast Mode	22.3 kS/sec	818.33 kS/sec
Interface Type	Serial, I2C	Serial, I2C	Built In MCU
Operating Supply Voltage	3 V - 5.5 V	2.7 V - 5.5 V	Built In MCU
Single Purchase Price	$5.27	$1.58	N/A
Bulk Purchase Price	$2.34	$1.09	N/A

3.3.4. Battery

A single cell series setup (1S1P) was chosen for its simplicity and lower cost compared to a 2S1P or 1S2P setup. More complicated setups (such as 2S1P and 1S2P) require complicated balancer circuits and battery management systems to avoid risk of imbalanced cells and dangerous situations. This design only requires protection on one cell which is built into the charging circuit. It also opens up the ability to charge the cell from a 5V power source such as a cell phone charger. This ability is great for consumers as it simplifies the process of finding a charger and makes backups easy to find. In order to use this 1S1P setup, the cells need to be capable of consistent voltage over a wide range of current draw, making the 18650-cell design desirable. Two main models of 18650 cells were compared before making a final decision, the LG Lithium Ion INR18650-HG2, and the Samsung INR18650-25R.

We decided to use the LG INR18650-HG2 for our initial testing and implementation. Both cells would work for our implementation, offering high max discharge rates and charge rates of 0.5 C. The main difference between these cells is the capacity, with the LG cell having a 20% advantage over the Samsung competitor for only a 2.5% increase in price when buying in bulk.

Table 13: Battery Cell Comparison

Part Number	INR18650-HG2	INR18650-25R
Manufacturer	LG	Samsung
Nominal Voltage	3.6V	3.6V
Capacity	3000mAh	2500mAh
Standard Charge Rate	• Constant Current: 1500mA • Constant Voltage: 4.2V • End Current (Cut Off): 50mA	• Constant Current: 1250mA • Constant Voltage: 4.2V • End Current (Cut Off): 125mA
Max Discharge Rate	20A	20A
Cutoff Voltage	2V	2.5V
Single Purchase Price	$7.85	$7.75
Bulk Purchase Price	$7.06	$6.91

3.3.5. Fuel Gauge

A battery fuel gauge is an important part of a BMS because it allows the overall system to intelligently decide when to request charging and power down. Without this in place, the device would be able to drain the battery cell to much lower voltages than recommended and possibly cause permanent damage. For this specific part, two devices were compared, the Texas Instruments bq27541-G1 and the MAXIM MAX17043G+U.

After much consideration, the onboard ADC was chosen due to its lower power draw and more simple configuration. The extra features offered by the bq27542 would be useful, but are covered by other components in the BMS chain. This device will provide adequate feedback on battery state and prevent any unwanted damage to the Li Ion battery cell.

Table 14: Fuel Gauge Comparison

Part Number	bq27542-G1	MAX17043G+U	Onboard ADC
Manufacturer	Texas Instruments	MAXIM	NXP/Freescale
Supply Voltage	2.7V-5.5V	2.5V-4.5V	N/A
Active Current	131 uA	60 uA	N/A
Communication Interface	I2C and HDQ	I2C and Alert Interrupt	N/A
Voltage Tracking	Yes	Yes	Yes
Current Tracking	Yes	No	No
Temperature Monitoring	Yes	No	No
Single Purchase Price	$3.07	$3.28	N/A
Bulk Purchase Price	$1.44	$1.49	N/A

3.3.6. High Voltage

Because a 1S1P battery cell setup was chosen, the nominal voltage from the battery circuit is 3.6V, which is too low to provide adequate power to the motor and some of the logic. With this in mind, a boost converter circuit is needed to step the voltage up to 7V. Two boost converter ICs were analyzed in the table below.

The 2S selected due to its simplicity and ease of implementation. The battery charging IC is easily updated to the 2S charging version and the rest of the power circuit remains untouched. The 2S configuration is the same efficiency

Table 15: Boost Converter Comparison

Part Number	LM2700QMT-ADJ/NOPB	LT3479	2S Battery Pack
Manufacturer	Texas Instruments	Linear Technology	N/A
Minimum Input Voltage	2.2V	2.5V	7V
Maximum Output Current	2.5A	3A	20A
Voltage Output Range	2.2V-17.5V	2.5V-24V	7V-8.4V
Undervoltage Protection	Yes	No	No
Short Circuit Protection	Yes	Yes	No
Cutoff Voltage	2V	2.5V	0V
Single Purchase Price	$5.09	$3.56	$7
Bulk Purchase Price	$3.75	$3.18	$6.50

3.3.7. H-Bridge

Table 16: H-Bridge Comparison

Part Number	L239D	SN754410
Manufacturer	STMicroelectronics	Texas Instruments
Vcc1 (Logic Supply)	4.5V-5.5V	4.5V-5.5V
Vcc2 (Load Supply)	4.5V-36V	4.5V-36V
Output Current (Load)	600mA	1A
Single Purchase Price	$3.90	$2.43
Bulk Purchase Price	$2.06	$1.20

The H-bridge has the job of controlling power and direction to the selected motor. This circuit should be capable of delivering enough current to the motor to reach the desired torque output, while also working at the required voltage supplies as given from the power regulation system.

Both of these parts are very similar in their operating ratings and would work for the project. The SN754410, however, stands out with its slightly higher operating current and lower purchase price.

3.3.8. Operational Amplifier

This design is going to need a couple of operational amplifiers in order to design filters, amplifiers and other electrical equipment. The number of channels, the slew rate, and that it needs to be capable of running off a single supply.

We selected the TL084CN operational amplifier. Both of the chips are manufactured by Texas Instruments, a well known and highly reputable company. The group also knows a Texas Instrument employee that assists with operational amplifier sales. This will allow for easy troubleshooting and help should we need it. They both were single supply compatible, this allows the design to be significantly easier. It had a higher slew rate. It also had a higher number of channels, which allows for fewer chips on the printed circuit board. This will save space and make the printed circuit board cheaper. The fact that it has twice the number of channels make up for the bulk purchase price being a bit more expensive than the LM358 bulk. Additionally, the single purchase price of the TL084CN is cheaper than that of the LM358.

Table 17: Operational Amplifier Comparison

Part Number	LM358	TL084CN
Manufacturer	Texas Instruments	Texas Instruments
Slew Rate	0.3 V/us	13 V/us
Number of Channels	2	4
Single Supply Compatible	Yes	Yes
Single Purchase Price	$1.00	$0.60
Bulk Purchase Price	$0.15	$0.194

3.3.9. Linear and Switching Regulators

Switching regulators use a buck converter topology to step an input voltage down to a regulated power rail. These circuits require external inductors and capacitors to assist in this step down and often need to be filtered for noise. Linear regulators are simple components which operate as variable resistors in series with the load. The resistance is varied to control the voltage to a regulated state and the rest of the power is dissipated in heat. The following components were chosen for comparison for the 5 volt rail.

For the purpose of this project, the LM7805CV was chosen. LM7805CV is a linear regulator which takes Vin from 7V to 35V and outputs a regulated 5V DC value. This 5V rail is used to power the TL084CN Operational Amplifier which is used as a preamp to boost the value coming from the microphone before it is sent to the ADC. This IC comes in a TO-220 package which makes testing very easy since the legs fit into the breadboard sockets. This also enables use of an external heatsink if the heat dissipation is too high. For implementation, the IC also comes in a SOT-23 package which is much more compact , as long as the power dissipation is low enough. The regulator can supply up to 1.5A of current which gives a large amount of headroom in case future design revisions include higher draw 5V components.

Table 18: 5V Regulator Comparison

Part Number	LM7805CV	LT1076-5
Manufacturer	STMicroelectronics	Linear Technology
Input Voltage	7V-35V	8V-45V
Maximum Output Current	1.5A	3A
Thermal Overload Protection	Yes	No
Single Purchase Price	$1.00	$6.78
Bulk Purchase Price	$0.15	$3.50

Table 19: 3.3V Regulator Comparison

Part Number	LD33V	MAX640
Manufacturer	STMicroelectronics	MAXIM
Input Voltage	4.3V-15V	4V-11.5V
Maximum Output Current	950mA	225mA
Thermal Overload Protection	Yes	No
Single Purchase Price	$0.60	$4.31
Bulk Purchase Price	$0.194	$3.54

The LD33V was chosen for its simplicity and greater output current which could be useful if future additions need a regulated 3.3V rail. The LD33V is a linear regulator which takes Vin from 3.5V to 15V and outputs a regulated 3.3V DC value. This 3.3V rail is used to power the MSP430 Master Control Unit as well as the SSD1306 OLED display. This IC comes in a TO-220 and SOT-23 package which allows easy breadboarding in the testing phase and compact implementation in the integration phase. This IC can supply up to 950 mA of current which should be ample for current plans and any future changes which could require use of the 3.3V line.

3.3.10. Motors

Motor selection for physical turning of the peg needs to be done carefully in order to provide adequate speed as well as torque. One of the desired features of this device is to allow for easy de-stringing and re-stringing of the guitar so a minimum rotational speed of 30 RPM was chosen to keep this functionality practical. The minimum torque was determined by physical testing on a guitar.

Torque is a measure of rotational force, measured in units of force*length. In order to measure the needed torque to tune a knob on the guitar, the peg was attached to a lever arm and a calibrated scale was used to measure the force needed to rotate the lever. The following equation was used to calculate the torque being applied.

$$\tau \text{ (Torque)} = F * D \text{ (Force * Distance)}$$

Table 20: Torque Requirements

String	Distance (in)	Force (oz)	Torque (oz-in)
1	6	2.1	12.6
2	6	2.5	15
3	6	3.8	22.8
4	6	2.7	16.2
5	6	3.1	18.6
6	6	4.5	27

Since testing shows that a value of 27 oz-in is required to rotate the thickest string on a 6 string guitar, a minimum output torque of 35 oz-in was chosen to guarantee adequate tuning across all guitars. Two motors are chosen for testing purposes in this project; The Parallax continuous rotation motor, and the uxcell 61 RPM Standard Gearmotor.

Table 21: Motor Comparison

Part Number	Continuous Rotation Servo #900-00008	Standard Gearmotor
Manufacturer	Parallax	uxcell
Input Voltage	4V-6V	3V-12V
Stall Current	190 mA	500 mA
Output Torque	38 oz-in	69 oz-in
Rotation Speed	0-50 rpm	0-60 rpm
Control Method	PWM	H-Bridge
Weight	42.5g	154g
Single Purchase Price	$13.99	$12.48

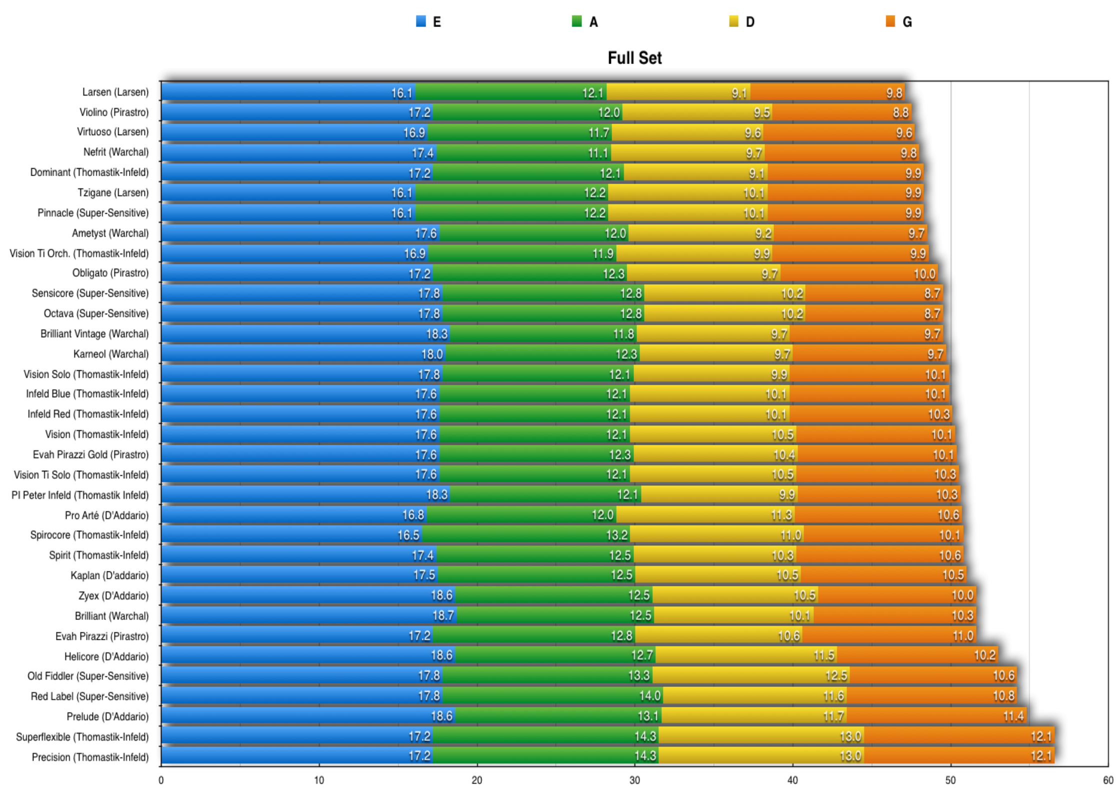


Figure 15: String Tension (lbs.) By Manufacturer [19]

The Parallax motor provides just enough torque to meet the requirement while turning up to 50 rounds per minute (RPM). This torque, when combined with the speed should match up with the required <10s tuning requirement.

The uxcell motor provides a marginally higher speed and torque output at the cost of more power draw and weight. This motor is also adequate to meet the requirements set earlier in the paper.

The decision between motors proved too difficult to narrow down with only specifications from the data sheets, therefore, both were purchased and will be tested. The Parallax motor provides a more simple implementation, needing only power and a PWM signal from the microcontroller, but it is also a more bulky form factor than desired for the final design. The uxcell motor provides nearly double the torque of the Parallax motor and rotates faster which would provide a better end experience for the user. This comes at the cost of a more complicated drive circuit with an H-bridge, and more weight to the end product.

3.3.11. Display

Table 22: Display Comparison

Part Number	Monochrome OLED	1.8" TFT LCD
Manufacturer	Adafruit	SainSmart
Input Voltage	5V	5V
Input Current	20 mA	50 mA
IC Driver	SSD1306	ST7735R
Communication	I2C	SPI
Resolution	139 PPI	114 PPI
Colors	2 bit	18 bit
Single Purchase Price	$19.50	$10.99
Bulk Purchase Price	$9.89	$4.17

The display is one of the most important factors when it comes to standalone usage without a smartphone. The display needs to be bright enough supply a high enough contrast ratio, and have a large enough resolution to prevent strain when trying to read characters. With this in mind, two types of display technology were considered, OLED and LCD.

The SainSmart 1.8" TFT Color LCD Module, uses a backlight which shines through a matrix of light polarizing crystals. These crystals can be twisted using electric current to either allow light to go through, or prevent light from going through. These displays offer a good amount of brightness, but the contrast ratio is often measured to be around 500:1, which is lower than the recommended 1000:1 for low eye strain. These displays are also usually lower resolution which means characters will appear more blocky and harder to read.

The Adafruit Monochrome 0.96" 128x64 OLED module uses individual light emitting diodes for each of its pixels. When a pixel is commanded black, it is turned off and emits no light. This gives OLED displays a terrific advantage in contrast ratio which is seen as near infinite as well as better efficiency when a mainly black background is used. This display also boasts a resolution of 139 ppi which increases readability for the end user.

For this project, the OLED display was selected for its power efficiency and great contrast ratio. It also provides a much better resolution and offers the chance at greater efficiency when displaying black colors. This display proves adequate for the requirements and human factor.

3.3.12. Microcontrollers

The MSP430 family of microcontrollers is known for its ultra-low-power performance which comes at a low financial cost utilizing a 16-bit RISC CPU. This makes an MSP430 microcontroller an ideal choice for use in embedded systems where low power performance is a necessity.

3.3.12.1. MSP430G2553

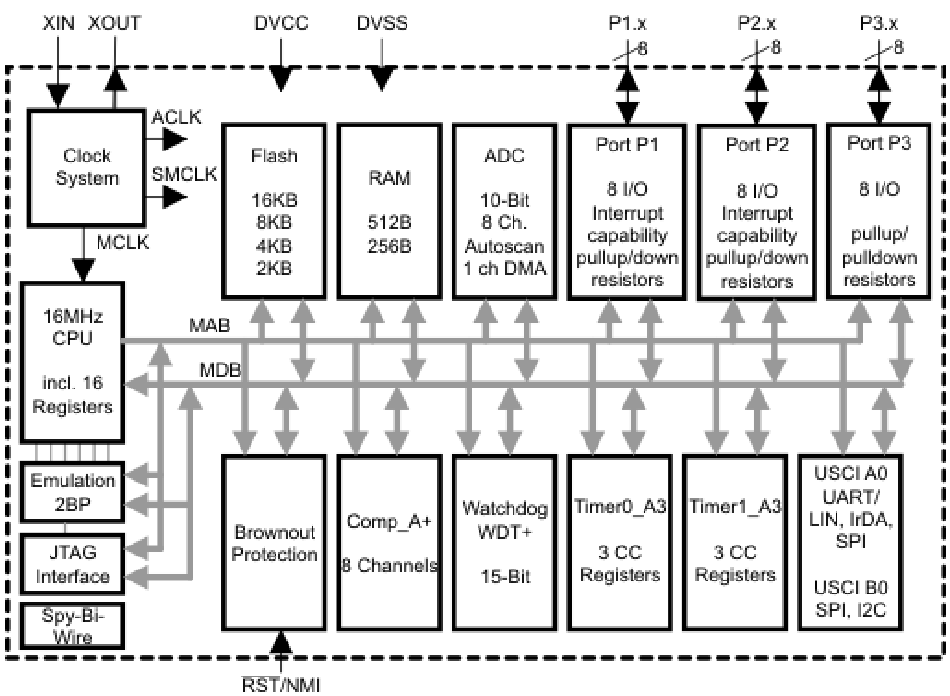


Figure 16: MSP430G2xx Functional Block Diagram Courtesy of Texas Instruments

The 16-bit MSP430G2553 runs at a 16MHz frequency and offers 512 Bytes of RAM and 16KB of Flash along with providing communication via I2C, SPI, and UART. A voltage source capable of supplying between 1.8V to 3.6V is required for operation in one of its five power-saving modes. The most commonly used modes of operation are Standby Mode, requiring a current of 0.5uA, and Active Mode which requires 230uA at a frequency of 1MHz and a voltage of 2.2V. This particular series of microcontroller also offers an extremely fast wake-up time of less than 1us, and a Brownout Detector allowing much safer and more reliable operation as the controller with shutdown if the power supply dips. This MCU is available from Texas Instruments for $2.50.

3.3.12.2. MSP430F5529

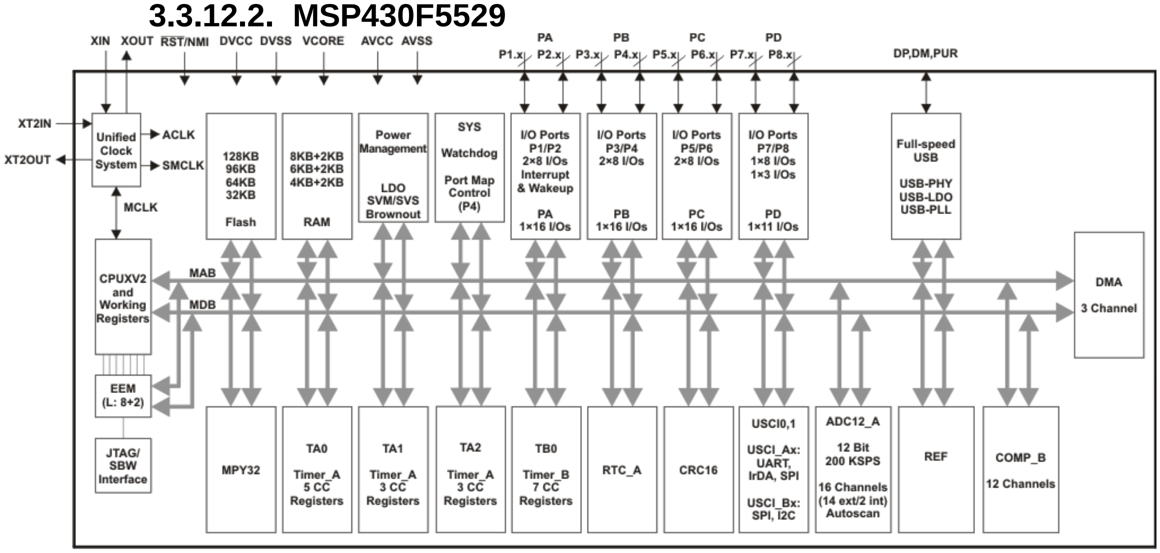


Figure 17: MSP430x5xx Series Functional Block Diagram Courtesy of Texas Instruments

The MSP430F5529 runs at a 25MHz frequency and boasts 10KB of RAM, the most in its series.128KB of Flash is available along with communication via I2C, SPI, and UART. A voltage source capable of supplying between 1.8V to 3.6V is required for operation in one of its five power-saving modes. This MCU consumes 404uA/MHz in Active Power mode and 2.5uA in Standby Power mode. The wake-up time is much slower compared to its G2 series counterpart at 3.5us. This microcontroller also comes equipped with a Brownout Detector for increased safety and reliability. This MCU is available from Texas Instruments for $8.06.

3.3.12.1. Microchip ATUC64L3U

Microchip's ATUC64L3U 32-bit AVR microcontroller offers high performance at low power with a max CPU frequency of 50MHz. An operating voltage of 1.62V-3.6V is required, and the device produces 9nA of current leakage. The controller comes equipped with 64KB of Flash program memory and 16KB of

RAM. Communication between 1 SPI, 2 I2C, and 1 USB devices is supported. This chip is available via Microchip Direct for $5.71.

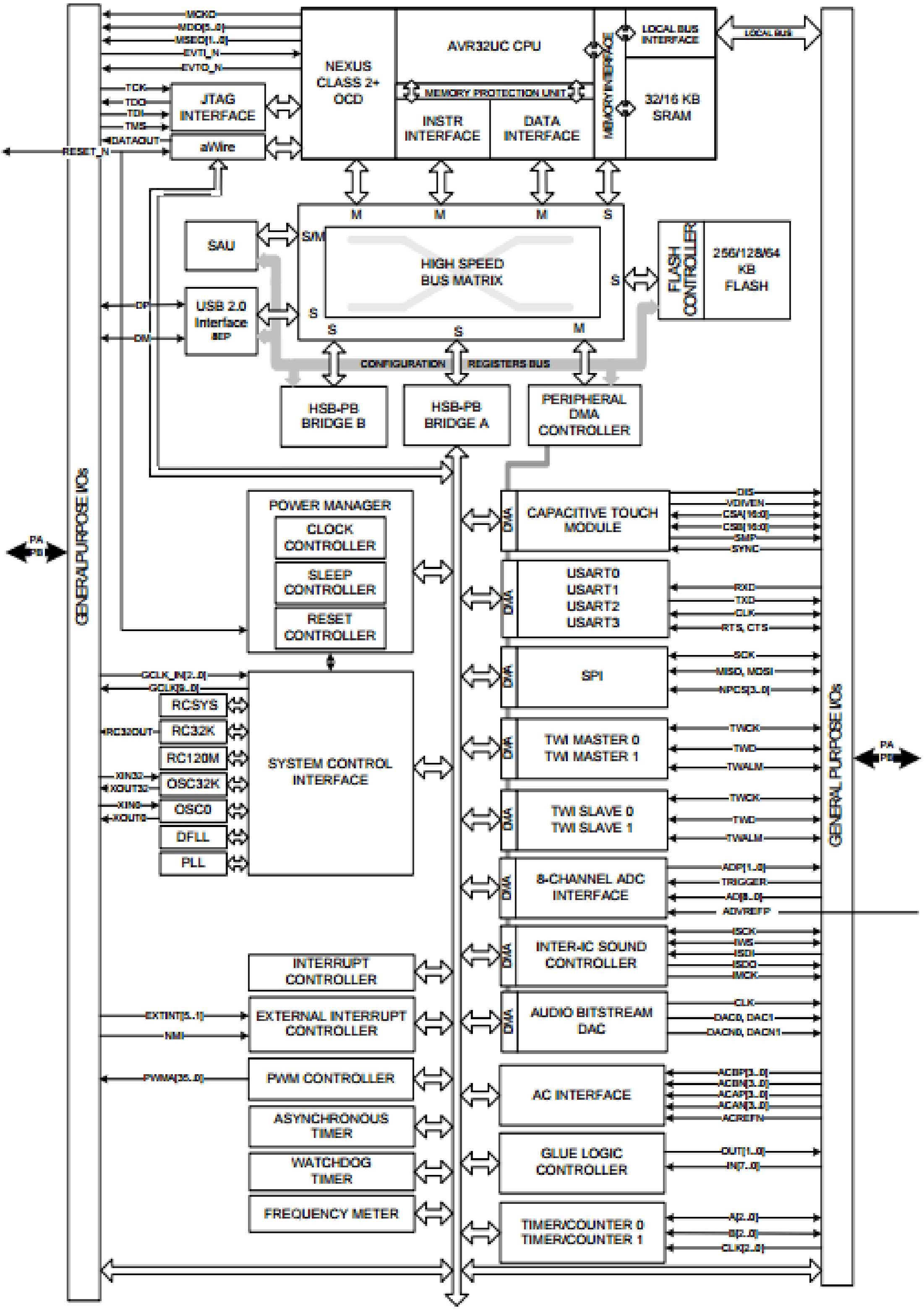


Figure 18: ATUC64L3U Functional Block Diagram Courtesy of Microchip

3.3.12.1. Microchip dsPIC33EV32GM104

The 16-bit dsPIC33EV32GM104 produced by Microchip has a processing speed of 70 MIPS under an operating voltage of 4.5V-5.5V. In addition, 4KB of RAM and 32KB of Program Flash Memory is available onboard. Communication is supported for 2 UART, 2 SPI, and 1 I2C peripheral. This chip is available via Microchip Direct for $3.20.

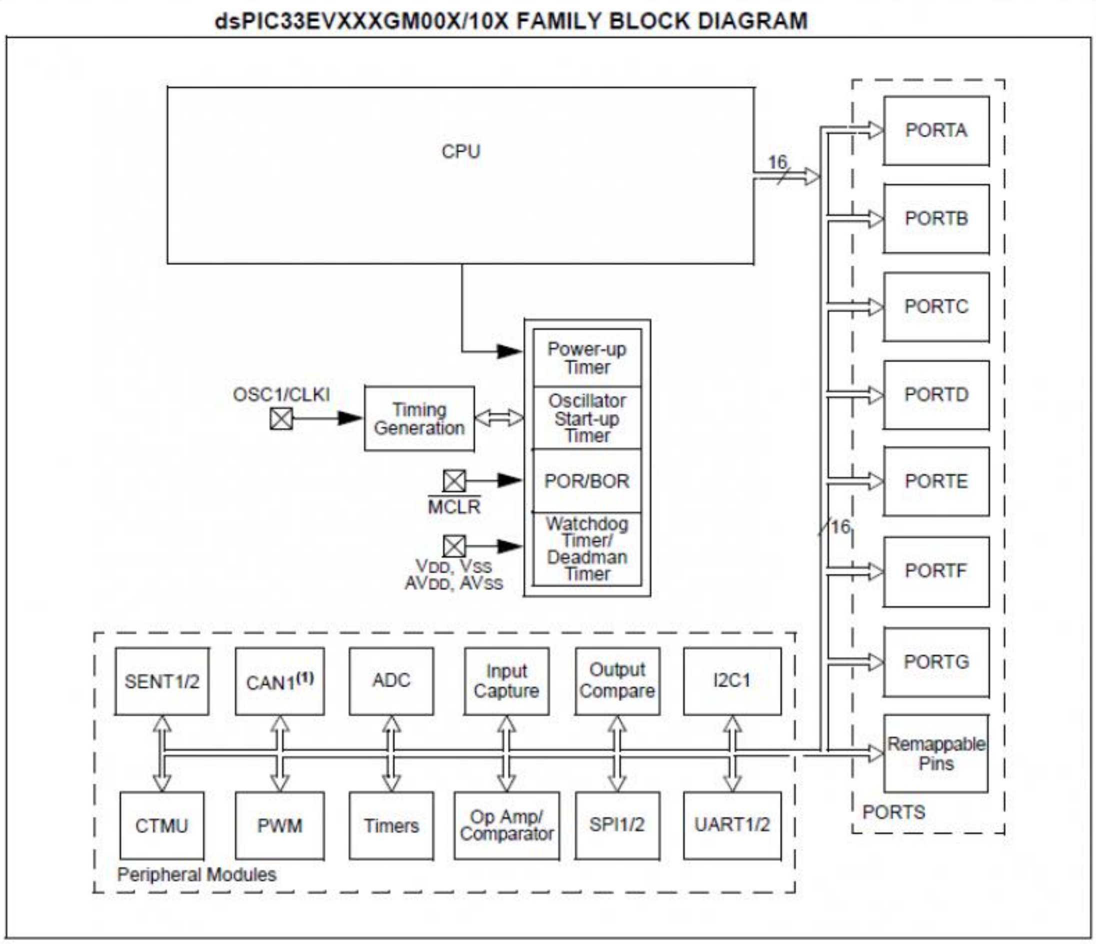


Figure 19: dsPIC33EV32GM104 Block Diagram Courtesy of Microchip

3.3.12.2. Intel Quark D1000

The 32-bit Intel Quark Microcontroller D1000 runs at a base frequency of 33MHz. The operating voltage range is 1.62V-3.63V with a TDP of 25mW. When operating in TDP-down mode, only 1.6mW is dissipated by the MCU. I/O communication via SPI, I2C, 24GPIO, and UART is provided by the chip. Additionally, 8KB SRAM, 32KB of instruction Flash, and 4KB of data are available on-die. This microcontroller may be purchased directly from Intel for $9.63.

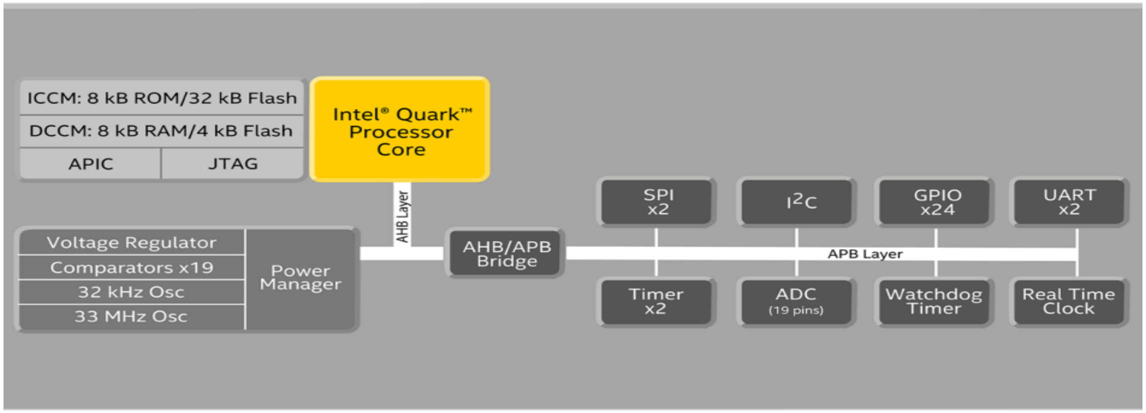


Figure 20: Intel Quark D1000 Functional Block Diagram Courtesy of Intel

3.3.12.3. Teensy 3.2

PJRC's Teensy 3.2 ARM-based, 32-bit MCU runs at an extremely fast frequency of 72MHz. Included onboard is 64KB RAM and 256KB Flash. Two 16-bit ADC's are included. The chip offers a variety of communication options in the form of, 3 SPI, 1 I2C, 1 USB, and 9 touch-sensitive pins. With the relatively high specs comes a high operating voltage of 3.6-6.0V and a higher cost of $19.80 per unit.

Table 23: MCU Comparison

Name	Operating Voltage (V)	Bit Width (bits)	Frequency (MHz)	Flash (KB)	RAM (KB)	Price (USD)
MSP430G2553	1.8-3.6	16	16	16	0.5	2.50
MSP430F5529	1.8-3.6	32	25	128	10	8.06
Quark D1000	1.62-3.63	32	33	8	32	9.63
dsPIC33EV32GM104	4.5-5.5	16	70 MIPs	32	4	3.20
ATUC64L3U	1.62-3.6	32	50	64	16	5.71
Teensy 3.2	3.6-6.0	32	72	256	64	19.80

3.3.13. Connection Selection

To go along with the guitar tuner, there will be an app made. The app has various functions, one of which is being able to save previous tunings. The automatic guitar tuner will need to be able to pair with a cellphone that has the app installed. There are different ways of doing this, either by USB, Bluetooth, or Wi-Fi direct connection. Through research on each of these three types of connection, it will be determined which connection is the best one to use for the project at hand.

First USB connection is looked at. “USB, which stands for Universal Serial Bus, is the cables, connectors and communication protocols used in a bus for connection, communication, and power supply between devices.[1]” A bus is “a communication system that transfers data between components inside a computer, or between computers.[1]” So a USB a type of connection between two devices that can transfer data between one another. There are three main types of USB connectors: the standard format, the mini format, and the micro size format. The standard format is used for desktops, or any other portable equipment like USB flash drives. Both the mini and micro are both used on mobile devices, like cell phones, however the micro USB connection is what is used in most cell phones today. Since our project will need to pair with cell phones, the micro USB will get further researched. Most of the major cell phones making companies use micro USB connection, expect for Apple which until 2012 used their 30-pin dock connector. Now Apple uses a Lightning connector for their cell phones, the iPhone. It can’t be assumed everyone that uses the device will be a non-Apple user, so the Lightning connector will also be looked at alongside the micro USB connection.

The micro USB, the thinner replacement of the mini USB, was introduced in 2007. There are three types of micro USB, Micro-A, Micro-B, and Micro-AB. Each of these is a USB 2.0 connection, for example the Micro-B is a USB 2.0 to Micro-B connection, so it will have the speed that USB 2.0 offers. USB 2.0 was released in 2000 and has High Speed transfer rates of 480 Mbps (Megabits per second), or 60 MBps (Megabytes per second). However the actual High Speed throughput of transfer rates is typically up to 35-40 MBps, or 280-320 Mbps. There was a new release of USB, USB 3.0, which was released in 2008, and then USB 3.1 was released in 2014. USB 3.0 has faster transfer speeds of 5 Gbps (Gigabits per second), however most smartphones today still use the universal micro-B, and therefore USB 2.0, connection. Other specifications with USB 2.0 are that provides up to 500 mA (milliamps) of power consumption and can only handle one direction of data transfer at any given time. USB 3.0 has two unidirectional data paths, one to receive data and the other to transfer data. USB 2.0 doesn't have this ability, as it can only handle one direction of transfer at once, so this is a limitation of USB 2.0. Other limitations of USB 2.0 include the maximum data rates, actual data throughput, and power. These have been explained above. A major limitation to USB connection is the cable length.

USB connection requires a cable to connect devices, while things like Bluetooth and Wi-Fi direct allow devices to connect to each other wirelessly. Though these two also have limitations to the distance that's needed to stay connected, they can both be used at a further distance than a USB connection. To keep a Full speed or Hi-speed rate, a USB 2.0 connection has a maximum cable length of five meters, or about 16 feet and five inches. This could be extended by connect some extension cables together, but this would slow down the speed of the transfer rate. Since our project is a hand held device that connects to a cell phone, the length issue wouldn't be a major issue. A bigger issue would be the fact that a USB port would need to be added onto the automatic guitar tuner. USB ports are bulky and would take up space in the design, and with our project being a handheld device we're trying to keep it as small as possible. However, USB 2.0 only covers most of the modern day cell phones, but Apple uses their Lightning connector for their cell phones, and this is looked at next.

Version	Speed	Bits/sec	HD movie 25GB
USB 1.1	Low speed (LS) Full speed (FS)	1.5 Mbps 12 Mbps	~9.25 hours
USB 2.0	High speed (HS)	480 Mbps	~14 mins
USB 3.0	SuperSpeed (SS)	5 Gbps	~70 sec
USB 3.1	SuperSpeedPlus (SSP)	10 Gbps	~35 sec

Figure 21: USB Generational Speeds [20]

The Lightning connector for Apple cell phones was introduced in 2012. It replaced Apple's previous connector, the 30-pin dock connector, with the major differences being that Lightning is smaller and it can be inserted into an Apple cell phone with either side of it facing up. It has eight pins on the top and bottom, giving it a total pin count of sixteen. The lightning connector currently can only transfer data at around 25-35 MBps, or 200-280 Mbps, which puts it at USB 2.0 speed, similar to the Micro-B connection explained above. Though some Apple devices, such as the newer Macbooks, are able to support USB 3.0. USB 3.0 transfer rates are significantly faster than USB 2.0, as it can transfer at rates up to 625 MBps, or 5 Gbps, which is 10 times faster than USB 2.0. The application that's being made for the automatic guitar tuner is being made as a cell phone application, and since no current iPhones use USB 3.0, the transfer speeds will be the same as the Micro-B USB connection. Therefore, in conclusion, USB connection will offer max transfer speeds of 35-40 MBps, with the only issues being having to use a cable to connect the automatic guitar tuner to a cell phone, and the amount of space that a USB port would take up in our project design. This is as opposed to transferring the information wirelessly, through Wi-Fi direct, which don't need a bulky USB port, each of which is described below.

Wi-Fi direct, initially called Wi-Fi P2P, is a Wi-Fi standard enabling devices to easily connect with each other without requiring a wireless access point. A wireless access point, or WAP, "is a network hardware device that allows a Wi-Fi compliant device to connect to a wired network.[2]" A WAP connects to a wired Ethernet connection and the WAP is then able to provide wireless connection "using radio frequency links for other devices to utilize that wired connection.[2]" A WAP is commonly used today on college campuses and in homes as a way for devices to connect to the internet wirelessly. In these cases the Wi-Fi networks are set up in “infrastructure mode” and they act as a central hub to which the Wi-Fi enabled devices can connect to. However in this case, a WAP is needed to be able to share data between devices, Wi-Fi direct doesn’t require or need there to be a WAP nearby to be able to transfer data between devices, as one device would act as the access point. This would enable the automatic guitar tuner and a cell phone to pair with just one another and transfer data wirelessly, much like Bluetooth. In our projects case when the two devices, the cell phone and automatic guitar tuner, first connect, through Wi-Fi direct, they will negotiate to determine which device will act as the access point.

Though in our projects case the automatic guitar tuner would end up being the access point. This is because any Android 4.0, which was released by Google in October 2011, cell phones support Wi-Fi direct, however no iPhone device currently implements Wi-Fi Direct, instead iPhone’s operating system, iOS, has its own feature for Wi-Fi connection between two devices. Thus, since only one of the two devices needs to have Wi-Fi direct capabilities, making the automatic guitar tuner have Wi-Fi direct support, and making it the access point for phones to connect to, eliminates the problem that iPhones present. This way whenever any cell phone needs to connect to the tuner, through Wi-FI direct, the cell phone will be able to easily find the guitar tuner and connect to it.

Wi-Fi direct claims to “supports typical Wi-Fi speeds, which can be as high as 250 Mbps.” This is from the Wi-Fi Alliance website, where Wi-Fi Alliance is the worldwide network of companies that brings Wi-Fi to us. The speed that Wi-Fi direct transfer data at, 250 Mbps or 31.25 MBps, is slightly lower than the speeds that USB 2.0 offers, but it comes without having to use any cables. Wi-Fi direct also only needs to be on one of the devices for it to be able to work, while USB requires each device to have some kind of USB port, so the two devices can be connected. Also, according to Wi-Fi Alliance, Wi-Fi direct is able to have a range of up to 200 meters. This is much longer than what USB 2.0 offers, which to get the fastest speed USB 2.0 offers is only 16 feet and five inches. Though getting 200 meters could be hard to obtain inside a house, as there would be obstacles in the way, such as walls, so the actual distance inside would be shorter than 200 meters; but Wi-Fi direct, in the end, offers a distance further than USB is able to offer. Ideally the user’s cell phone would be close to the automatic guitar tuner, so that data could transfer between the two devices easily and as fast as possible.

Bluetooth connection is another form of wireless connection, and one that is widely used today. Bluetooth has been around since its first release in 1998, with the newest release, Bluetooth 5.0, that came out on December 6, 2016. Bluetooth is “a wireless technology for exchanging data over short distances, using short-wavelength UHF radio waves in the ISM band from 2.4 to 2.485 GHz(giga hertz), from fixed and mobile devices.[3]” UHF stands for Ultra high frequency, and it’s the International Telecommunication Union(ITU), which is an agency of the United Nations that’s responsible for issues dealing with information and communication technologies, designation for radio frequencies in the range of 300 MHz(megahertz) to 3 GHz(giga hertz). In other words,

Bluetooth works by allowing devices to connect to one another by radio waves, as opposed to wires or cables. Bluetooth uses radio waves, by using a variation of a radio technology called frequency-hopping spread spectrum. Frequency-hopping spread spectrum, abbreviated FHSS, “is a wireless technology that spreads its signal over rapidly changing frequencies. Each available frequency band is divided into sub-frequencies. Signals rapidly change ("hop") among these in a predetermined order. The variation of FHSS that Bluetooth uses is called Adaptive Frequency-hopping spread spectrum, abbreviated AFH. AFH works by “ improves resistance to radio frequency interference by avoiding crowded frequencies in the hopping sequence.” The key idea with AFH is to use only the “good” frequencies, by avoiding the “bad” frequencies. However, there was a problem called dynamic interference that arose, but gradual reduction of available hopping channels and backward compatibility with legacy bluetooth devices was resolved in version 1.2 of the Bluetooth Standard (2003).

Bluetooth has a master-slave structure. A master-slave structure is “a model of communication where one device or process has unidirectional control over one or more other devices. In some systems a master is selected from a group of eligible devices, with the other devices acting in the role of slaves.[3]” Two devices connect by pairing to one another, after this is done the devices are able to interact and transfer data between one another. Bluetooth has transfer rates of 25 Mbps, or 3.125 MBps, and the range of Bluetooth is 300 feet, or about 92 meters. This is slower than both Wi-Fi direct and USB 2.0, however bluetooth is more common with cell phones than Wi-Fi direct and it’s wireless, unlike a USB connection. While range wise, Bluetooth can go further than USB, but has a shorter distance than Wi-fi direct.

Table 24: Connection Comparison

Connection type	**USB 2.0**	**Wi-Fi Direct**	**Bluetooth**
Range	5 meters	200 meters	92 meters
Transfer rate(max)	280 Mbps	250 Mbps	25 Mbps
Compatible	Both Android & iOS	Android	Both Android & iOS

It can be seen from the table above that USB offers that fastest transfer rate at 280 Mbps, however USB also offers the shortest distance of the three connection types. USB also isn’t wireless like Wi-Fi direct and Bluetooth, making it less convenient to use. Wi-Fi Direct offers the biggest range, however it’s only compatible with cell phones that run on Android, as iOS, which is Apple’s operating system, has it’s own Wi-Fi to Wi-Fi peer connection. Though, the app that is being made along with the automatic guitar tuner is only going to be made to run on Android, and Wi-Fi Direct only needs one of the two devices to have Wi-Fi direct support, while Bluetooth needs both devices to have Bluetooth capabilities, USB also requires the same thing. The choice comes down to Wi-Fi Direct or Bluetooth, as wireless communication is more convenient for users. However, through the research done, Bluetooth is ultimately the best way of connection to use for communication between the automatic guitar tuner and the mobile application that is also being created as a part of our senior design project. This is because Bluetooth is more commonly used and known to users, as opposed to Wi-Fi Direct, which makes the connection process more user friendly.

3.3.14. Bluetooth Connection Specifications

As stated above, Bluetooth is a wireless form of communication between two devices. Bluetooth was designed so that wires and cables would no longer be needed for communication between devices. Today Bluetooth is common on all major smart phones, it can also be found in cars. Bluetooth has a low power consumption, with standard Bluetooth, the latest standard being Bluetooth 3.0, peaking at 30mA and Bluetooth low energy peaking at 20mA. Bluetooth low energy, or LE, was released in 2010 and it’s called Bluetooth 4.0. Bluetooth 4.0 has similar transfers rates and range as Bluetooth 3.0, however it’s more energy efficient. Bluetooth 4.0 was designed primarily for devices that transfer data frequently. The app that is being made for our project will be able to save previous tunings, so if the user wanted to keep their cell phone connected to the automatic guitar tuner for an extended period of time, having low power consumption will allow both the user cell phone and the automatic guitar tuner keep a longer battery life.

As stated earlier, Bluetooth capabilities are required on both devices. The devices connect to one another by pairing together. One device, the slave, will search to try and find the master device. Once the master device is found, the two devices will pair together. To be able to have two devices connect, they need to be in close proximity to one another, however, this varies based on what type of Bluetooth is being used.

Bluetooth connection works by using a process called bonding, and a bond is generated through a process called pairing. The pairing process is triggered either by a specific request from a user to generate a bond, thus, pairing requires some kind of user interaction. Once the pairing has completed successfully, a bond forms between the two devices. This enables the two devices to connect to one another in the future without having to repeat the pairing process. The devices make a relationship by creating a link key. Once the devices store the same link key, they are paired. Once the link key is generated, a ACL, Asynchronous Connection-Less, link is encrypted to protect exchanged data against any possible eavesdropping. ACL is a communications protocol, which is a system of rules that allows devices of a communication system to transmit information, that is used as a transmission link for data communication in Bluetooth. A link can be deleted by the user from either device, this removes the bond between the devices.

Since Bluetooth uses radio communication, how far apart the two devices can be depends on the class of radio that's used. There are three different classes of Bluetooth, Class I, Class II, and Class III.

Table 25: Bluetooth Radio Class

Radio Class	**Class I**	**Class II**	**Class III**
Range(meters)	100	10	1
Power consumption(mW)	100	2.5	1

Class I radios offer a very large maximum range of 100 meters, or 300 feet, however the power consumption is rather large at 100 mW. This wouldn't be ideal for our project, as we're trying to keep the power consumption as low as possible. Class III radios offer the shortest distance of 1 meter, or 3 feet, as well as the lowest power consumption of 1 mW. Although the power consumption is low, the range is too small. Class II radios are commonly found in mobile devices. This class types also offers a good range of 10 meters, or 33 feet, while also having a small power consumption of 2.5 mW, making this class type ideal to use for the automatic guitar tuner.

Bluetooth operates as a short-range radio frequency with packet-based protocol, with a master-slave structure. Packet-based protocol is a process that operates by taking the transmitted data and putting the data into sized blocks, or packets. A packet is a formatted unit of data that consists of control information, which is the information that is provided to the data for delivery, and user data. There are 79 designated Bluetooth channels that each packet is able to transmit on. Each of these channels has a bandwidth, the range of frequencies within a given band, of 1 MHz. Since Bluetooth operates in the frequency range of 2402 and 2480 MHz, the first pack will start at 2402 MHz, and increment in value of 1 MHz, until it gets to 2480 MHz. Bluetooth performs 800 hops per second using AFH. This is done so that there's less interference with other devices that use this bandwidth.

Bluetooth creates its own network called a piconet. A piconet is a network that links a wireless group of devices using Bluetooth protocols. A piconet starts with two devices, and can grow up to eight devices. One of the devices is designated as the master, the main controlling unit, while the devices the follow the master unit are called the slave units. This master-slave structure allows for the Bluetooth system to have no collisions. There aren't collisions because each device is assigned a specific time period to transmit data, thus avoiding collisions or overlapping with other connected devices. This allows for data to only be transferred between two devices at a time, the master and one of the slaves. There could even be other piconets in the area, and there wouldn't be a problem with communication. This is due to the fact that there are 79 channels for data to hop, making the probability of interfering with another piconet very small, thus allows for communication with little or no interference. This is another reason that makes Bluetooth an excellent communication system to use for our project.

Bluetooth started out with version 1.0, which had many problems, to now with the latest release being Bluetooth 5.0. Though Bluetooth 5.0 was released in December 2016, though it's new features focus on the emerging Internet of Things, IoT, technology. Smart phones today have Bluetooth 4.0, as well as the latest release of Bluetooth 4.2. Bluetooth version 4.0 was released in 2010, with the biggest difference between it and earlier versions, is that version 4.0 offers low energy, abbreviated LE, protocols. The difference is that Bluetooth LE allows for short bursts of long range radio communication, which makes it ideal for the emerging Internet of Things technology. While, in comparison, earlier versions of Bluetooth have Bluetooth BR/EDR, which stands for Basic Rate/Enhanced Data Rate. Bluetooth BR/EDR establishes a short range continuous wireless connection, which makes it ideal for streaming music, but not ideal in power consumption like Bluetooth 4.0 is. Bluetooth LE is a entire new protocol stack for rapid build-up of simple links, where it's aimed at low power consumption. Chip designs for 4.0 also allowed for two types of implementation, single-mode and dual-mode.

In single-mode implementation, only the low energy protocol stack is implemented. Single-mode chips feature a lightweight Link Layer. The Link Layer provides ultra-low power idle mode operation, simple device discovery, and reliable point-to-multipoint data transfer. It also provides advanced power-save and secure connections at a low cost. While in dual-mode implementation, Bluetooth Smart functionality is integrated into an existing Class Bluetooth controller. Thus, in dual-mode, the device is able to support Bluetooth LE as well as the regular Bluetooth BR/EDR.

Bluetooth version 4.1 was released in 2013, and it was a software update, not a hardware update, to version 4.0. The update added new features to improve consumer usability. Some of the new features include, increased co-existence support for LTE, bulk data exchange rates, aid developer innovation by allowing devices to support multiple roles simultaneously, dual-mode, and many more.

Pairing Mechanisms

The pairing mechanisms for Bluetooth have changed significantly since Bluetooth version 2.1 was released, as Bluetooth 2.1 introduced simple pairing. Before 2.1 was released, any devices that had Bluetooth 2.0 and before had Legacy pairing, where each device had to enter the same PIN code to successfully pair. Limited pair devices, like a hands-free headset, have a fixed PIN hard-coded into the device. Numeric input devices, like mobile phones, allowed the user to enter a value up to 16 digits in length. Alpha-numeric input devices, like PCs and smartphones, allowed the user to enter full UTF-8 text as a PIN code. This kind of pairing made it non-user friendly and needed an update, which is what Bluetooth did with version 2.1. Version 2.1 introduced Secure Simple Pairing(SSP), which just works, with no user interaction. The only user interaction that could be needed is if a device prompts the user to confirm the pairing process. This made Bluetooth simpler and more user friendly.

Health Concerns

There aren't any health concerns regarding Bluetooth. The radio frequency the Bluetooth uses is harmless radiation that can not cause cancer. Also, the maximum power output from a Bluetooth device is 100 mW, which is lower than the lowest powered cell phones, which also aren't a health concern.

Security

In regards to the security of Bluetooth, Bluetooth version 2.1 addressed some major issues. Before Bluetooth 2.1, encryption isn't required and can be turned off at any time, and the key is only good for 23.5 hours. Any key used that's longer than this time allows for simple attacks. Bluetooth 2.1 fixed this by making encryption required for all non Service Discovery Protocol connections, adding a new encryption pause and resume feature, and requiring the key be refreshed before it expires.

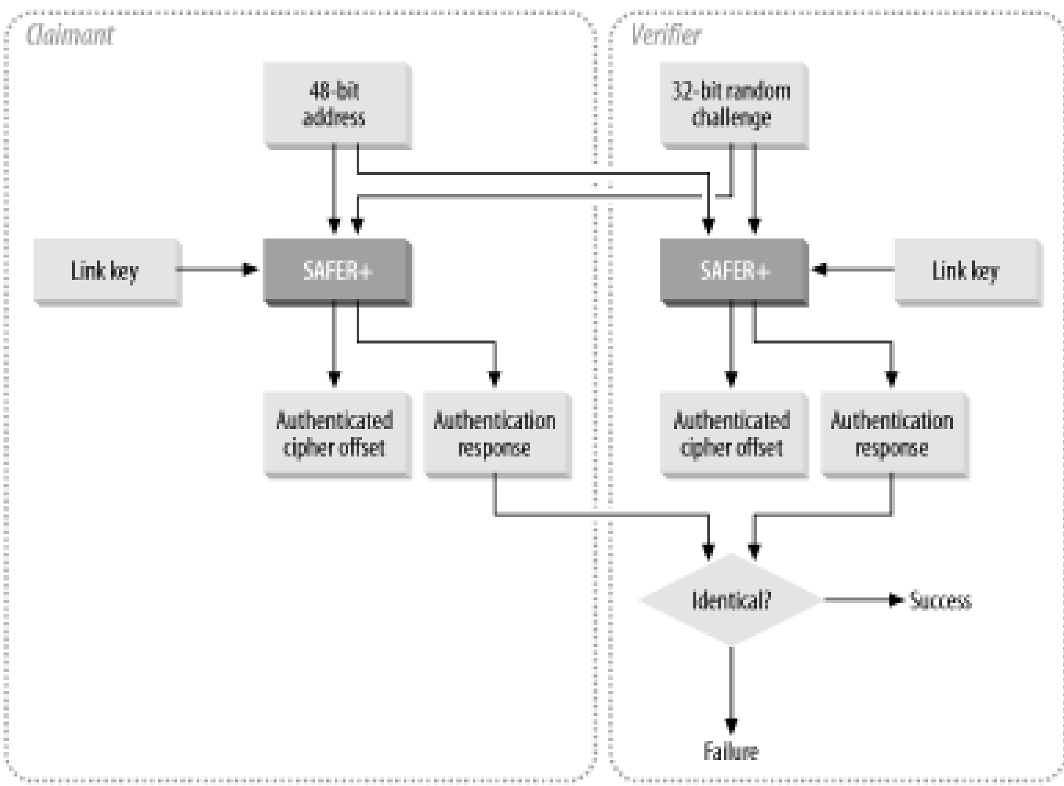


Figure 22: Bluetooth Encryption Process [21]

3.3.15. Bluetooth Protocol Stack

The Bluetooth protocol stack contains many logical layers. "The Bluetooth Host contains the logical layers of the Bluetooth architecture.[4]"This includes the core implementation of the Bluetooth stack, such as the high-level layers of the architecture that support and extend the functionality of the Bluetooth stack. The Bluetooth protocol stack is a layered set of protocols that define the core functionality of Bluetooth.

At the top of the protocol stack is the OBEX, object exchange. The OBEX is a communications protocol used to exchange binary objects between devices. OBEX is a protocol that was developed and maintained by the IrDA(Infrared Data Association), it's a session level protocol. Although it was initially designed for infrared, it was adopted by the Bluetooth Special Interest Group to be used for Bluetooth technologies. In Bluetooth, OBEX is implemented on a Baseband/L2CAP/RFCOMM stack. The objects OBEX exchanges are used for a variety of purposes, such as establishing a connection and sending data. In Bluetooth, a OBEX connection works by a device first trying to set up an OBEX communication session with another device. The device wanting to set up the connection is called the client device. Once the session is set up, communication between the two devices can begin. Where the information exchanged between the two is in the format of a packet.

The layer below the OBEX, in the Bluetooth stack, is the COM Port Emulation. This part of the protocol stack "is an API that supports virtual COM ports over RFCOMM channels.[4]" The COM Port Emulator generates a serial stream of data that can create data flow between devices.

The next layer below that COM Port Emulator are the RFCOMM and SDP. The RFCOMM, which stands for Radio Frequency Communications, Serial Port Emulator "emulates RS-232 serial ports over the L2CAP layer.[4]" The L2CAP layer is the layer that is below both the RFCOMM and SDP. The RFCOMM protocol is based on the TS07.10 specification at the European Telecommunications Standards Institute, or ETSI, website. RFCOMM provides serial-port emulation, so it supports legacy serial-port applications, such as the COM Port Emulator. RFCOMM also includes various functionality, such as being the base for the COM port emulation, implementing multiplexing and flow control, as well as being the transport layer for OBEX protocol over Bluetooth.

The other layer below the COM Port Emulator is the SDP, which stands for Service Discovery Protocol. The SDP "manages the discovery and publishing of supported Bluetooth services and parameters between devices.[4]" The SDP is bound to the L2CAP layer, which is below the SDP. So the SDP contains the actions for both the servers and clients of Bluetooth services. A service is defined here as any feature that is usable by another Bluetooth device. The service works by "the client communicating with an SDP server using a reserved channel on a L2CAP link to find what services are available. Once the client finds the desired service, it requests a separate connection to use the service. This reserved channel is dedicated to SDP communication so that way a device always knows how to connect to the SDP service on any other device.[4]" SDP servers maintain their own database. A database here is "is a set of service records that describe the services that the server offers.[4]"

Below both the RFCOMM and the SDP is the L2CAP, which stands for Logical Link Control and Adaptation Protocol. This layer of the protocol stack communicates directly with the HCI, which is the next layer below this one. The L2CAP "converts data from high-level layers into a format that is supported by lower-level layers of a Bluetooth controller.[4]" It provides many services, one of which is Packet SAR, where SAR stands for Segmentation and Reassembly. Packet SAR "converts packets from high-level layers into packets supported by the Baseband layer.[4]" The baseband layer is one of the lower layers of the stack, so the Packet SAR converts the high-level packets into a form that the lower layers expect. The L2CAP also provides protocol multiplexing and group abstraction. Group abstraction "maps groups of addresses from high-level layers to piconets, which are supported by the Baseband layer.[4]" The multiplexing protocol is used for higher layer protocols, like RFCOMM and SDP, to allow applications to use a single ACL link. The L2CAP keeps track of the data, by using the concept of channels. A channel is just a representation of the data that flows between the L2CAP layers. The L2CAP plays a vital role in the communication between the lower and upper layers of the protocol stack, so it's required in every Bluetooth system.

The layer below the L2CAP, called the HCI, which stands for host controller interface, is the highest lower level part of the protocol stack. This layer "provides the Bluetooth stack with access to a Bluetooth controller. It's an interface for Bluetooth hardware that is responsible for controller management, link establishment, and maintenance. [4]" The HCI also has direct access to the L2CAP layer, and "communicates with a Bluetooth controller through the HCI transport layer. [4]" The transport layer used is the RS-232 Transport layer, and its objective is to "make it possible to use the Bluetooth HCI over one physical RS-232 interface between the Bluetooth Host and the Bluetooth Host Controller. Event and data packets flow through this layer.[5]" So this layer "implements communication between the Bluetooth Host and controller with a small set of functions that send and receive commands, data packets, and events. [4]" Where a Bluetooth Host is what contains the core implementation of the Bluetooth stack, and the Bluetooth controller is what implements the lowest level protocols of the Bluetooth protocol stack. Therefore, since the HCI is the layer below the L2CAP and the highest lower-level of the protocol stack, it acts as a boundary between the lower and upper layers of the Bluetooth protocol stack.

The OBEX, COM Port Emulator, SDP, RFCOMM, L2CAP, HCI are what make up the Bluetooth Host. The HCI transport is what communicates between the Bluetooth Host and the Bluetooth Controller. The Bluetooth controller implements the lowest layers of the Bluetooth protocol stack. The controller consists of the LMP, Baseband and Radio.

The first layer of the Bluetooth Controller is the LMP, which stands for Link Manager Protocol. The Link Manager Protocol "manage logical link establishment between Bluetooth devices, which includes authentication and encryption. [4]" The LMP also "creates. Updates, and removes logical links and logical transports, and updates parameters for physical links to Bluetooth devices. [4]" With the LMP being the first layer of the Bluetooth Controller, which is also connected to the HCI, it's what translates the HCI commands it receives into baseband-level operations. The Bluetooth link that our project will be using is a ACL, or Asynchronous Connectionless link. This type of link is used for data communication, and exits between a master and a slave as soon as a connected is made. The LMP is the protocol in the Bluetooth stack that manages this link.

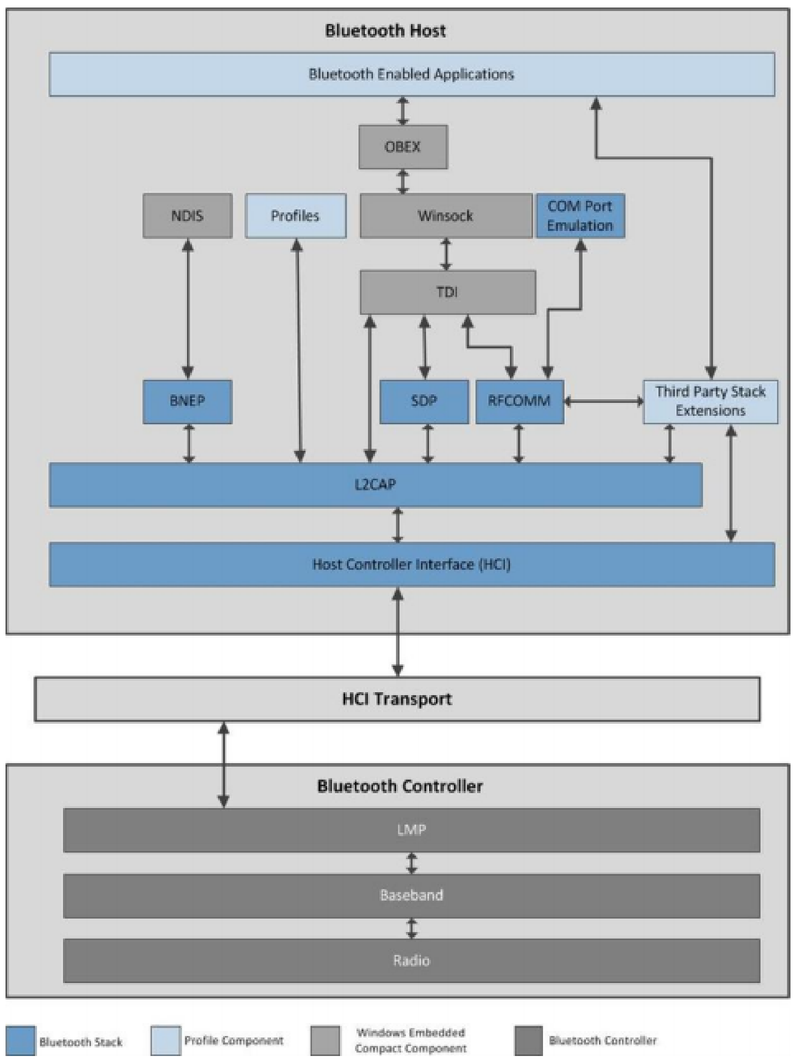


Figure 23: Bluetooth Protocol Stack

The next layer, below the Link Manager Protocol, is the Baseband layer. This layer is "a link controller that forms the physical layer of the Bluetooth architecture. It establishes and manages the physical radio frequency, abbreviated RF, link between Bluetooth units that form a piconet. [4]" The baseband is responsible for formatting the data for transmission to and from the radio layer that lies below it. The link controller part of it is responsible for carrying out the LMP commands and establishing and maintaining the link that the LMP sets out.

The final, and lowest layer of the Bluetooth controller, as well as the Bluetooth protocol stack, is the radio layer. The radio layer consists of a Bluetooth RF transceiver. This layer is responsible for "transmitting data to and from the Baseband. [4]" The RF transceiver is what modulates and demodulates the data into RF signals for transmission in the air. This layer must have certain modulation characteristics, such as radio frequency tolerance, and sensitivity level.

3.3.16. Bluetooth Module Selection

The automatic guitar tuner is going to be able to pair with a mobile device through Bluetooth connection. This requires that a Bluetooth module be selected for our project, so that the two devices will be able to connect to one another. There are a very large amount of Bluetooth modules to choose from, but the Bluetooth module selected should be easy to use, reliable, have low power consumption, and be small. The most attention was given to those factors. The selected microcontroller that is being used for our project is able to work with a UART or SPI interface, however UART interface is the preferred so this is also given attention to determine which Bluetooth module should be selected. The Bluetooth module that gets selected should also be able to exchange data with a high baud rate. The microcontroller being used for this project, the TEENSY 3.2 Mixed Signal Microcontroller, has a voltage range of 1.8-3.6V, as a result the Bluetooth module needs to operate at a voltage in the range of 1.8-3.6V. The module selected must also have a capable communicating range of at least 10 meters, which is a class 2 Bluetooth range, with a larger range being a plus.

Another consideration when selecting a module is the internal settings of the Bluetooth module, and if they can be changed. The internal settings include the modules name, password,and baud rate. The name of the module needs to be changed so that the user is able to easily find and connect to our automatic guitar tuner, and the password needs to be changed to ensure security of our device. By default the baud rate on Bluetooth modules can be set to be low, so this could also need to be changed so that the exchange rate of the data can be higher between the automatic guitar tuner and the mobile phone.

3.3.16.1. HC-05

The HC-05 is a Bluetooth module that is designed for transparent wireless communication setup. It contains the Cambridge Silicon Radio BC417 2.4 GHz Bluetooth Radio chip. This chip is a fully qualified Bluetooth v2.0+ EDR(Enhanced Data Rate) chip. It's compliant with version "2.0.E.2 of specification for both 2Mbps and 3Mbps modulation modes. [7]" This chip is BlueCore4-External. BlueCore4-External is "a single chip radio baseband IC for Bluetooth 2.4 GHz systems including enhanced data rates to 3 Mbps. [7]" BlueCore4-External was designed to reduce the number of external components, this ensures lower production costs. Some keys features of the chip include its radio, transmitter, receiver, auxiliary features, baseband and software, physical interfaces and Bluetooth stack. The radio is a common TX/RX, TX/RX is the abbreviation for Transmit and Receive, terminal that eliminates the need for an external antenna switch. The transmitter offers Class 2 and Class 3 support, which is a range of up to 10 meters, or 33 feet. Some of the auxiliary features of this chip include an on-chip linear regulator and it can run in low power mode. In the software side of the chip, it has an external 8Mbit Flash, and an internal 48 Kbyte RAM. This all allows for full speed data transfer, and full piconet operation. The chip also has a UART interface, which is the preferred interface, making it a viable option to use. The Bluetooth stack runs on-chip in a variety of configurations, including the Standard HCI (UART or USB) stack and the fully embedded to RFCOMM configuration.

The module itself has a dimension of 12.7x27.0mm, that contains an antenna, RF Xformer, used for matching, a 3.3-5V regulator, a 26 MHz crystal, CSR Bluetooth Radio chip, and a 8 MB Flash Memory chip. The module is a low powered Class 2, range of up to 10 meters, device. This range is more than enough for our project. This Bluetooth module has up to +4dBm RF transmit power. Where dBm is an abbreviation for the power ratio in decibels of the measured power referenced to one milliwatt(mW), and 4 dBm converted to mW, is 2.51mW. This is the common transmit power for Bluetooth Class 2 devices. This module also has a power operation of 3.3V, so it will be able to be powered by our selected MCU. "The module contains a UART interface, with a programmable baud rate, and the HC-05 can support up to a 460,800 baud rate. [8]" This is more than enough for our project, and the ease of being able to change the baud rate is an added plus when considering this module for our project.

The module is usually soldered onto a breakout board, which makes it easier to use and connect to our MCU. The best board for the HC-05 is the YourDuino BT board. The board has 6 pins, which are the KEY, VCC, GND, TXD, RXD, and STATE pins. The key pin determines what mode the HC-05 is set too. If it is brought HIGH before power is applied, then the HC-05 is forced into the AT Command Setup Mode. The VCC is used for the power supply, up to 5V, and the GND is the ground pin. The TXD pin is the Transmit Serial Data pin, so it's the pin used to transmit data to the MCU. The RXD pin is the Receive Serial Data, so it's the pin used to receive data from the MCU. Lastly, the STATE pin tells whether there's a connection or not.

The HC-05 has two modes of operation, the Command Mode and the Data Mode. The Command Mode is the mode where AT commands can be sent to the HC-05, while the Data Mode is the mode where the HC-05 transmits and receives data to another bluetooth module. The default mode for the HC-05 is Data Mode, and the HC-05 also has a default configuration. The default configuration is a baud rate of 9600 bps, which means it can transmit 9600 bits per second, with 8 data bits and 1 stop bit. The default password is 1234, and the default name is HC-05. The password and name would need to be changed for better security, and this can be done by getting the HC-05 into the Command Mode of operation. “This is done utilizing the KEY pin by connecting the KEY pin high before applying power to the module. This will put the module into Command Mode at a 38,400 baud rate. [9]” Now the HC-05 module is ready to take commands and the password and name can now be changed using AT commands.

There is a similar module called HC-06, however this module is a slave only device while the HC-05 module is able to be set in either master or slave mode. This adds potential flexibility for our project, and this is one of the main reasons it is considered for use in our project. To get the module into Master mode, the HC-05 module first has to be in Command Mode. After this is done a few commands have to be sent to set the HC-05 mode to Master, resetting the module, a code to allow connections to be made, setting up the PIN, and lastly a command to start searching for devices. The module can be set to Slave mode with a few commands. First by resetting the HC-05 module to its defaults, clearing any previously paired devices, setting the module to SLAVE, and finally displaying the SLAVE address so that it can be known which device is needed to be connected to. The ability to be able to change the password, name, and baud rate, as well as switch back and forth between this module different mode, make it a good contender to use it for our project because of its ease of use and setup.

3.3.16.2. Adafruit Bluefruit LE nRF8001 Breakout

Adafruit’s Bluefruit LE nRF8001 Breakout is another exceptional Bluetooth module looked at for use in this project. This is another Bluetooth module that, when connected to our MCU, will allow for an easy to use wireless connection between our project and a mobile phone. “It works by simulating a UART device beneath the surface, sending ASCII data back and forth between the devices, letting you decide what data to send and what to do with it on either end of the connection. [10]” This module is Bluetooth Low Energy, as opposed to regular Bluetooth, which means this module uses low power, thus making it an ideal pick to use for our project. The application being made to go along with this project will be for Android devices, and Android supports Bluetooth Low Energy, so it would be reasonable to be able to use this Bluetooth module and not encounter any problems.

Also classic Bluetooth has “lengthy contracts and loops to go through to create iOS/Android peripherals, so that you’re able to legally design and distribute in the App Store. [10]” Bluetooth Low Energy doesn’t have these issues, which makes it a great choice compared to classic Bluetooth which still has a lot of restrictions around it on the iOS/Android platform. [10]” The nRF8001 Breakout module handles all the BLE radio and low level work, while doing it all over SPI. This allows it to be able to be easily used with any kind of microcontroller. All the pins needed are on the bottom of the PCB of this module, and they're all 5V compliant. This gives the user the option to use this peripheral with 3V or 5V microcontrollers. Though the supply voltage for this module is 1.9-3.6V, and since our selected microcontroller runs at 3.3V this works out nicely. Most MCUs run at 5V, but since ours doesn't, no voltage divider will need to be added to drop the voltage down to safely run this module.

The module itself has a dimension of 29x28mm, this puts it at a similar size to the HC-05. "The module is available in a compact 5x5 mm QFN32 package. [11]" The "backplate of the QFN32 capsule must be grounded to the application PCB in order to achieve optimal performance. [11]" This module presents an ease of use for adding Bluetooth to our project as it's integrated with a Bluetooth Low Energy compliant radio, slave mode link controller and host. The nRF8001 also has an on-chip non-volatile memory for storing service configurations, and offers an optional output signal (ACTIVE) that is activated before the radio becomes active. This timing signal allows the user to be able to control the peak current drain, which means it won't allow an overload of the power supply. This signal can also be utilized to control application circuitry, thi avoids noise interference when the nRF8001 radio is running. The separate UART interface gives the user access to the direct test mode, and in this UART interface the user can control the BLE radio, test radio performance and optimize the antenna. The radio of the nRF8001 features a Bluetooth Low Energy RF transceiver, common TX/RX terminals, and low current used during connection. The module also features power management, a battery and temperature monitor, and an integrated 16 MHz crystal oscillator, while also having the correct interfaces preferred for our project.

The nRF8001 Breakout module has 10 total pins, the SCK, MISO, MOSI, REQ, RDY, ACT, RST, 3Vo, and GND. The SCK pin is the SPI data clock. The MISO is the SPI data out pin, this is the pin that the module uses to send out data to the MCU. The MOSI is the SPI data in pin, this is the pin that receives data from the MCU. The data level on the pins is 3V, which is fine for our project as the selected MCU operates at 1.8-3.6V. The RDY is the ready pin, and it is the pin that lets the user know that data is ready to be read. The ACT(active) is an output from the module, it's what lets the host know when the module is buy. The RST is the reset pin. The 3Vo is the output from the 3.3V regulator on board. Lastly the GND is the ground pin for data and power. Overall this module offers exception features, as well as being easy to use, which make it another quality bluetooth module to use for this project.

3.3.17. Adafruit Bluefruit LE UART Friend – Low Energy (BLE)

Adafruit's Bluefruit LE UART Friend, with Bluetooth Low Energy, is a powerful and easy to use Bluetooth Low Energy device. This is another Bluetooth module that, when connected to our MCU, will allow for an easy to use wireless connection between our project and a mobile phone. The LE UART Friend is the updated version of the LE nRF8001 Breakout. The main difference between the two is that the LE UART Friend can advertise as several different kinds of devices, and there’s also an interface on the UART Friend that lets the user set connection parameters. Overall the UART Friend is more flexible than the nRF8001 Breakout module.

The LE UART Friend has multiple functions, but for our project the standard UART TX/RX connection profile would be mainly utilized, if selected. “In this profile, the Bluefruit acts as a data pipe, that can transmit data back and forth from an iOS or, in our projects case, Android device. [12]” The LE UART Friend also has an easy to use and learn AT command set. This gives the user full control over how the LE UART Friend behaves. The LE UART Friend has impressive specifications. The LE UART Friend has a dimension of 21x32mm and only weighs 3.4g, as well as a 256KB flash memory, as well as a 32KB SRAM. Flash memory is a type of memory that retains data in the absence of a power supply, and SRAM is a type of memory chip that’s faster and requires less power than dynamic memory does. Both of these parameters are more than enough for our project. The LE UART Friend also boasts a peak current draw of only 20mA, for radio when actively transmitting/receiving data. The LE UART Friend also has a 9600 baud transport rate, same as the nRF8001 Breakout module. The LE UART Friend has 5V safe pin with an on board 3.3V voltage regulation, however since our selected microcontroller has a max voltage output of 3.6V, we don’t have to worry about this Bluetooth module getting overpowered or burnt out. Another exceptional specification this module offers is the easy AT commands to set up and run this module quickly, easier than the other Bluetooth modules discussed above.

The Bluefruit LE UART Friend has two modes, command mode and data mode. How to get the module set into each mode is described in the following paragraph. In UART Data mode, the module can be utilized as a transparent UART connection to the app. This makes the data transfer simple. Data gets sent to the app when any 9600 baud data is received on the RXI pin and any data from the app is automatically transmitted via the TXO pin. [12]” Overloading the microcontroller can be a problem, to keep this from happening the flow control pins can be utilized. This is done by keeping the CTS pin high until the device is ready for more data, then grounding the CTS pin. This lets the module know that it can send more data. If the LED on the module blinks two times followed by a three second pause, then it’s in Data mode. UART Data mode gives the ability to connect the app and send/receive data transparently.

Bluefruit LE UART Friends other mode, the Command mode, gives the user the ability to enter a variety of AT style commands to configure the device or retrieve basic information about the module of Bluetooth Low Energy connection. [12]” Similar to UART Data mode, overloading the microcontroller can be a problem, to keep this from happening the flow control pins can be utilized. This is done by keeping the CTS pin high until the device is ready for more data, then grounding the CTS pin. This lets the module know that it can send more data. If the LED blinks three times followed by a three second pause, then this means the module is in Command mode. In the Command mode, the Bluefruit LE UART

Friend uses a Hayes AT-style command set to configure the device. The commands can be found on the Adafruit website, so they are easily attainable and can be easily learned. There's also a built-in HID keyboard that can be used to send keyboard data to any BLE-enabled Android device. This adds more flexible and ease of use if this module is selected for use in our project.

Unlike the nRF8001 Breakout that has 10 pins, the LE UART Friend only has 8 total pins. The eight pins are VIN, GND, TXO, RXI, CTS, RTS, MOD, and DFU. The VIN pin is the pin that provides the power to the Bluetooth module. This module needs to be supplied with, at the minimum, 3.3V, this won't be a problem as our selected microcontroller runs at this voltage, so this module would get powered easily. The GND is the common ground for the modules power and logic. The TXO is the UART transmit pin out of the LE UART Friend, so this is the pin that would transmit data to the MCU, and it has a 3.3V logic level. The RXI is the UART receive pin into the LE UART Friend, so this is the pin that would receive data from the MCU, and it has a 3-5V logic level. The CTS is the clear to send hardware flow control pin into the LE UART Friend. This pin is used to tell the Bluefruit LE UART Friend that it can end data back to the microcontroller over the TXO, transmit, pin. To be able to enable data transfer out, this pin will need to be changed from its default pulled high position to ground. The CTS pin uses 3-5V logic, like the RXI pin. The RTS pin is the read to send flow control pin out of the Bluefruit LE UART Friend. At 9600 baud transmit rate this pin isn't needed, and, if needed, the RTS pin is a 3.3V logic level. When set low the DFU pin forces the Bluefruit LE UART Friend module to enter a firmware update mode to update the firmware over the air. Though, once powered up this pin can also be used to perform a factory reset. Lastly, the MOD pin is the mode selection pin. The Bluefruit LE UART Friend has two modes, Command and Data, similar to the HC-05. The modes can be set by either using the slide switch to select the mode, or by setting the pin voltage. To use the slide switch to select the mode, the pin simply needs to be disconnected. To set the mode by setting the pin voltage, set the pin High to go into Command mode, or set the pin Low to be in UART/DATA mode. This pin is level shifted, and uses 3-5V logic. The nice thing about the Bluefruit LE UART Friend is that the user can set up which pins are to be utilized for TX, RX, and CTS flow control, as well as the Mode and RST pins.

While the Bluefruit LE nRF8001 Breakout uses a nRF8001 BLE module, the Bluefruit LE UART Friend uses a nRF51822 BLE module. The "nRF51822 is built around ARM Cortex M0 CPU with 256kB/128kB flash plus 32kB/16kB RAM for improved application performance. [13]" The Bluefruit LE UART Friend comes with the 256kB flash memory plus 32kB RAM, as stated above. The embedded 2.4 GHz transceiver inside the module supports Bluetooth Low Energy. The "nRF51822 incorporates a rich selection of analog and digital peripherals that can interact without CPU intervention through the Programmable Peripheral Interconnect, abbreviated PPI, system. [13]" A flexible 31-pin GPIO mapping scheme allows Input/Output interfaces to be mapped to any device pin, like our selected microcontroller. This enables the user complete design flexibility with pin-out location and function. This module has 3 data rates of 2Mbps, 1Mbps, or 250kbps, with a +4dbm outpower. This BLE module also features low power consumption, as at +4dbm it only has a power consumption of 11.8mA when using on-chip DC-DC at 3V. This BLE module on the Bluefruit LE UART Friend offers exceptional features as well as exceptional power management. Overall the Bluefruit LE UART Friend is another strong option to use as a Bluetooth module for our automatic guitar tuner project.

Table 26: Bluetooth Module Comparison

Spec	**HC-05**	**nRF8001 Breakout**	**BLE UART Friend**
Dimension(mm)	12.7 by 27	29 by 28	21 by 32
Power needed for Operation(V)	1.8 to 3.6	1.9 to 3.6	1.8 to 3.6
Power consumption(mA)	30	25	20
Baud rate(bps)	Default: 9,600 Up to: 460,800	9,600	9,600
Weight(g)	9.6	1.8	3.4
UART capable	Yes	Yes	Yes

The main factors in deciding the Bluetooth module are size, power consumption, baud(transfer rate), UART interface, and ease of use. The HC-05 module wins in size, though it's only slightly smaller than the BLE UART Friend module. All three modules have a transfer rate of 9,600 bits per second(bps), though the HC-05 can go much higher than this, but 9,600 bps is fast enough for our project. All three modules also have a UART interface, have small power consumption levels, and can each be powered by our selected microcontroller. Each module is easy to use with plenty of tutorials, examples, and lists of AT commands that can be easily found and learned. These modules don't have much to separate them, except for the HC-05 being a slave and master device, while the other two are not, however the other two modules are both Bluetooth Low Energy devices so they use less power. The other two modules would also be able to be easily soldered onto our overall PCB, as the nRF8001 and BLE UART Friend both have through hole pins, while the HC-05 does not and thus would be harder to solder onto the overall PCB design. This as well as the nRF8001 and BLE UART Friend both utilizing Bluetooth Low Energy technology ultimately eliminate the HC-05 as a selection. The nRF8001 Breakout is eliminated because the Bluefruit BLE UART Friend is the updated version of the nRF8001, thus making the Bluefruit BLE UART Friend the Bluetooth module our project decided to go with.

3.4. Parts Selection Overview

Below is a chart of all selected major products and their manufacturers. For more information of each system, please see the corresponding section for each system in section 3.3. Each of these parts was chosen according to a series of comparisons among other contenders and should display the best possible choice with the knowledge at hand. These parts are to be tested rigorously in each of their individual use cases to fully analyze their performance against expectations. This testing will be done at both software and hardware levels. The testing takes place in section 7.

Table 27: Part Selections

System	**Chosen Manufacturer**	**Chosen Product**
Battery Charging IC	Texas Instruments	BQ2057CSN
Microphone	Mallory Sonalert Products Inc.	PMO-6027P-40LDQ
Battery	LG	INR18650-HG2
H-Bridge	Texas Instruments	SN754410
Linear Regulator	STMicroelectronics	LM7805CV
Linear Regulator	STMicroelectronics	LD33V
Motor	uxcell	Standard Gearmotor 60 RPM
Display	Adafruit	Monochrome OLED 0.96”
Microcontroller	PJRC	TEENSY 3.2
Operational Amplifier	Texas Instruments	TL084CN
Bluetooth Module	Adafruit	Bluefruit BLE UART Friend

4. Related Standards and Realistic Design Constraints

Standards are usually seen as an industry level of quality of attainment. There are numerous standards across all industries which outline proper design practice and help consumers receive a better end product. Many of these standards cost money to acquire, and thus, are out of the budget constraint of this project. This is because standards are marketed towards large companies where purchasing and following a standard is considered liability management. Some standards are provided free of charge, and are detailed below.

4.1. Related Standards

The main standards covered in this project revolve around battery safety, wireless communication, and security. Batteries are safe when treated properly, but can turn into fire hazards when improperly designed. Wireless communication is heavily regulated by the Federal Communications Commission (FCC) because it only takes one device malfunctioning to jam communications to all devices in the nearby area. Security is also taken seriously to protect user information and prevent malware from gaining access to the user's smartphone.

4.1.1. Wireless Communication Standards

Wireless communication is an important part of any part of modern technology. The past decade alone has seen the development of high-tech smartphones which have expanded our ability to interact with computers. As such, many standards have been developed which aim to create easy to use systems for inter-computer communication.

4.1.1.1. Networking (Wireless) Standard

802.11	Wireless	Basics of physical and logical networking concepts.
802.11a	Wi-Fi	**1)** Specifies a PHY that operates in the 5 GHz U-NII band in the US - initially 5.15-5.35 AND 5.725-5.85 - since expanded to additional frequencies **2)** Uses Orthogonal Frequency-Division Multiplexing **3)** Enhanced data speed to 54 Mbps
802.11b	Wi-Fi	**1)** Enhancement to 802.11 that added higher data rate modes to the DSSS (Direct Sequence Spread Spectrum) already defined in the original 802.11 standard **2)** Boosted data speed to 11 Mbps **3)** 22 MHz Bandwidth yields 3 non-overlapping channels in the frequency range of 2.400 GHz to 2.4835 GHz **4)** Beacons at 1 Mbps, falls back to 5.5, 2, or 1 Mbps from 11 Mbps max.
802.11g	Wi-Fi	**1)** Extends the maximum data rate of WLAN devices that operate in the 2.4 GHz band, in a fashion that permits interoperation with 802.11b devices **2)** Uses OFDM Modulation (Orthogonal FDM) **3)** Operates at up to 54 megabits per second (Mbps), with fallback speeds that include the "b" speeds
802.11n	Wi-Fi	**1)** Higher-speed standards **2)** Several competing and non-compatible technologies **3)** Top speeds claimed of 108, 240, and 350+ MHz
802.11ac	Wi-Fi	**1)** Top speeds claimed of 54Mbps, 5GHz standard **2) MU-MIMO** (multi-user MIMO) supports up to 4 clients **3)** Modulation of QAM-256 (compare QAM-64 in 802a – g)

Figure 24: IEEE 802.11 Version Concepts

The Network, or IEEE 802.11, Standard, is the IEEE Standard for Information technology. It specifically deals with telecommunications and information exchange between systems, such as local and metropolitan area networks. The purpose of this standard “is to provide wireless connectivity for fixed, portable, and moving stations within a local area [6].” The 802.11 Standard consists of a family of standards and amendments. The first was 802.11-1997 standard of the family, however 80.11b was the first widely accepted standard. 802.11b deals with the 2.4 GHz ISM, which stands for industrial, scientific and medical, radio bands, which is the band that Bluetooth deals with. To avoid interference from other forms of communication that uses this band, 802.11b uses DSSS, which stands for direct-sequence spread spectrum, and orthogonal frequency-division multiplexing, or OFDM. 802.11b is in the 2.400-2.500 GHz spectrum, and each spectrum is divided into channels. Each one of these channels has a center frequency and bandwidth. This particular band is divided into 14 channels, spaced 5 MHz apart, with a center frequency of 2.412 GHz. In addition to specifying the channel center frequency, 802.11 specifies a spectral mask. The spectral mask defines the power permitted to each channel. The mask requires the signal be a minimum of 20 dB from its peak amplitude at plus or minus 11 MHz from the center frequency. This makes the channel 22 MHz wide. 802.11b was enhanced to 5.5 and 11 Mbit/s in 1999.

4.1.1.1. Bluetooth Standard

Bluetooth is a wireless technology standard for exchanging data over short distances using short wavelength radio waves in the 2.4 to 2.485 GHz range. Bluetooth is what’s being relied upon for communication between the automatic guitar tuner and mobile application. The original Bluetooth standardization was overseen by IEEE and called IEEE 802.15.1, however IEEE no longer overlooks the Bluetooth standard. The Bluetooth SIG now oversees the development of the Bluetooth specification. Bluetooth devices are able to connect to one another wirelessly to exchange data between one another. If a device were being developed and wanted to have Bluetooth capabilities, then the person or company developing it would have to go through a qualification process. The qualification process is overseen by the Bluetooth special interests group(SIG), and it’s done to make sure the device conforms to the Bluetooth specifications and to strengthen the Bluetooth brand. The primary objective of the process is for the developer to demonstrate that their product is compliant to the correct Bluetooth specifications. For our project this won’t have to be worried about. Though a device is being made to have Bluetooth, the Bluetooth chip that is selected has already been through the process, we are simply adding the chip to our design to use it’s Bluetooth capabilities. Thus this standard won’t affect our project.

4.1.1.2. RS-232 Standard

RS-232 is a telecommunications standard for the serial communication of transmitted data. This standard specifies hardware and software implementations for serial communication between two devices. In other words, “RS-232 is defined as the “interface between data terminal equipment and data communications equipment using serial binary data exchange. [14]” RS-232 was introduced in 1960 by the Electronic Industries Association(EIA). Recently, however, RS-232 compared to other serial interfaces has slower transmission speed, short cable length, and a large voltage swing. USB has displaced RS-232 from most of its peripheral interface roles, though the RS-232 is still used in some devices today in networking equipment.

4.1.2. Wired Communication Standards

Wired communication standards set the expectations for internal system communication within a device. For the purpose of this project, only serial communication will be discussed. There are two main forms of serial communication; synchronous transmission, and asynchronous transmission

4.1.2.1. Synchronous Transmission

Synchronous transmission uses a shared clock line between two devices in order to ensure the devices are temporally linked. Data is sent from a master device in the form of 1s and 0s each time the clock line flips. The slave device reads the clock line and samples in between clock flips to determine if the master is sending a 1 or 0. There are two main categories of synchronous communication which are used in the components chosen above; inter-integrated circuit (I2C) and Serial Peripheral Interface (SPI).

I2C is a multi-master and multi-slave transmission protocol where data is requested and sent in packets of bytes. [6] It was originally developed by Philips Semiconductor (now NXP Semiconductor) to enable short range intra-board communication between microprocessors. The standard has since been adopted by more than 50 companies and is the de Facto standard for short range synchronous serial communication. There are two main signals which need to be shared between devices to enable communication. The Serial Data Line (SDA) is used to send data in byte-long packets from one device to another. The Serial Clock Line (SCL) is used to send a consistent clock signal. The master device controls the clock signal on the bus and syncs its sampling to the clock. When a master device wants to communicate with a slave device, it generates a clock signal on the SCL bus. Each slave on the line starts listening for data, and the master sends out the address of the specific slave device. The particular slave device corresponding to the address sent will then reply that it is present. The master will either request data from the slave, or request the slave to receive data. The slave brings the clock line to zero until it is ready for either scenario, where it allows the clock line to raise backup and data is transferred.

SPI is very similar to I2C, though it uses a different master-slave selection protocol. SPI devices are typically four port devices, with Master Out Slave In (MOSI), Master In Slave Out (MISO), Serial Clock (SCLK), and Slave Select (SS) channels. SPI busses operate with one master which controls the SS, SCLK, and MOSI lines while monitoring the MISO line. SPI devices can communicate in both directions at the same time. This bi-directional communication ability is known as full duplex. This communication standard is also much more flexible, allowing for customizations in inter-device communications. This allows custom serial interfaces to be configured and allows bit-wise communication as opposed to byte-wise communication for I2C.

Overall, I2C and SPI are very similar, but they both have their special use cases where they perform better than the other. I2C is useful for scenarios where there are many devices (up to 127) which need to communicate together. The I2C protocol also allows for any of these devices to act as a slave or master, which simplifies the process of data handling greatly. SPI has the benefit of greater customizability which can make repetitive communication much more efficient if the data can be split into smaller-than-byte segments. SPI is also more useful for scenarios where only one master and few slaves are on the bus, which allows for single selection signals to be sent instead of full addresses. SPI, finally, has the ability to send data in full duplex, which doubles the throughput of the system when data is efficiently managed.

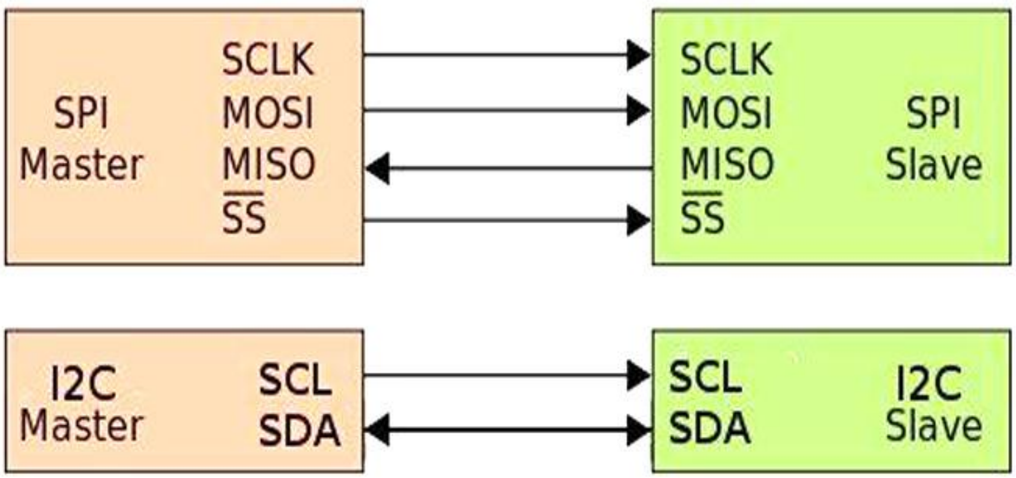


Figure 25: SPI Vs I2C

4.1.2.2. Asynchronous Transmission

Asynchronous transmission is generally used in the form of Universal Asynchronous Receiver/Transmitter (UART). UART generally only requires two pins to communicate in both directions, Tx and Rx along with a common ground. A single pin configuration can be used when data only needs to go in one direction, which is called simplex. This type of system is known as asynchronous because there is no common clock line being transmitted between devices. Devices are connected together from Tx to Rx, where the Tx side transmits, and the Rx side receives. Both devices are configured to a common clock speed at which bits are transmitted from the Tx pin. The receiving side samples the signal at the Rx pin at a multiple of the set clock speed (often 8x) and correlates the samples to bytes of data. Due to the lack of clock line, UART devices send start and stop bits which are configurable on either end. These lines of bits signify the start and stop of the message, where the total sample is then compiled into the proper data type. UART has been updated and used in various standards such as RS-232, RS-422, and RS-485. Modern systems can use UART to transfer information as fast as 460 Mb/s.

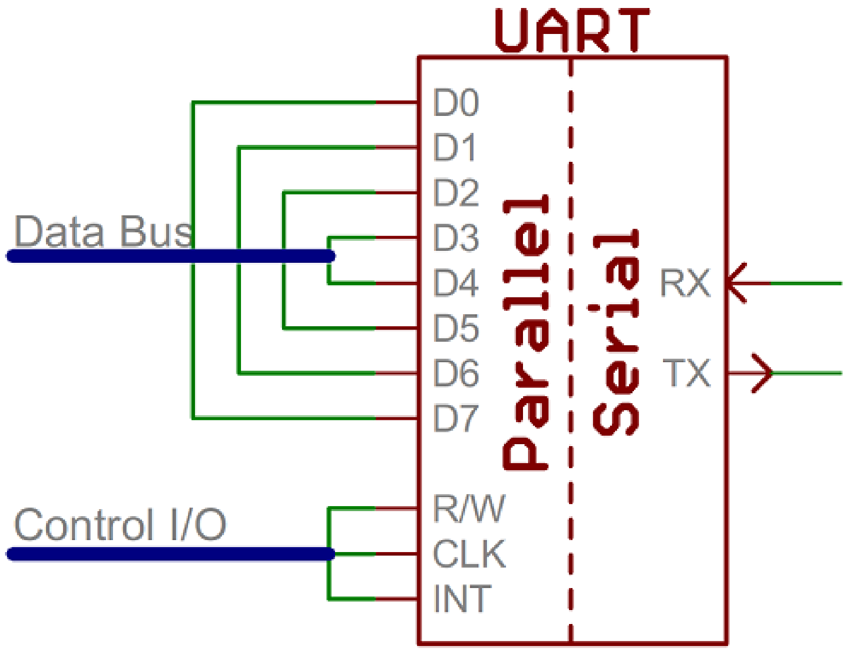


Figure 26: UART

4.1.2.3. Universal Serial Bus (USB)

Universal Serial Bus' (USB) are the current standard for interfacing between devices. They provide a convenient and generic, or universal, way for electronics manufactured by separate companies to maintain data transfer compatibility. Uses of USB's range from audio transfer to battery charging. Multiple physical configurations of USB exist to help tailor to the needs of a device's dimensions.

Type A is the most common USB port, and is available on almost all modern personal computers. The Mini versions of the port were the first attempt at minimizing the design to accommodate the rapidly shrinking size of modern electronics. The Mini versions have since been replaced by Micro USB's. The Micro versions are an even smaller design than the mini, and are commonly found on handheld devices.

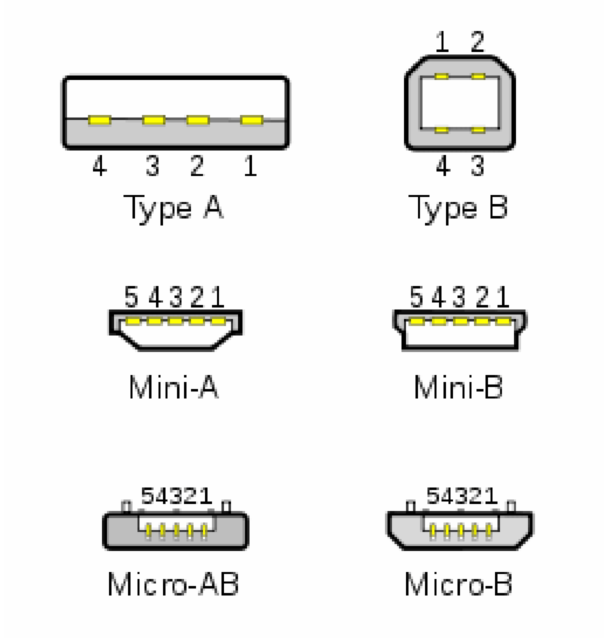


Figure 27: USB Standards

4.2. Realistic Design Constraints

When designing a system, there are constraints that must be considered. These constraints are vital to the design to provide a quality product. These constraints were carefully selected to ensure applicability and feasibility of the constraint.

4.2.1. Economic and Time Constraints

Budget is a constraint every project runs into. In this design the economical constraint will limit the quality of electrical components used in this design. Some features may need to be eliminated due to economic reasons. The budget for this project is provide by this group and set at $320.
Time constraints will also play a significant role in part selection and testing procedures. Due to lead times, desired parts may need to be substituted. Multiples of each part will also be ordered to eliminate procurement issues if something is broken.

4.2.2. Environmental, Social, and Political Constraints

Environmental constraints affect this design as well. Tuning knobs for stringed instruments come in all different sizes and shapes. The design will need to be able to turn all types of knobs, therefore making the design significantly harder.

The impact of social constraints will also play a significant role. The application for this device will be designed for android devices. This means that the charging cable for the device should be a cable most if not all android users should have. This means that we must use the Micro - B USB plug, which is the standard for android cell phone chargers.

4.2.3. Ethical, Health, and Safety Concerns

When communicating with people's cell phones, there will be vulnerabilities and pressures involved. Due to the ethical constraints placed on ourselves, there will be no advertisements or data collection from the phone application. When communicating, we need to ensure that the communication is secure.

We need to ensure that this product is designed with ergonomics in mind. The weight and size of the product, as well as the shape will all affect how easy this product is to use. If not, the product may cause some stress in joints and muscles. This stress could lead to injury if repeated enough or if the user stresses the same area at a different time.

Safety needs to be constantly considered for both the stringed instruments the tuner is tuning, as well as the operator. The servo that turns the tuning knobs on the instrument must not damage the instrument in any way. The possible damages include breaking the tuning knob, breaking strings, scratching the head board, etc. If the servo motor on the device is too strong and if it catches something not allowing the tuning knob to turn anymore it may twist the wrist of the user. This could cause injury.

4.2.4. Manufacturability and Sustainability Constraints

When choosing parts, we need to ensure that they will be available for future procurement in batches that could allow for mass production. We also need to consider the pricing of both the price of a single unit of each product as well as the bulk pricing. If there is significant pricing discrepancies, the bulk pricing should be weighted more. This will allow for maximum profits should the device be put to market. We should also not include special pricing due to the fact we are students or any short term sales. This will not translate to mass production.

4.2.5. Mobile Application Development Constraints

For the scope of this project, the mobile application is restricted to Android development due to the more stringent constraints that come into play when developing for iOS. Both Android and iOS have their own Integrated Development Environment (IDE), Android Studio and Xcode respectively.

Apple's Xcode IDE runs exclusively on Mac OS, short of a virtual machine, it is impossible to develop applications with Xcode on another OS such as Windows or Linux. Xcode does support development using a variety of programming languages. The common supported languages are Apple's own language named Swift, Objective-C, Javascript, Python, and a variety of other languages. Since Swift is Apple's language, it is the most well documented when it comes to mobile development and is the most widely used language for iOS development.

Android's Android Studio IDE is compatible with Windows, Linux, and Mac OS. The primary development language is Java although XML can be used to a large extent. Even though Android Studio does not support as many languages as Xcode, it supports a single language very well and Java is one of the world's most powerful and widely used programming languages and is extremely well documented which is a clear strength over Xcode's exclusive Swift language.

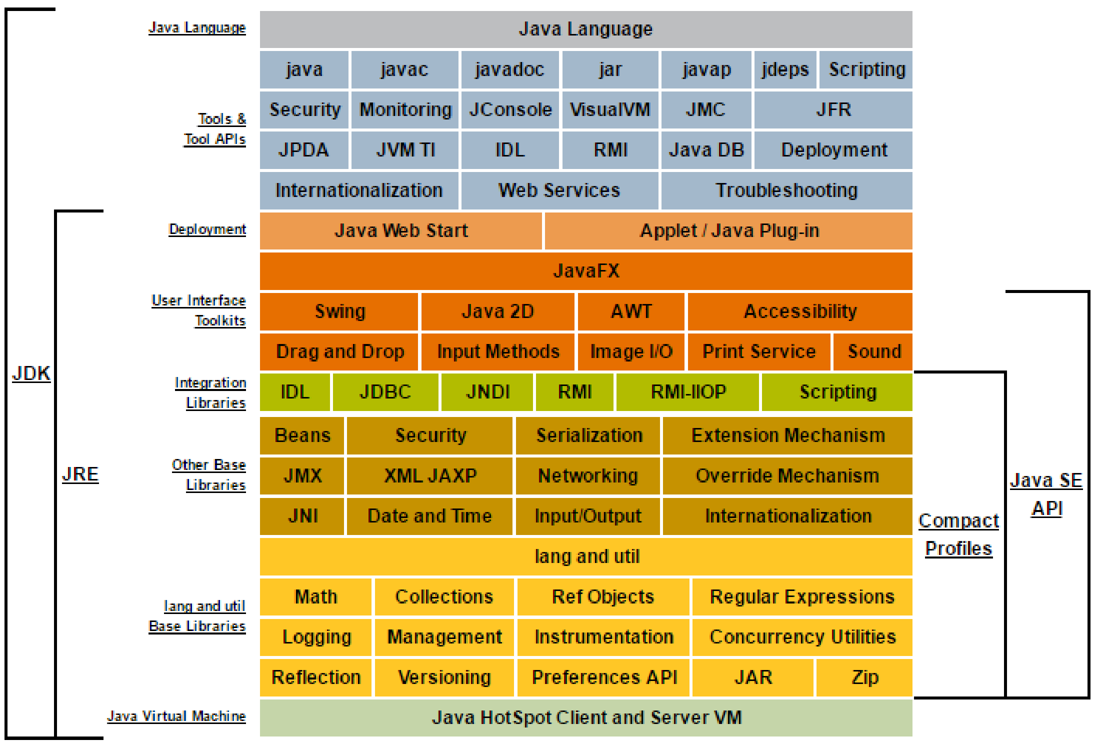


Figure 28: Java Platform Standard Edition 8 Documentation [22]

Android is also known for being much more approachable towards independent app developers. Via their website, Android provides very specific and detailed guides to help with app design. This not only includes common programmable components like fingerprint scanner or gesture recognition, but also more art-based principles such as organizing the layout and deciding on the style of the app. Apple does provide similar guides, but they are not nearly as exhaustive.

The final, and perhaps the most important, factor towards choosing Android over iOS is development ease of access. Apple holds its app store to a much higher standard than Android when it comes to what apps it will allow in the store. That is both a good and bad thing. While it promotes quality app content, it can be a deterrent to independant developers.

iOS charges an annual $100 to host an app within their store. In addition, the app must go through a review process. While the app review process is generally fairly short, an average of two days, it is not cohesive with a time limited development schedule. If the app does not pass the review process, which consists of a large list of requirements, it is declined and the developer must make changes then resubmit the app for review. On the other hand, Android requires a one time payment of $25 and no review process, the app is immediately added to the Google Play Store. Due to Android's superior accessibility, it is the clear choice for this project's mobile application needs.

With regards to Android OS version, not all OS versions are supported by the app. This is due to version constraints. Older OS's do not provide the same library support as newer versions. Therefore, the android application is designed to be compatible with the most popular Android OS versions, 4.1.x and above. The figure below shows the version distribution of Android OS users.

Version	Codename	API	Distribution
2.2	Froyo	8	0.2%
2.3.3 - 2.3.7	Gingerbread	10	3.0%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	2.7%
4.1.x	Jelly Bean	16	9.0%
4.2.x		17	12.2%
4.3		18	3.5%
4.4	KitKat	19	36.1%
5.0	Lollipop	21	16.9%
5.1		22	15.7%
6.0	Marshmallow	23	0.7%

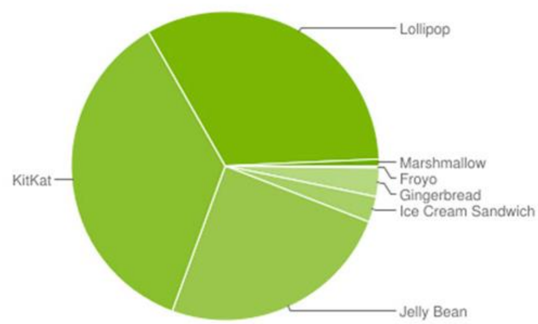


Figure 29: Android Market Share Spread [23]

5. Project Hardware Details

This section refers to Hardware as the physical circuits and integrated circuits which are used to perform all functions of the device. This section breaks the design into a few subsystems and describes their function.

5.1. Initial Design Architectures and Related Diagrams

The first assignment for this design project was to create a document called “Divide and Conquer” where this project was proposed and broken down into the initial steps needed to achieve a successful result. With the initial research into existing products on the market and what we wanted to produce, the initial diagram was created showing the main subsystems of this project.

This initial brainstorm is subject to future change, but thus far, has held up to further research. This organization diagram is subject to change upon further research and testing, but it is important to document the initial overview of what was expected for comparison to the final products.

The assignment of responsibility is also subject to change, with each team member contributing with our collective knowledge gained in the past stay at the University of Central Florida. Some parts of the project have proven to require much more effort, whereas others have proven more simple to implement than previously expected. With this in mind, the division of work is kept at a fluid level, where information is shared among the whole group, and effort can be applied as needed.

The following three sections include design aspects for each subsystem, including how it is split up and why it is necessary for the success of the overall design. The overall system is split into three subsystems: User Interface, which includes buttons, display, and Bluetooth module. Charging and Power Regulation, which includes battery, charging circuit, and regulators. Signal Processing and Control, which includes sound feedback, MCU, and motor driver signaling.

5.2. First Subsystem – Battery and Power Regulation

The battery and power regulation circuit is the key to providing a fully wireless handheld device. The main design goals of this section are to provide fast charging via a micro-USB charging port, user protection via built in safety circuits, and adequate power provision for motor and logic components.

5.2.1. Battery Charging

The first stage of power comes from an LG INR18650 HG2 lithium ion battery cell. This cell provides a nominal 3.6V output and can source up to 20A of current. The datasheet also specifies a 0.5 C charge rate (1500mA) which means the battery is capable of being charged in the three hours specified in the requirements. The battery charging IC, the Texas Instruments BQ2057CSN, was found to be adequate for this subsystem due to its 4.5V-15V input range and 2A charge rate capability. This IC also provides a thermistor input which can be used to monitor battery cell temperature levels and shut down if an over or under temperature condition is detected along with status LED and cell conditioning to bring damaged cells back to operating voltages.

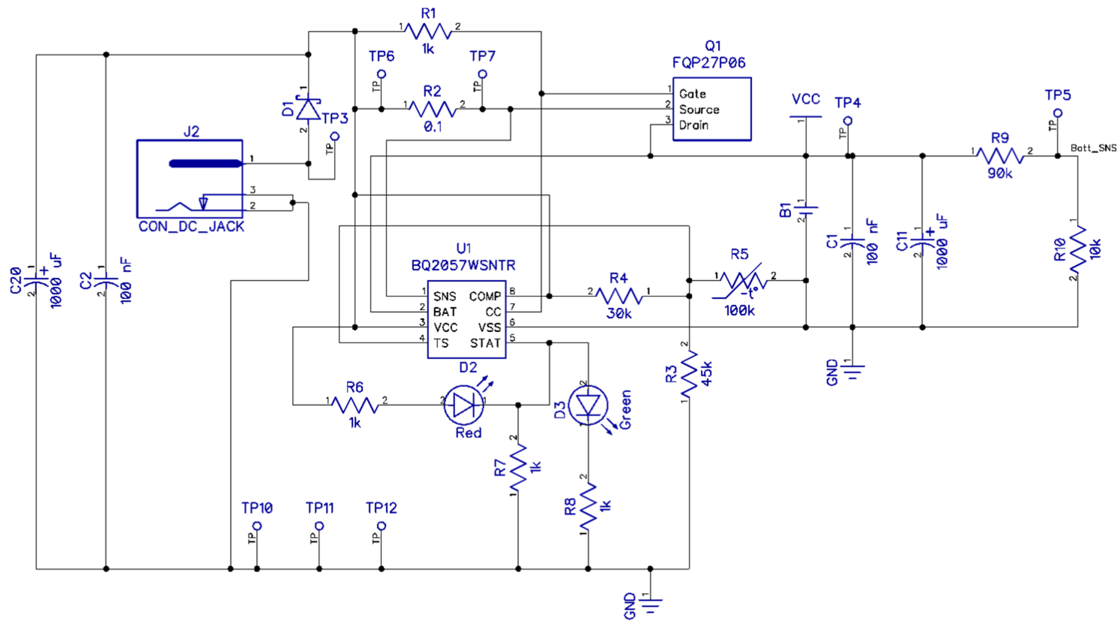


Figure 30: Battery Charging Circuit

The above circuit displays the bq2057C linear lithium ion charging circuit. This circuit is configured to achieve all requirements for the battery section, most importantly, the 3 hour maximum charge time, which equates to a constant current of 1000 mA. The bq2057C IC regulates the current using the CC pin to drive the gate of a P-channel MOSFET. The device aims to keep the voltage drop across the sense resistor, R1, at 100 mV. In order to choose the correct value Rsns, the following equation was used.

$$Rsns = Vsns/Iout = 0.1/1 = 0.1\Omega$$

The circuit provides a status pin which is used in a source/drain configuration to switch two LEDs. When the circuit is charging, the green LED is on and the red LED is off. When the circuit is done charging, or the temperature is too high, the green LED is off and the red LED is on.The charging circuit provides a temperature feedback input which can turn off charging should it exceed calculated values. Temperature feedback is provided by a 100k NTC thermistor which is placed in a voltage divider configuration with two calculated values. The selected cutoff temperatures were chosen to be 70℃ and -20 ℃ respectively.
The selected thermistor had the following values for Rth and Rtc.

$$Rth = 17.5k\Omega$$
$$Rtc = 990k\Omega$$

The following equations were used to calculate the required values of R4 and R5.

$$RT1 = R4 = (5*Rth*Rtc)/(3*(Rtc-Rth)) = 29.69k\Omega$$
$$RT2 = R5 = (5*Rth*Rtc)/((2*Rtc)-(7*Rth)) = 46.6k\Omega$$

These values were rounded to 30kΩ and 40kΩ respectively and should provide for safe operation of this circuit.

5.2.1.1. Power Regulation

After charging, the battery is tasked with providing power to the entire system, including the motor and all of the logic. After doing research and selecting components, three regulated voltages need to be created from the single nominal battery potential. These voltages are split by their intended use: The motor needs 7V, the logic needs 5V and 3.3V. The 7V and 5V supply lines are a little tricky to provide since they are higher than the nominal 3.6V provided from the battery cell. A boost circuit is required to step the DC voltage up. Since the motor is the only load on the 7V supply, there is a higher allowable ripple voltage when compared to the more sensitive components on the 5V and 3.3V rails. With this in mind, the following ICs were chosen:

Table 28: Power Regulation ICs

Function	Manufacturer	Part Number
Linear Regulator 5V	STMicroelectronics	LM7805CV
Linear Regulator 3.3V	STMicroelectronics	LD33V

This power regulation system needs to be capable of supplying enough current to all other electrical subsystems. These subsystems are split by their specified input voltage characteristics and are listed in sorted into tables for reference when designing the selected power regulator.

Table 29: 5V Current Draw

Parts (5V)	**Current Draw**
Teensy 3.2	35 mA
TL084CN	60 mA
579-MCP3221A5T-I/OT	250 uA
OLED	20 mA
Total (5V)	115.25 mA

Table 30: 7V Current Draw

Parts (7V)	**Current Draw**
5V	115.25 mA
Motor	500 mA
Total	615.25 mA

The first stage of power regulation is the boost converter circuit which steps the 3.6Vdc nominal battery voltage up to 7Vdc. This voltage is used to drive the motor, and also feeds powers the two linear regulators. The two linear regulators are used to step the voltage down to 3.3V and 5V respectively. Linear regulators provide low noise outputs at the cost of efficiency. Since the 3.3V and 5V rails are only being used to power logic, this lack of efficiency is acceptable and does not decrease battery lifetime significantly.

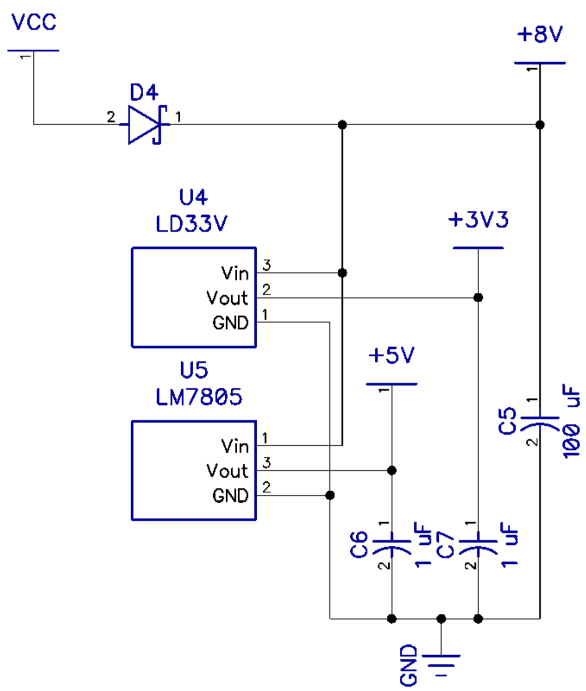


Figure 31: Power Regulation

The nominal 7.6V from the 2S battery configuration is regulated by the LD33V and LM7805 linear regulators. These provide stable, low noise outputs for the microcontroller and audio circuits. The system is protected by an input diode and line capacitance which helps prevent any excessive dip in the input to this circuit.

5.3. Second subsystem – Processing and Control

The processing and control system is dominated by the MCU, which is tasked with the audio processing, display output, Bluetooth communication, and motor control signaling.

5.3.1. MCU

The microcontroller is the core of the tuner system. The MCU is responsible for outputting and receiving data from the various subsystems. When applicable, received data is recognized as meant for control via the Button Input subsystem or Bluetooth Communication subsystem, output to another subsystem such as the Bluetooth Communication or Backlit Display subsystems, or input data from the Rechargeable Battery and Power Regulation, Bluetooth Communication, or Sound Feedback subsystems.

Final microcontroller to be decided, tentatively the TEENSY 3.2. The Teensy will communicate with the Bluetooth module, analog to digital converter, battery fuel gauge, LCD, and H-bridge. The schematics for these connections are broken apart as seen below. The following figure models the MCU connection to the battery fuel gauge and H-bridge/motor.

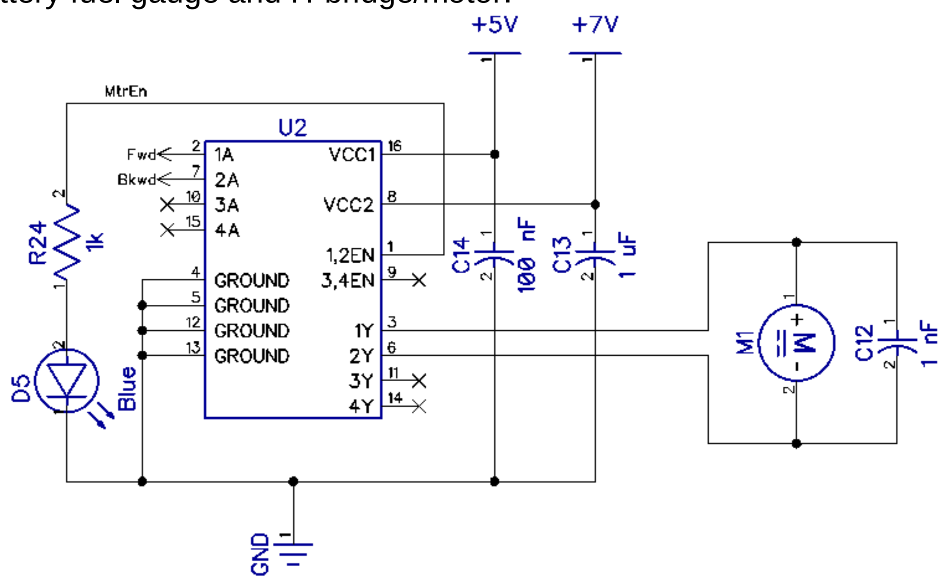


Figure 32: H-Bridge and Motor

The MCU uses three pins to communicate with the H-bridge, and three pins to communicate with the battery fuel gauge. P2.3 and P2.5 are used to control the direction of motor's spinning, while P2.4 is used to enable/disable the motor. The H-bridge also receives power from two different sources. Vcc1 receives 5V for logic operations, while Vcc2 receives 7V for power to the motor coils.

P1.6 and P1.7 are SCL and SDA I/O for I2C communication, respectively. These two pins are connected to the MAX1704G SDA and SCL pins, which utilize a pull-up resistor on each line. The MAX1704G receives power from the battery input, VDD, and also compares this voltage against its own internal reference table. It outputs the voltage as a 12-bit floating point number and also outputs an estimated battery percentage, so the MCU can gauge when to disable further operation.

The below circuit shows a modified schematic in the case that the Parallax servo is used in the final design. The servo is controlled via PWM and takes in the standard 7V input. This circuit is more simple compared to the H-bridge driver setup and is preferred, but the Parallax servo might not supply adequate torque for all stringed instruments to be tuned. This system still includes a blue LED to highlight when the servo is in use, though it needs to be manually controlled by the MCU since an enable pin is not necessary.

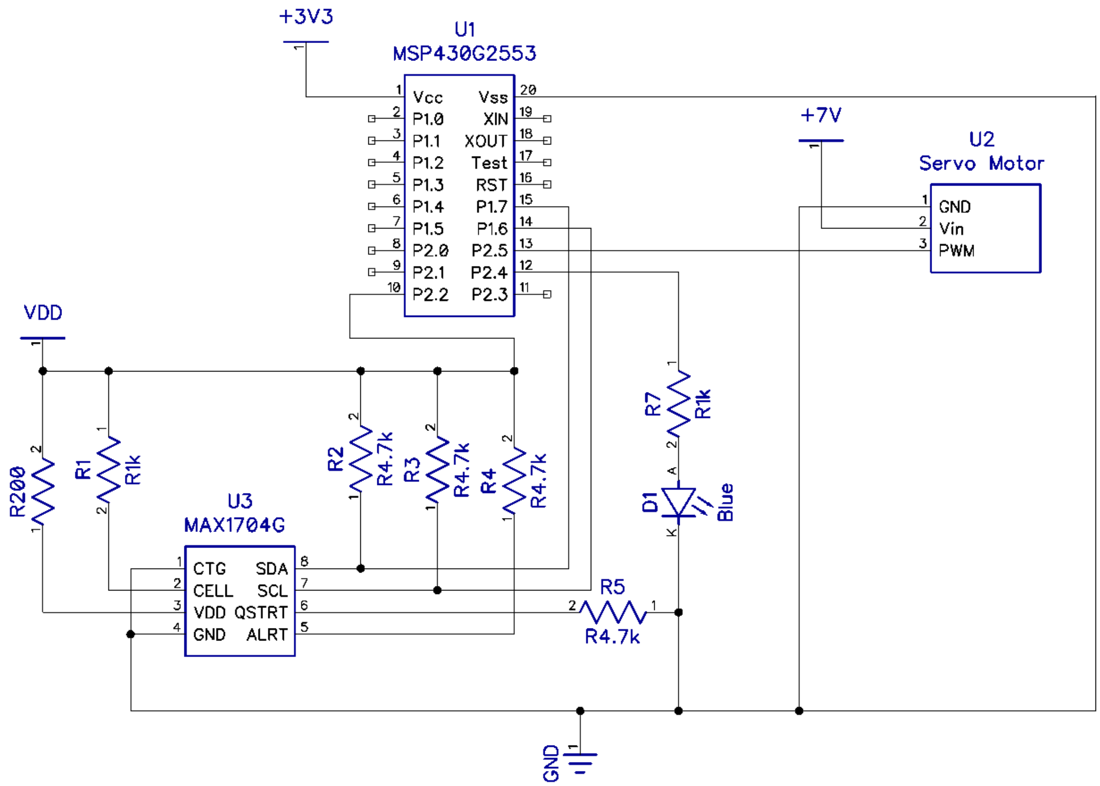


Figure 33: MCU, Servo, Battery Fuel Gauge Schematic

The MCU communicates with the Bluetooth module using UART which covers pins P1.1 and P1.2 respectively. This module is capable of two way communication, so the MCU can receive information from the phone application, as well as send information to the phone application. This part of the project uses a breakout board due to the high frequency requirements of Bluetooth communication. The possibility of creating an embedded version of this device within the main PCB is still under investigation pending Senior Design II. The following figure shows the configured circuit for Bluetooth module testing with the Adafruit BLE UART Friend.

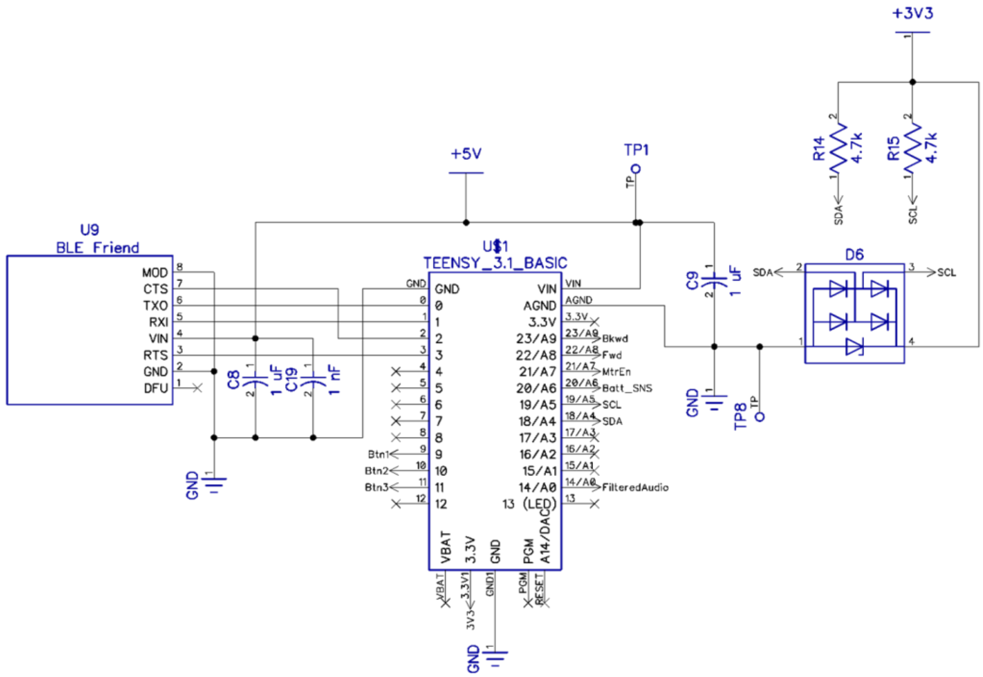


Figure 34: MCU and Bluetooth Module

This module has its own internal 3V Low Dropout (LDO) regulator which can be supplied by the same power rail as the MCU. The mode (MOD) pin is shorted to ground to allow for UART/Data mode which is the proper configuration for the needs of the project. The clear to send (CTS) pin is directly controlled by P1.0 which allows for hardware flow control. The read to send (RTS) pin is connected directly to P1.3 which is read to check availability of device for communication from MCU to Bluetooth Module. This enables the MCU to send data out as well as receive data according to the firmware on the device. The transmit (TXO) connects directly to P1.1 which is the RXI pin on the MSP430 device, while the receive (RXI) connects directly to P1.2, which is the TXI pin on the MSP430 device.

5.4. Third Subsystem – Audio Signal Processing

The audio signal processing is absolutely vital for this project. One of the main purposes of this device is to accurately determine the frequency of a note being played. In order to do this a microphone will be used to turn the sound waves coming from the instrument into an electrical signal that we can process and refine into a clear signal for the microcontroller to determine the frequency. This subsystem includes the microphone, an amplifier, a filter, and an analog to digital converter. In the figure below, the whole audio signal processing subsystem can be seen.

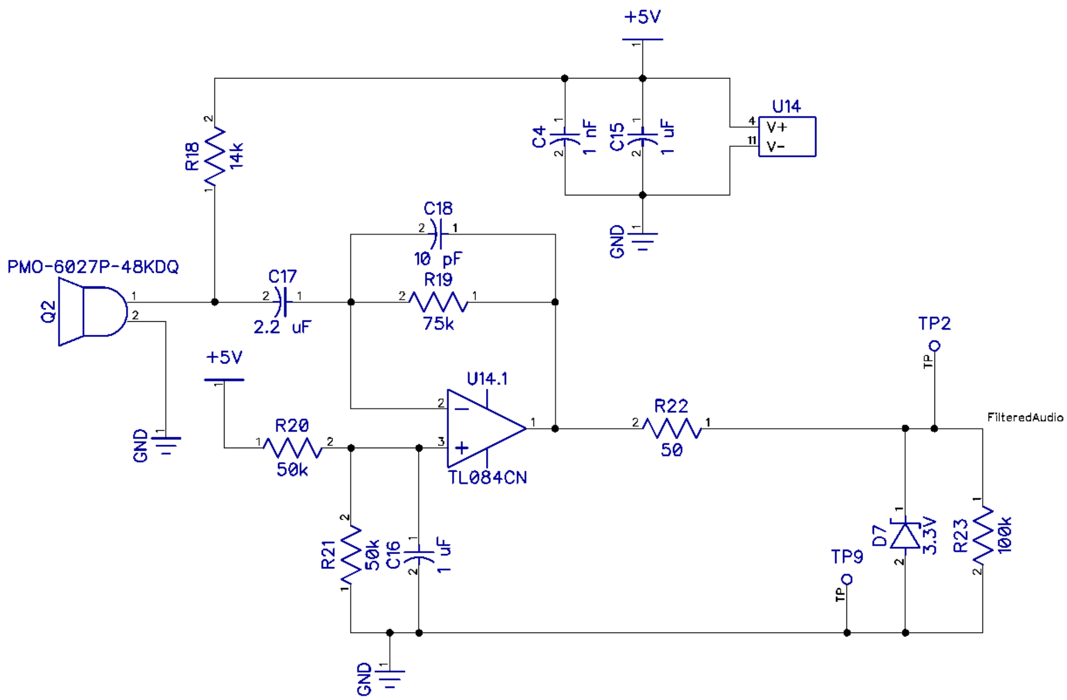


Figure 35: Audio Amplifier/Filter/ADC to MSP430

5.4.1. Microphone

After selecting the microphone that will be used, it was necessary to test it to ensure that it would be able to provide a signal that would be able to be used throughout the rest of the audio signal processing subsystem. First the microphone was provided a 2.5V input voltage and used an oscilloscope to measure across the voltage across the microphone. This turned out to be inconclusive due to an excess amount of noise. The microphone configuration was then changed to include a 1kΩ resistor in series with the microphone. This was done to see how that would impact the signal. The oscilloscope was moved to measure the voltage across the resistor instead of across the microphone. This provided a better picture of how the microphone was performing however it still was extremely noisy. We wanted to try one last thing before going back to the drawing board. We added a 100nF capacitor in parallel with the resistor. This addition allows for some of the noise to be filtered out and we were able to see clearly the microphone's output waveform. The final testing configuration can be seen in the figure below.

With this configuration we used a function generator attached to a speaker to test different frequencies to see how the microphone responded. The output voltage waveform, while clear, was roughly 2 mV peak to peak. This would be an issue later in the system. The analog to digital converter only has a resolution of a little more than 1mV. Due to this constraint an amplifier was necessary to be designed. In addition to the amplifier, in order to ensure only the desired frequencies were being converted into the digital signal an active low pass filter would need to be designed as well.

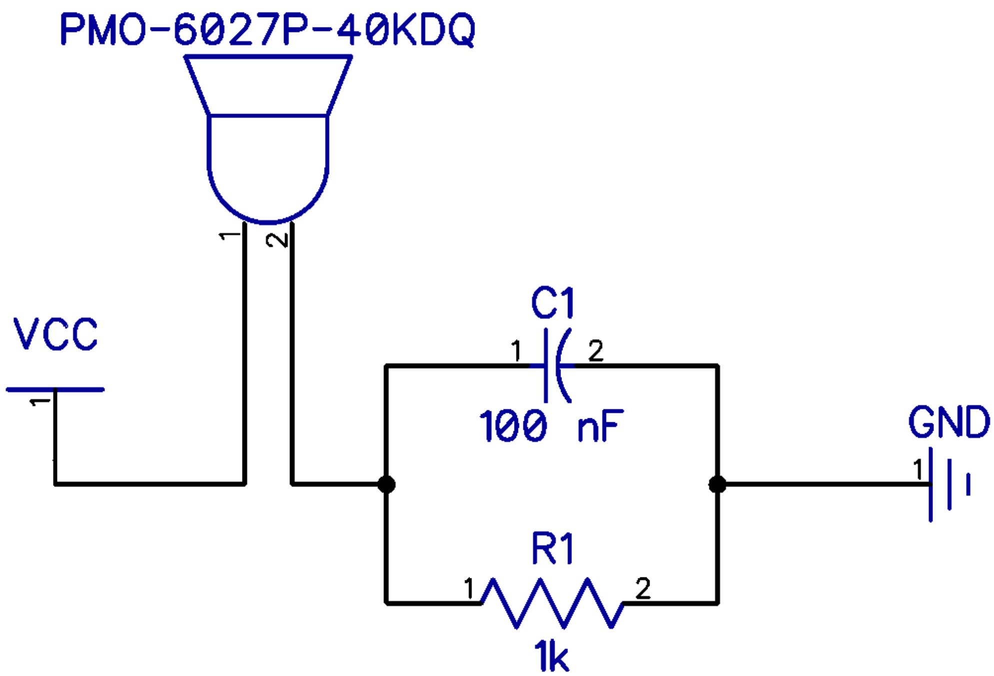


Figure 36: Microphone with Low Pass Filter

5.4.2. Amplifier

The microphone signal should first be amplified before being filtered. This is due to the fact that the signal from the microphone already has a very low voltage. While filtering, the signal will be attenuated and could possible lower the signal to noise ratio. Lowering the signal to noise ratio could negatively impact the ability to successfully determine the signal from the instruments.

The output voltage from the microphone is roughly 2mV peak to peak. This is not nearly high enough for the analog to digital converter to reliably convert the signal. Therefore the signal needs to be amplified. The signal coming from the microphone has a DC offset of about 2.5V. This voltage was used to activate the microphone; however this voltage should not be amplified. The reason for this is that the DC offset will also be amplified and cause the signal to clip due to the operational amplifiers that are used. In order to remove the offset, a capacitor is necessary.

In addition to the capacitor a standard inverting amplifier was designed. The amplifier was designed to have a gain of roughly 100 due to the extremely low output voltage from the microphone. To be able to have this gain, the amplifier uses a 1kΩ and 100kΩ resistor. The operational amplifier is also provided a ½ Vcc DC source. This allows for the output voltage to have a DC offset ensure that the signal will not be clipped if the voltage goes below zero. The resistor and capacitor actually end up acting like a high pass filter. Due to this behavior the capacitor was sized, at 5µF, so that it would actually assist in the process of reducing noise. This high pass filter was designed to have a ωo equal to 30 Hz. This satisfies the design requirement to capture the lowest frequency that the most common string instruments can produce. The amplifier will have an output voltage with a DC offset and an AC signal that is roughly 100 times the input voltage.

5.4.3. Filter

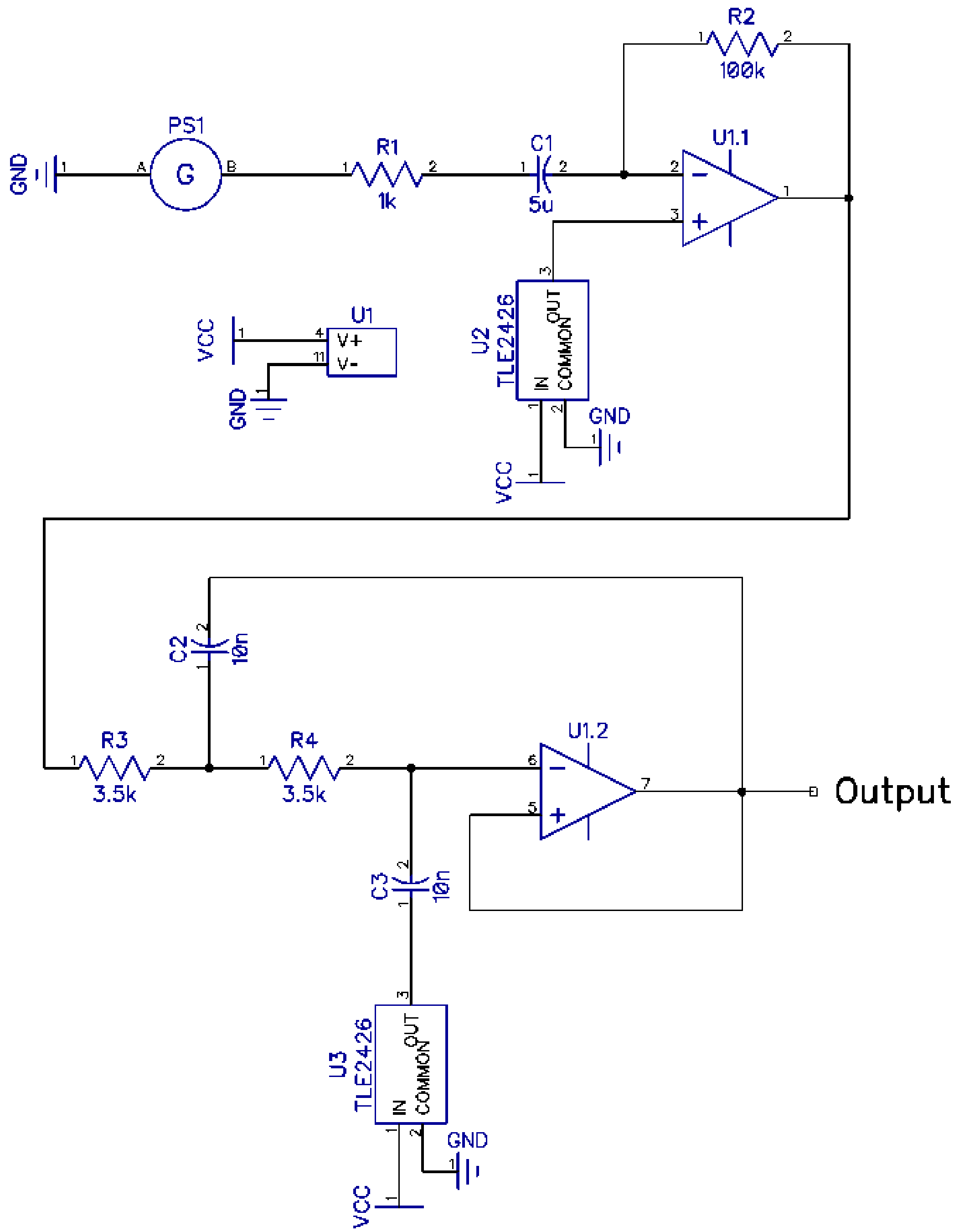


Figure 37: Active Filter Design

After the amplifier, the signal needs to be filtered. Filtering the signal will help reduce noise and ensure that the microprocessor is able to accurately tell what frequency the microphone is picking up from the instrument. There were plenty of difficulties when designing this filter. The input resistance of the operational amplifiers starts to affect the stage before. This caused some issues with attenuation. A second order, low pass, active filter was chosen to be used instead of a bandpass filter. This is due to the fact that the amplifier is already filtering out the low frequencies and the high attenuation factor will greatly reduce high frequency noise.

This filter makes use of two 3.5kΩ resistors and two 10nF capacitors. Along with the resistors and capacitors, the operational amplifier was provided a DC offset to ensure there would be no clipping when the amplifier output voltage would go lower than zero. This configuration results in a low pass filter with ωo = 3kHz with an attenuation factor of roughly 40dB/Decade The output voltage from the filter will have a DC offset with an AC signal. This AC signal will be roughly 100 times the input voltage for frequencies between 30Hz – 3kHz. The combined schematic for the amplifier and filter can be seen in the figure below. In addition to the schematic, the frequency response of that schematic can also be seen below.

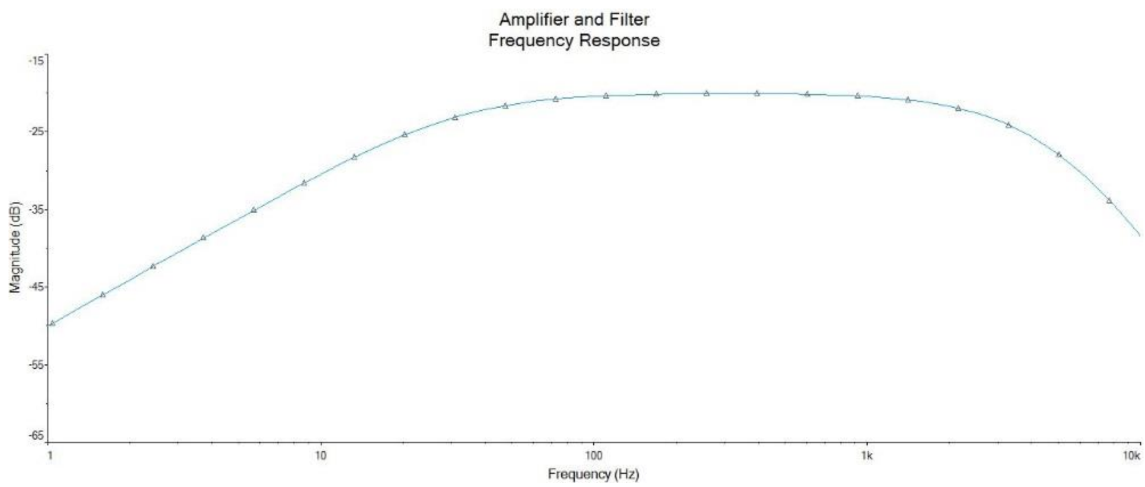


Figure 38: Filter Frequency Response

5.4.4. Analog to Digital Conversion Testing

Once the signal is amplified and filtered, we have our desired waveform. However this waveform is an analog waveform. In order to successfully determine the frequency of this waveform, it needs to be converted to an digital signal. This digital signal will then be sent to the microprocessor which will perform an fast fourier transform to determine the frequencies which most of the signal lies. From the fast fourier transformation, the frequency of the note played will be determined.

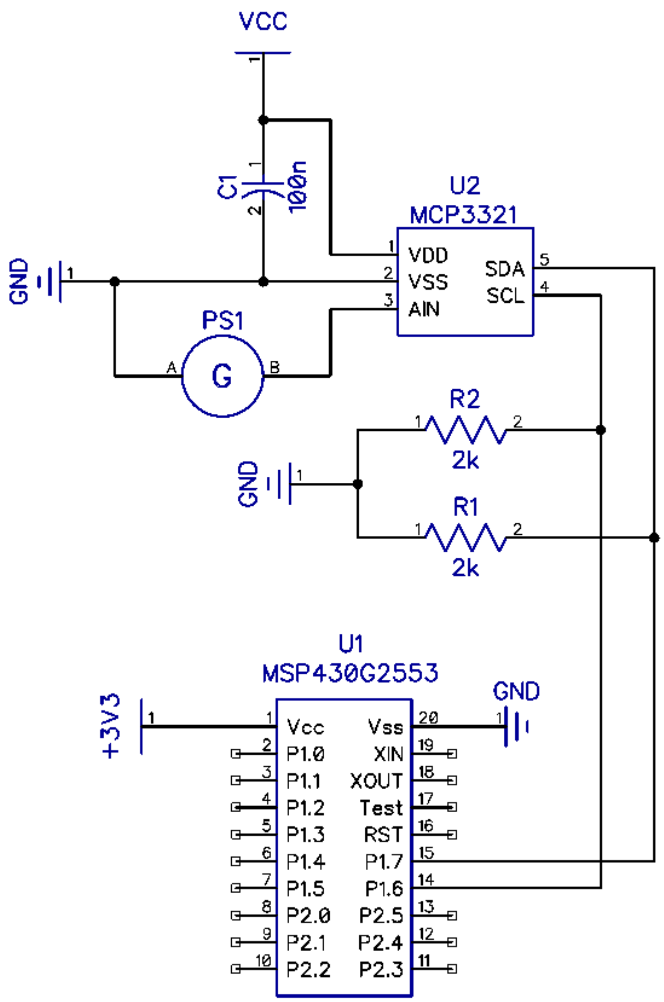


Figure 39: ADC to MCU Schematic

In order to test this component, we need to hook it up to the microprocessor and see if they can communicate using I2C serial communication. I2C serial communication was chosen due to both the analog to digital converter and the microprocessor supporting this type of communication. A function generator was also used to ensure we had a set frequency to compare the results from this test to. A decoupling capacitor was also added to assist in filtering out high frequency noise. Also there was two additional pull up resistors between the analog to digital converter and the microprocessor, one for the clock signal and the other for the I2C data bus. The overall schematic can be seen below.

The microprocessor was then programmed to try and communicate with the analog to digital converter. Once it was able to communicate, it was reprogrammed to read the results from the analog to digital converter. We were able to receive this data, however the program to perform the fast fourier transform has not been written. This means we were unable to verify, with the microprocessor, that the analog to digital converter had an output waveform had the same frequency as the input waveform. To mitigate this issue, we used an oscilloscope with fast fourier transform capability. We also used a phone application to see what the fast fourier transformation of the guitar should look like. It included additional peaks where the was harmonic frequencies. We were able to verify that the analog to digital converter had a fast enough sample rate for us to use.

5.5. Fourth Subsystem – User Interface

One of the most important aspects of a handheld device is how the user physically interfaces with it. This covers the form, fit, and function of the physical aspects of design. The device should be capable of being wielded by all hand sizes, should provide ergonomically placed buttons to provide easy standalone menu operation, and should provide clear feedback in the form of display and notification lights.

5.5.1. OLED Display

The OLED display provides the user with with concise and informative data regardless of whether the device is operating in manual or automatic modes. The display provides the current state the device is in (Idle, Ready, Tuning, etc.). In addition, the currently received frequency is displayed, along with the target frequency. The time taken to complete a full string tuning is displayed once automatic tuning completes. If operating in manual mode, the current direction, clockwise or counterclockwise, the motor is turning is shown.

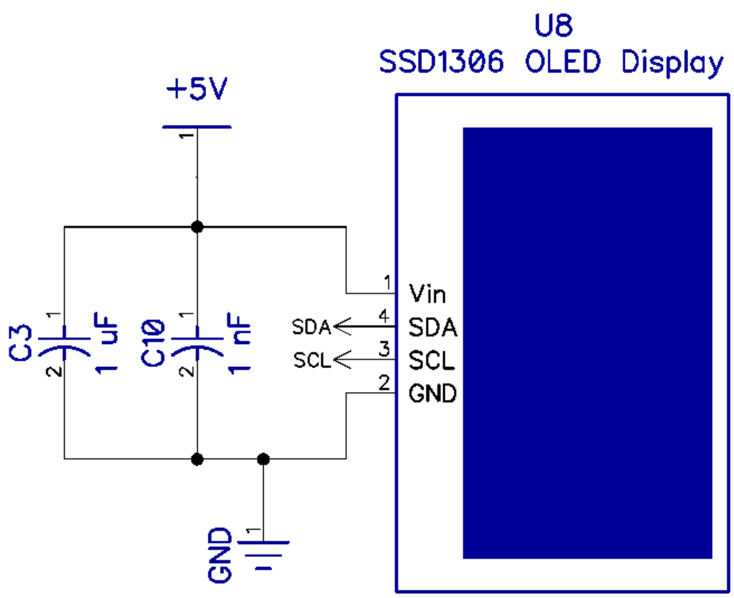


Figure 40: MCU to OLED Display

The above figure shows how the OLED display connects to the MCU via I2C. The display is powered from the 5V bus, while the MCU is powered from the 3.3V bus. The two components share a common ground which is needed to allow for low noise communication over the I2C bus. The display is tested by running an example Arduino script to send text and figure to the display.

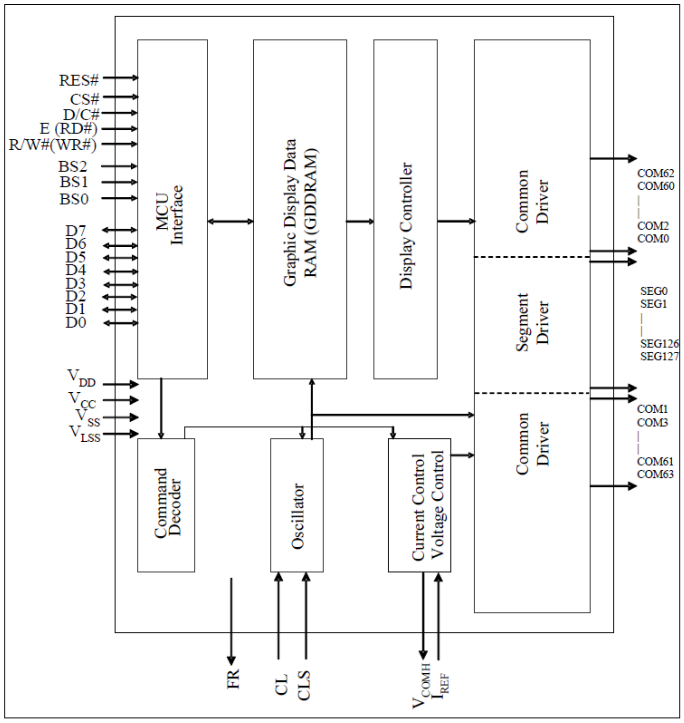


Figure 41: SSD1306 OLED Display Driver Block Diagram

5.5.2. Indicator Lights

The device has LED indicator lights built into the various circuits to give feedback for different functions. These lights can be important for discerning the current status of various subsystems of the device. The charging circuit lights up a green LED when the device is charging and lights up a red LED when the device is finished charging. The motor also has an blue indicator LED which lights up when the H-bridge is enabled. The following table shows the current projected LED signaling.

Table 31: Indicator Light States

State	Green	Red	Blue
On	Charging Good	Charging Bad	Motor Circuit Active
Off	Not Charging	Charging Not Bad	Motor Off

5.5.3. Functionality Buttons

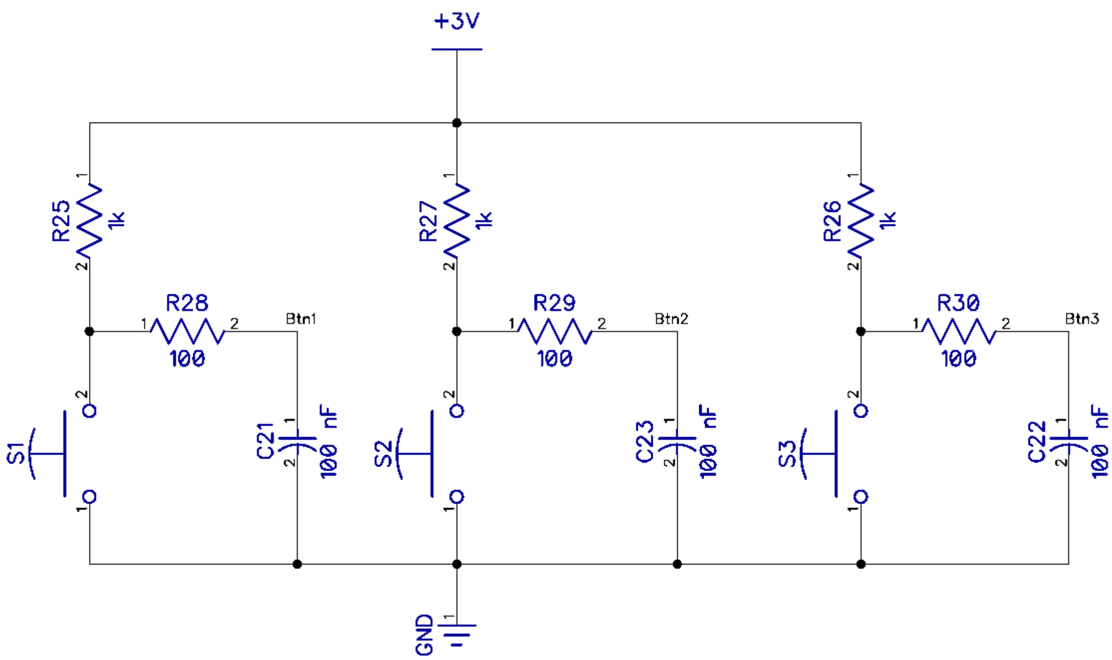


Figure 42: Button De-bounce Circuit

The physical buttons on the device include the power switch, the mode button, and the two tuning triggers. The power switch allows the system to be shut off completely to conserve battery life, and turned back on for regular operation. The mode button is pressed when the user is ready to begin the tuning process and wishes to specify the mode, manual or automatic. By default, the mode is manual. When the mode button is pressed the device toggles into automatic mode. Finally, the two tuning triggers are pulled to activate the servo motor when in manual operating mode. The upper trigger signals the motor to turn clockwise, and the lower trigger signals the motor to turn counterclockwise. All switches are physically debounced by using a resistor-capcatior network, along with the built in Schmitt Trigger internal to the MSP430 line. This RC network is seen in the figure below, showing a Normally Open pushbutton configuration being sampled by a pin on the MSP430.

5.5.4. Physical Design

The physical design of the device will be modeled in Solidworks. Solidworks is a 3D Computer-Aided-Design program which can be used to make parametric shapes and models. The physical form factor of this device needs to fit all of the electrical components as well as the mechanical switches and motors. It should be comfortable to wield one-handed by the majority of musicians and should have switches placed in easy to reach locations. The physical design of the body will be done after full electronic assemblies are created, in the Fall 2017 semester. The body will then be rapidly prototyped by utilizing 3D model exports of the PCBs and custom-modeled parts. The body will then be 3D printed using ABS or Nylon, which greatly speeds up the turnaround time from modeling to testing. After the models are printed, the fresh plastic parts are cleaned and prepared for assembly. The entire unit is assembled and any flaws are noted for change in the next revision. Using a 3D printer enables these new revisions to be rapidly tested until a final design is nailed down. If this product were to be brought to market, a different strategy would need to be used, such as injection molding, in order to increase yield and decrease machine cost.

Figure 43: Physical Mockup

The figure above shows the output of the slicing program, Slic3r, when the body model is processed. This program turns the Solidworks model into G-code which can be directly sent to a 3D printer for rapid assembly. The model is cut into individual layers which are 0.2mm thick and are stacked on top of each other to gain the desired physical model. The yellow layers are known as perimeters because they outline the edges of the model. The pink lines are known as infill, which in this case, is set to rectilinear for best structural integrity. The gap for the motor shaft and OLED display can easily be “bridged” by the printer and do not need support in this case.

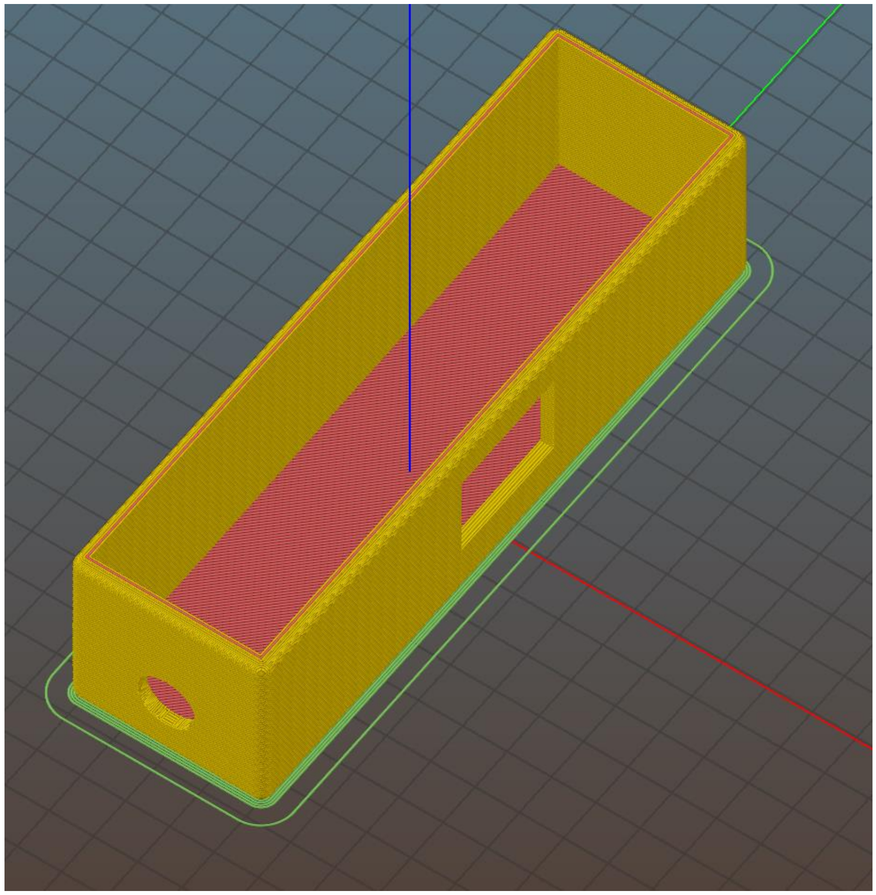


Figure 44: Slic3r Output

5.6. Overall Configuration

The overall configuration has held consistent to the original brainstormed idea. Each of these circuits was tested and confirmed to work as designed separately. The systems will be combined into one fully functioning system in Senior design II in the Fall 2017 semester.

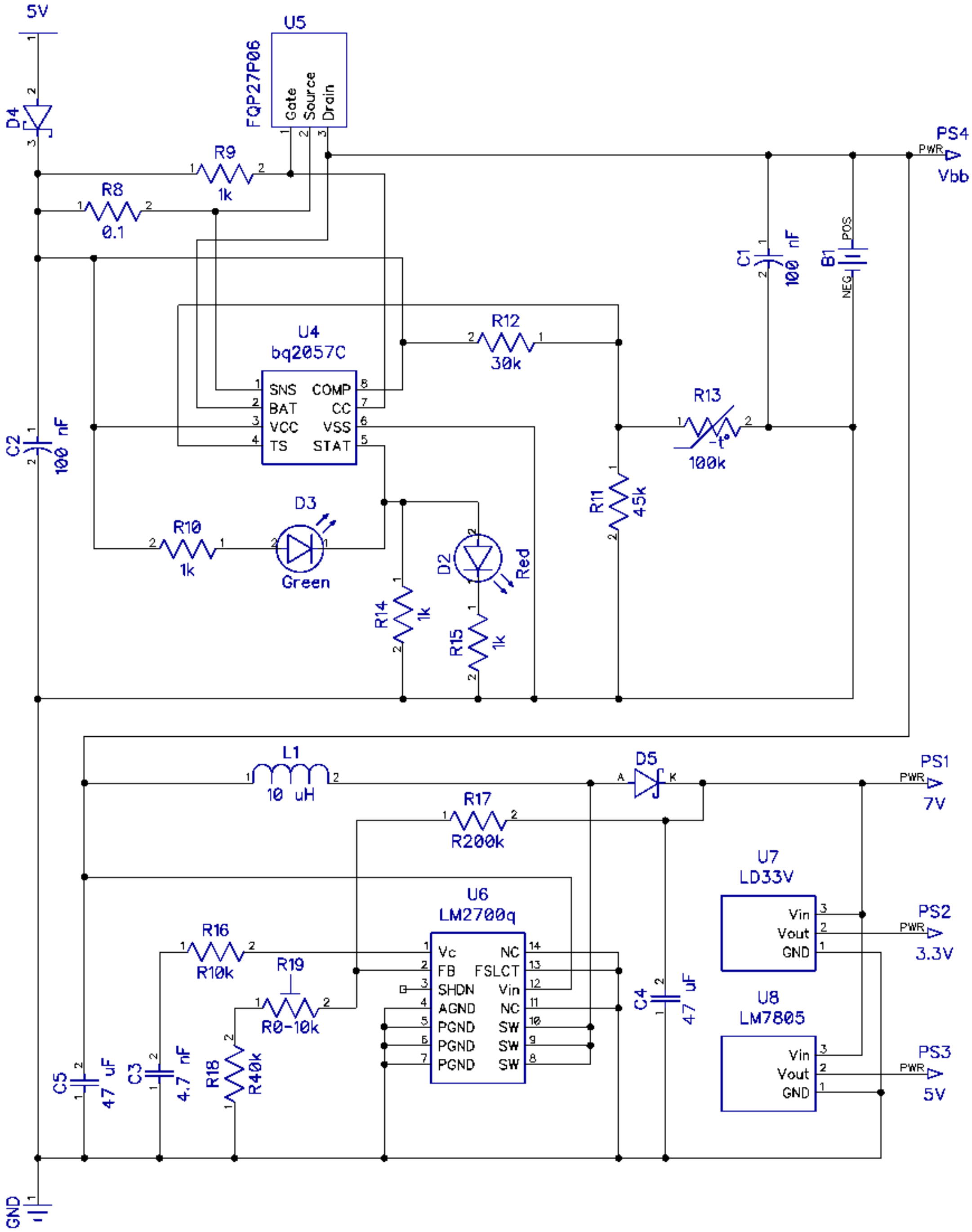


Figure 45: Automatic Guitar Tuner System Part One

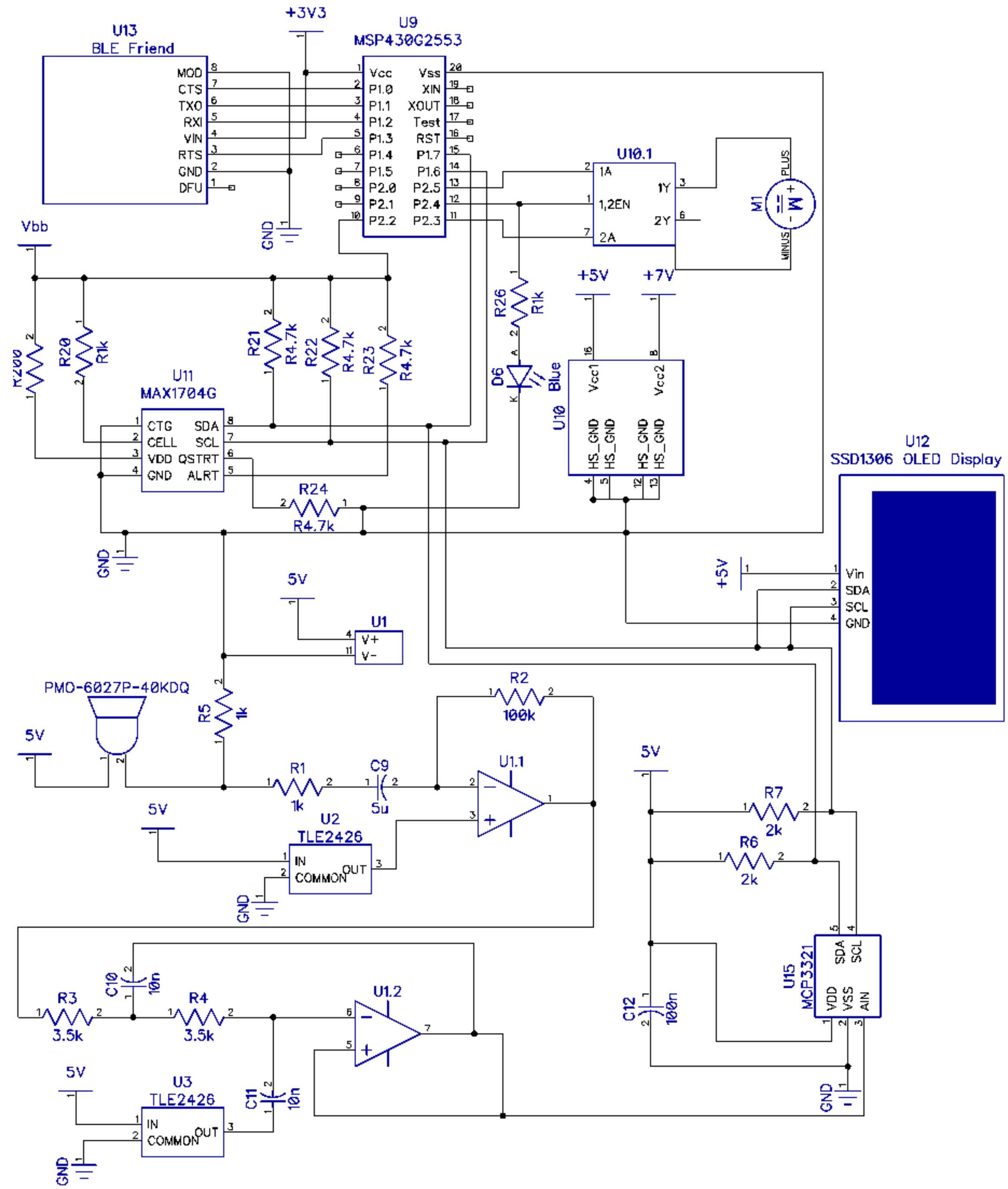


Figure 46: Automatic Guitar Tuner System Part Two

The PCB for this project was designed entirely using Diptrace. This program is an electrical CAD program which allows engineers to design everything from individual components, to schematics, to full PCB layouts. The entire design was brainstormed and then the schematic created in Diptrace.

All schematics were breadboarded and tested separately, then the full schematic exported to the layout editor. The physical board was designed to separate the high power components from low powered components as much as possible. The charging port, battery input, and motor controller were all placed near the top, the microcontroller placed near the middle, and the audio preamplifier placed at the bottom. The large separation between power and audio should provide for the cleanest signal for frequency extraction.

The digital PCB layout was exported to gerber and drill files and a physical copy was ordered from OSHPark. The boards are manually populated, using mainly surface mount passive components and through hole active components. This combination allows an easily tested board while still retaining a small physical size necessary for a handheld device.

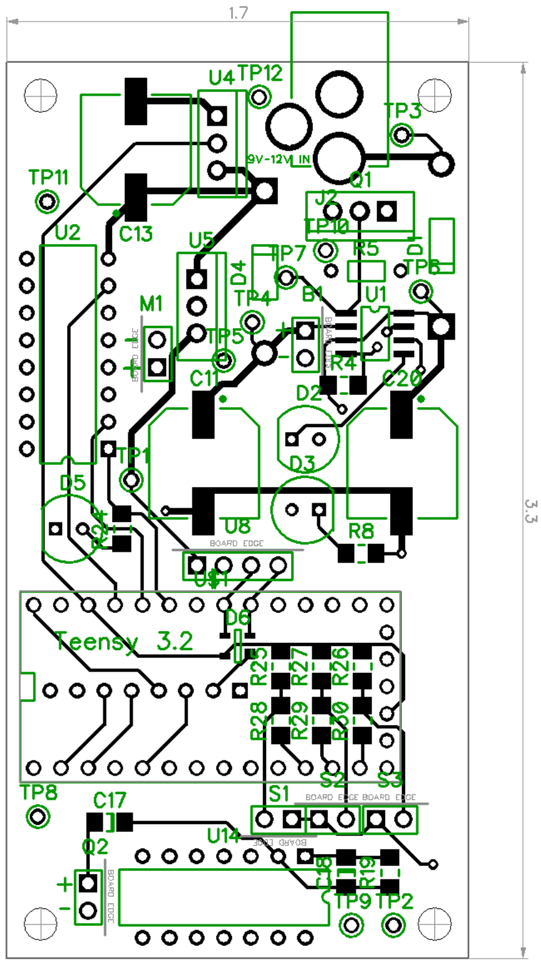


Figure 47: PCB V2

5.7. Summary of Hardware design

The device will use the microphone to pick up on sound waves produced by the stringed instruments. This signal will be put into a filter to eliminate unwanted frequencies. After the filter the signal will go to an amplifier. This is because the small signal produced by the microphone will be too small to accurately and reliably be picked up by the analog to digital converter. Once it has been amplified, the signal is supplied a DC offset to ensure that the entire wave is between 0 - 5 V. This is because the analog to digital converter does not convert negative values.

After the offset is added, the signal is then sent to the analog to digital converter, which converts the signal at 22.3k samples per second into 12 bit serial values. These serial values are sent to the microcontroller, where a fast fourier transform function will determine the frequency of the sound wave. The microcontroller will then compare this value to a lookup table for the desired frequency and determine the error. This error will then be used to send a signal to the servo system. The servo system will then rotate and either tighten or loosen the turning knobs. This will either raise or lower the frequency. The microphone will once again capture the sound waves and the cycle continues.

Note	Hz	Note	Hz	Note	Hz	Note	Hz	Note	Hz	Note	Hz	Note	Hz
C1	32.7	C2	65.4	C3	130.8	C4	261.6	C5	523.3	C6	1046.5	C7	2093.0
C#1	34.6	C#2	69.3	C#3	138.6	C#4	277.2	C#5	554.4	C#6	1108.7	C#7	2217.5
D1	36.7	D2	73.4	D3	146.8	D4	293.7	D5	587.3	D6	1174.7	D7	2349.3
D#1	38.9	D#2	77.8	D#3	155.6	D#4	311.1	D#5	622.3	D#6	1244.5	D#7	2489.0
E1	41.2	E2	82.4	E3	164.8	E4	329.6	E5	659.3	E6	1318.5	E7	2637.0
F1	43.7	F2	87.3	F3	174.6	F4	349.2	F5	698.5	F6	1396.9	F7	2793.8
F#1	46.2	F#2	92.5	F#3	185.0	F#4	370.0	F#5	740.0	F#6	1480.0	F#7	2960.0
G1	49.0	G2	98.0	G3	196.0	G4	392.0	G5	784.0	G6	1568.0	G7	3136.0
G#1	51.9	G#2	103.8	G#3	207.7	G#4	415.3	G#5	830.6	G#6	1661.2	G#7	3322.4
A1	55.0	A2	110.0	A3	220.0	A4	440.0	A5	880.0	A6	1760.0	A7	3520.0
A#1	58.3	A#2	116.5	A#3	233.1	A#4	466.2	A#5	932.3	A#6	1864.7	A#7	3729.3
B1	61.7	B2	123.5	B3	246.9	B4	493.9	B5	987.8	B6	1975.5	B7	3951.1

Figure 48: Frequency Vs. Musical Notes

The figure below shows the main breadboard testing performed for this project. This displays the individual test circuits for the analog filter/amplifier circuit, H-bridge circuit, battery charging circuit, fuel gauge circuit, and boost converter circuit. These circuits were individually tested and analyzed using a combination of digital multimeter and oscilloscope. The two multimeters were a Uni-T 61D and Uni-T UT210E. The oscilloscope was a Tektronix TDS 744A. These tools were acquired by a team member previously for personal use and were important for testing circuits in a timely manner.

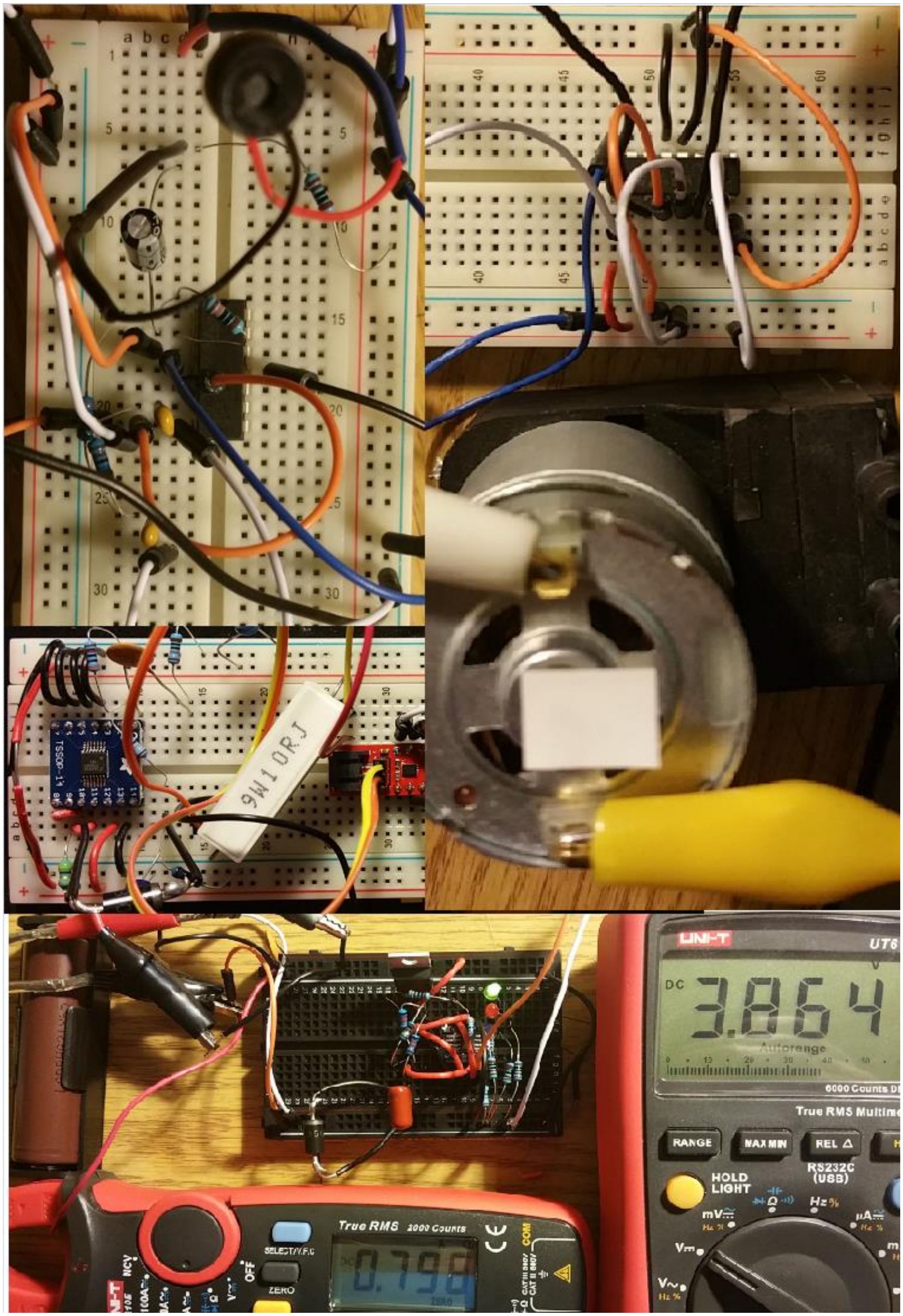


Figure 49: Electrical Testing Circuits

6. Project Software Design Details

All code running on the MCU is written in the C programming language. MCU code utilized Flash memory for all display related functionality. This is to maximize the availability of RAM for more time dependant computations. All MCU code related to frequency signal, motor, and button input utilizes RAM to maximize response time and processing throughput.

The Android application is developed primarily in the Java programming language. This is a design constraint enforced by all Android operating systems. The application is developed using Android's official IDE, Android Studio. The Extensible Markup Language (XML) is also employed.

6.1. Software Functionality

Software is split up between the physical device and the phone application. The physical device contains an TEENSY 3.2 which handles all processing and communication, while the application adds additional features for the user and communicates with the device over Bluetooth.

6.1.1. Android Application

While not completed during the Senior Design development time of the project, the Tuner Application will be available to download for free via the Google Play Store for all Android devices running Lollipop OS or any more recent Android OS release. The app will communicate via Bluetooth tether with the physical tuning device. Intended use of the app is as both a wireless control and additional information display for the tuning device. The app will consist of five screens, the Main Menu, Tuning, Tuning Library, and Options screens.

The Main Menu screen displays the currently selected tuning and the status of the Bluetooth connection between the application and the device. This screen is displayed by default when the application is first opened and may be returned to via a button on each of the four other screens. In addition, the main menu provides navigation points to the other four application screens. Finally, an exit button immediately shuts down the application and break the Bluetooth tether.
The Tuning screen is where all active communication between the application and the physical tuning device takes place. The active tuning is displayed to the user, this includes all strings involved in the tuning along with the specific frequency each string is to be tuned to. The user may then select an individual string, and lock it in for tuning. After locked in, the string's tuning frequency is sent from the application to the physical tuner.

Once the frequency is received by the tuner, the device waits for the user to activate tuning via physical button input. On activation, the device sends a signal to the application which then prompts the user to strum the string. The user may be prompted to strum multiple times until the intended tuning frequency is achieved. On tuning completion, statistics from the tuning session are displayed to the user including starting frequency and the time it took to tune the string. The user may now choose a new string to tune and the tuning process begins again. The Tuning Library screen consists of a local database of standard and user defined tunings. From this screen, the user may select a pre-existing tuning to become the active tuning, edit a pre-existing tuning, delete a preexisting tuning, and/or create new custom tunings.

Creation of a custom tuning begins by selecting the “New Custom Tuning” button. On selection, a UI form is displayed consisting of a user input box for the name of the new tuning, a button to add a new string to the tuning, and a “Save” button. When a new string is added, the existing form is expanded to contain a user input box for the string’s name and that string’s frequency. Once the user has completed the form in total, the “Save” button may be pressed to save the custom tuning to the local database and the user is returned to the Tuning Library screen. Editing or activating a pre-existing tuning begins with selecting a tuning by tapping on the title of a displayed tuning. Once selected, the tuning’s string titles and frequencies are displayed, along with an “Edit” button, a “Delete” button, and a “Set Active” button. The “Set Active” button will set that specific tuning as the active tuning which is used during the tuning process. The “Delete” button prompts the user with a dialogue asking if the user is sure he/she wishes to delete the selected tuning. The user may choose to accept the deletion, which will permanently delete the tuning from the database, or to cancel the deletion in which case the tuning is preserved. The “Edit” button works similarly to the creation of a custom tuning, the same UI form is displayed as in the custom tuning process with the difference that some input boxes will already be filled out with information that the tuning already possessed.

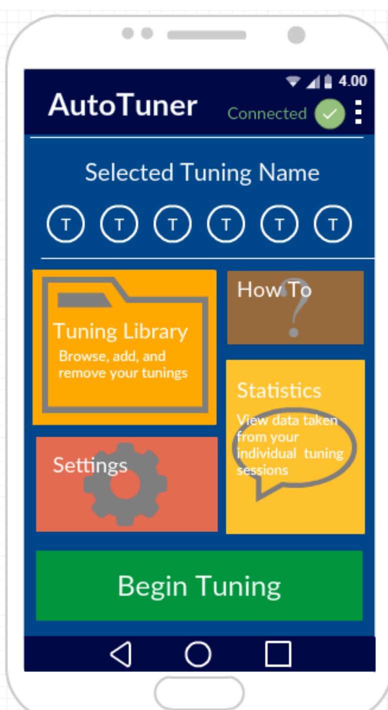


Figure 50: Android Application Main Menu Mockup

6.1.2. Microcontroller Software

By default, the MCU operates in Low-power mode 0 to conserve power. Once a signal is received via physical button input, the MCU will enter Active Mode. The software running on the MSP430 microcontroller utilizes a Finite State Machine algorithm, further described in the following section, to monitor and control the various subsystems that make up the device. During the tuning process, a transfer function is utilized to control the servo motor based on voltage level input from the ADC.

6.1.3. OLED

Utilizing the device manufacturer's Adafruit SSD1306 and Adafruit GFX libraries, the following library-defined functions are utilized in the project.

Table 32: OLED Functions

Function	Description
Adafruit_GFX(int16_t w, int16_t h)	Initializes the device software, arguments are the width and height of the screen
dim(boolean dim)	If true, dims the display by a default value (0x9F or 0xCF based on input voltage)
startWrite()	Initializes the OLED for data writing
write(uint8_t c)	Write a char (argument c) to the display
setTextSize(uint8_t s)	Set the size of the font
setTextColor(uint16_t c)	Set the color of the font
setTextWrap(boolean w)	Wraps the text when character reaches limits of display
setFont(const GFXfont *f)	Sets the font type to one of the library provided fonts

Output of the OLED is as seen in the table below:

Table 33: OLED Device States

Device State	Output
Device On, no button input	No output, display is dimmed to conserve power
Automatic mode button pressed	“Select Tuning”
Once a tuning is received via Bluetooth or the tuning has just been adjusted	“Listening…”
While the servo is tuning	“Automatic tuning in progress…”
Once tuning is complete	“Automatic tuning complete!”
While manually tuning	“Manual tuning in progress…”

6.1.4. Servo Motor

Texas Instruments’ servo control library, Energia, provides all functionality needed for this project. This functionality includes, reading from the servo (in degrees and microseconds), writing to the servo (in degrees and microseconds), and a simple timer.

6.2. Algorithm Description

The biggest software hurtle to overcome is the implementation of the frequency detection algorithms, Autocorrelation and Zero-Crossing. These both take the sampled 12-bit inputs from the ADC, offset the samples to properly center them along the zero axis of a sinusoidal signal, and convert them to a final detected frequency value. In order to compute an accurate frequency, the ADC samples at 49kHz until 1024 samples are retrieved. It is worth noting that each of these samples are actually an average of four samples taken during this time to help with signal accuracy.

Autocorrelation relies on the concept of cross-correlation, which involves measuring the similarities between two signals. The two signals used in this case are the signal made up of the samples from the ADC, and a time offset version of that same sample. The time offset in this case is discrete, it is not a true time value, but simply the sample offset by one measured value.

Each sample value is multiplied by the value following it, it is then added to a total sum value. This is done until the number of remaining samples has been exhausted for a single sum. This can be done up to 1024 times, one time for each sample. If a sum is detected to have crossed the zero axis of the signal twice, first up, then down, the periodic value is found. The sampling frequency is divided by this value, and the sample frequency is found.

The Zero-Crossing algorithm is a less processing-heavy form of Autocorrelation. Instead of computing a sum of all values a potential 1024 times, the ADC samples are processed as-is. The 1024 values are iterated over once, when a signal period is detected, the number of samples in that period is recorded.

A period is detected when the signal has crossed the zero axis a total of three times, once going positive to negative, then negative to positive, then positive to negative again. This is done multiple times, and the final number of samples per period is averaged together. Once an average is found, it is multiplied by the time it takes to retrieve a single sample from the ADC, and the sample frequency is found.

6.3. Coded Flow Chart

The coded flow chart seen in the figure below shows the brainstormed plan for menu navigation on the physical device. This menu will be viewed through the OLED display and navigated using buttons on the device. The menu will flow with a vertical scroll interface, using buttons for up and down movements, respectively. There is a center select button which will move further down the branch of menu selections. The flow chart starts with the initial opening of the interface by turning the device on. The main menu contains options for choosing Tuning Type, Adding Custom Tuning Type, Manual Tuning Mode, and Exiting. Each of these menus has further selections which can be made and offer a great range of functionality to the user.

The Choose Tuning Type branch allows the user to select the particular note which is desired for the string in question. This menu option can also store multiple tuning profiles, where all the strings can be tuned in sequence without needing to select the notes individually. Pressing the select button on the chosen Tuning Type will prompt the user to place the device on the guitar knob and strum the string. The device will automatically extract the carrier frequency in the sound of the string and tune the string to the desired note.

The Add Custom Tuning branch allows more flexibility in this device over the default profiles included in the memory. The user selects this menu and can create a new profile. The user selects the number of strings to be stored in the profile, then inputs the desired note for each string. The note can be input as a classical letter, or if the corresponding frequency is known, the numerical frequency. After the parameters are set to the user's preferences, the profile is saved and stored on the device memory for later use.

The Manual Tuning Mode branch can be used for a few functions which give greater control to the user. The user can input a desired note to be tuned to and press the select button. The device will then listen to the sound output of the string and display the corresponding note/frequency of the string. The user can then manually turn the knob and observe the frequency as it changes. In a way, this mode allows the user to train their 'ear' and learn the difference in notes and how to manually tune the guitar on their own. This menu also offers a Hold to String or De-string option which will spin continuously while the user holds the up and down buttons respectively. This function saves the user from having to repetitively turn the knob when changing strings on the instrument and prevents over tightening when restringing.

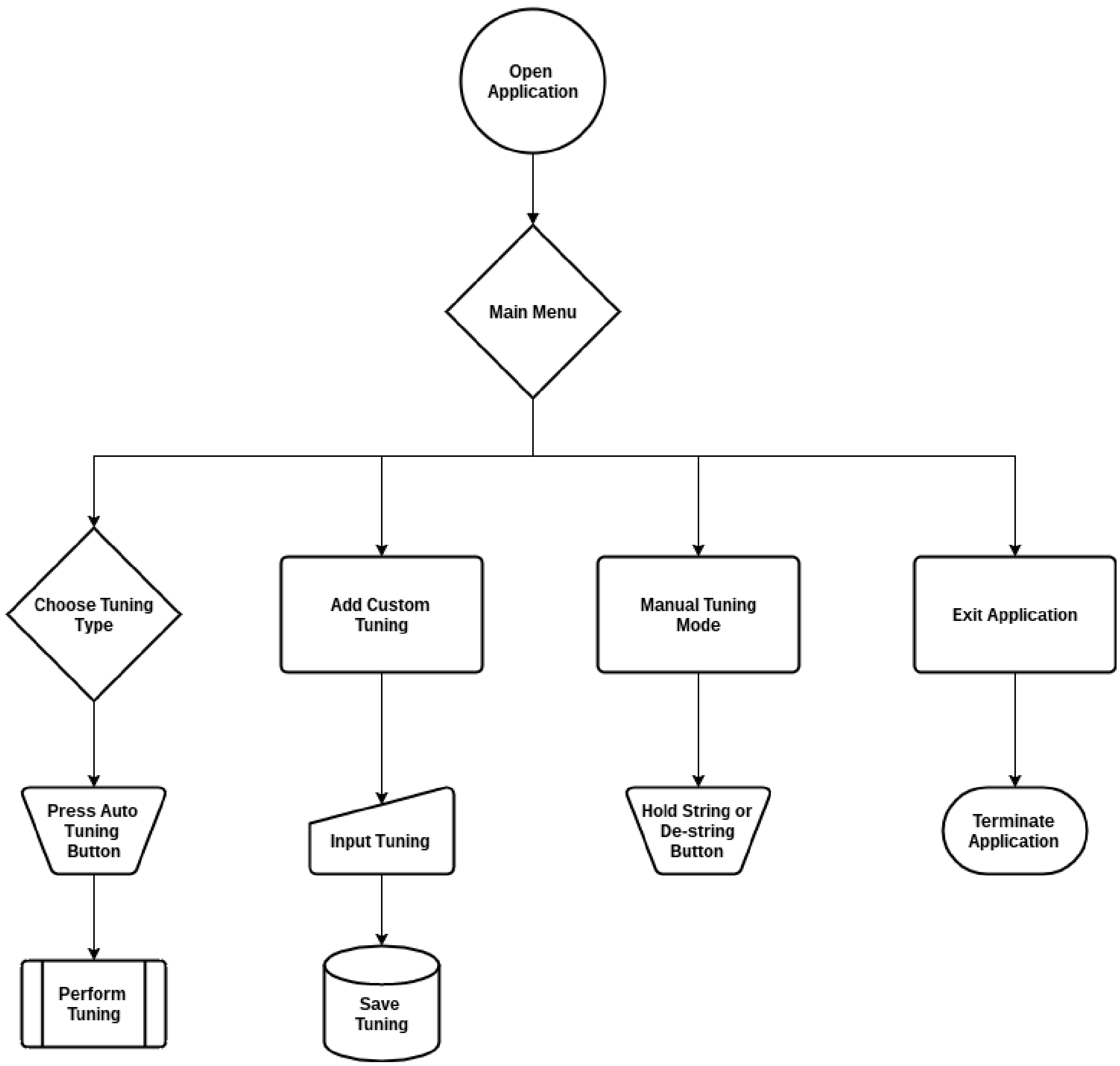


Figure 51: Coded Flow Chart

6.4. Control Signal Testing

Control signal testing is performed on parts which need to intelligently communicate with the MCU to play a part in the overall system. In order to accelerate this process and for the point of proof-of-concept, most control signal testing was performed by utilizing a Teensy 2.0++ microcontroller.

This microcontroller offers a form factor which easily fits on a breadboard and can utilize the Arduino IDE which greatly simplifies the process of finding sample code to perform software communications. The Teensy 2.0++ interfaces with the test laptop through USB and can print information to the serial interface to feedback.

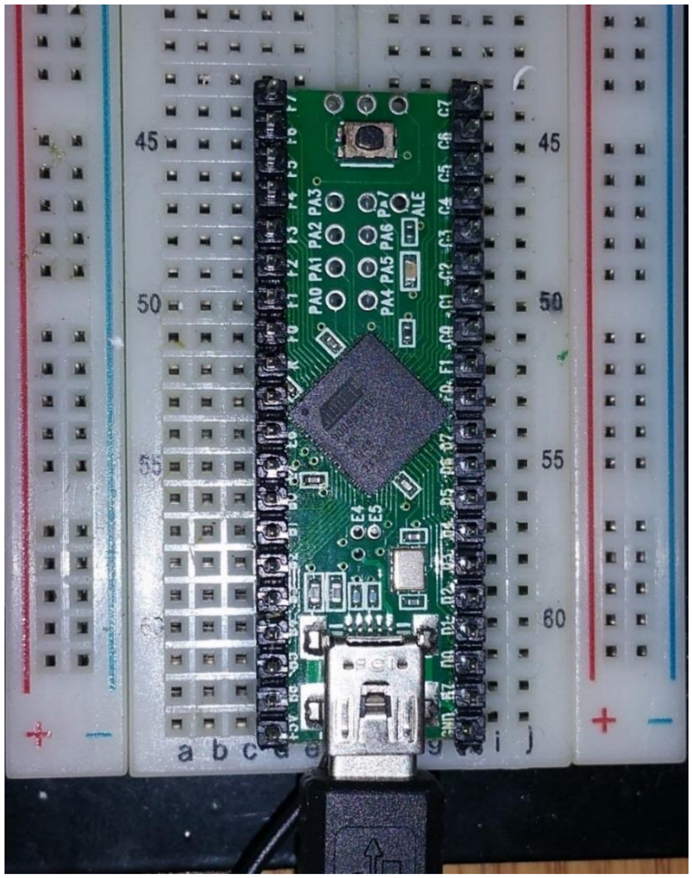


Figure 52: Teensy 2.0++

Since all control signal testing was done through the Arduino interface, the tentative TEENSY 3.2 microcontroller should theoretically be capable of the same performance due to the Texas Instruments Energia software. Energia is a direct port of the Arduino IDE which allows Arduino code to run on the MSP430 line of products.

6.5. Additional Software Features

In addition to the Android application, some small software control systems exist. These control systems include physical button input for manual servo control, Bluetooth input to initialize the automated tuning process, and information output to both Bluetooth and the LED screen. All control is maintained and regulated by the Finite State Machine (FSM).

The tuning system's software control is a Finite-State Machine (FSM). A FSM is responsible for tracking and controlling all events in a system based on the system’s current state. FSMs are useful in cases where complex systems must handle many events in a dynamic and organized fashion, and the system is expected to change states.

The system’s state helps the FSM determine what to do when an event is triggered. An event is classified as anything that is triggered by an external input to the algorithmic system. For instance, if the physical device is turned on, but no input is being received by the system, it is in an idle state. When idle, the FSM will immediately process any event that is triggered. If the device is receiving input via sound feedback, it is in the listening state. When listening, manual button input is rejected by the FSM and the servo will not be activated.

Events are classified as any external input that must be recognized and handled by the FSM. Events may be immediately processed, thrown into a queue to await processing while another event is handled, or completely ignored. Some examples of events include the pushing of a button and the receiving of data via Bluetooth or the ADC.

The following FSM diagrams describe the states and events the system monitors. There are two separate FSM’s, one for automatic tuning, and one more manual tuning. During automatic tuning, all manual tuning input is ignored. During manual tuning, all automatic input is ignored. If the system shuts down during either of these cycles, the system will revert to the Ready state on activation.

The default state of the system when turned on, is the Idle state. While in this state, all input is ignored other than the pressing of the Start button. This activates the Ready state. From this point, the user may engage either manual or automatic tuning. During manual tuning, the CW and CCW states refer to the direction the motor will turn the guitar knobs, clockwise and counterclockwise respectively. When automatic tuning, all frequency input occurs during the Listen state. Once received, the FSM continues to the Tune state where the motor will automatically tune in the required direction based on the frequency input.

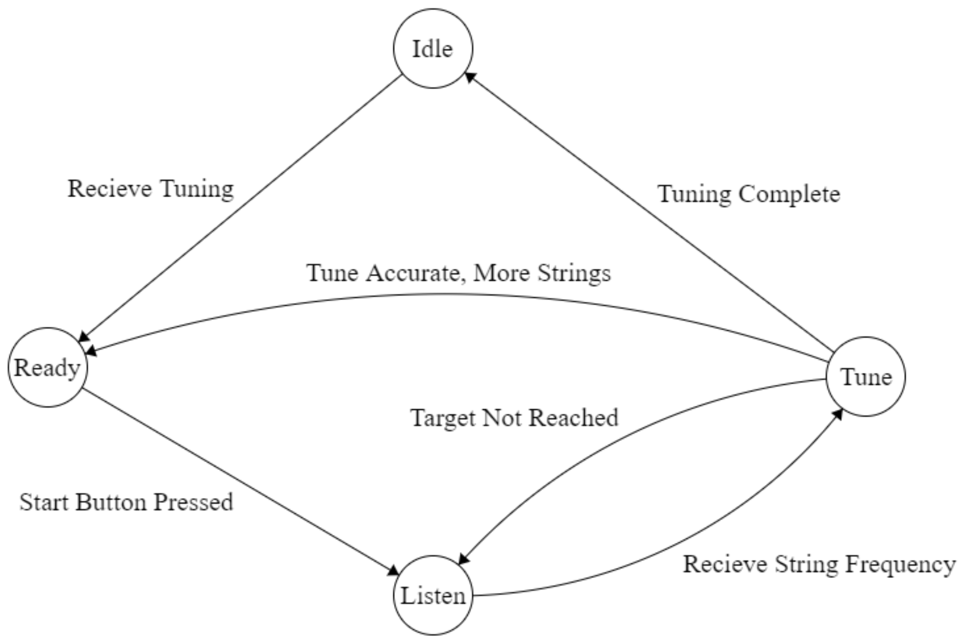


Figure 40: Finite State Machine: Automatic Tuning

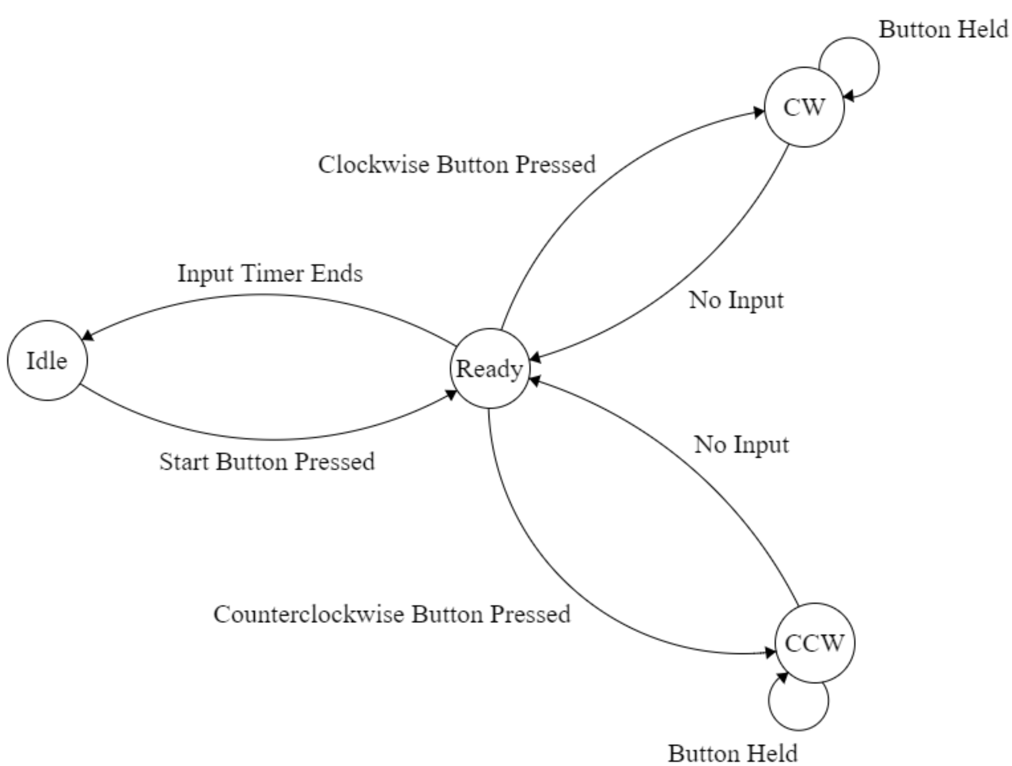


Figure 41: Finite State Machine: Manual Tuning

7. Project Testing and Prototype Construction

This project has several designs which need to be fully tested before being integrated together in Senior Design II. This section covers the testing methodology and results for each subsystem. Each subsystem was tested according to its expected use case and against the set requirements.

7.1. Prototype PCBs

Prototype PCBs will be created after testing of all subsystems but are not a main goal in this stage of the project.

7.2. Hardware Testing

Hardware testing of the display, battery charging, power regulation, motor, and buttons is be done in a manner which simulates projected use case and verifies requirements. Electrical signals are tested by analyzing voltage and current characteristics with multimeters, and waveform characteristics with an oscilloscope. Electrical circuits are tested by utilizing a breadboard to create easily modifiable circuits which can be changed as the design is narrowed down. Mechanical testing of the motor done to confirm the rotation speed and torque output meets the specification.

7.2.1. Battery Charging

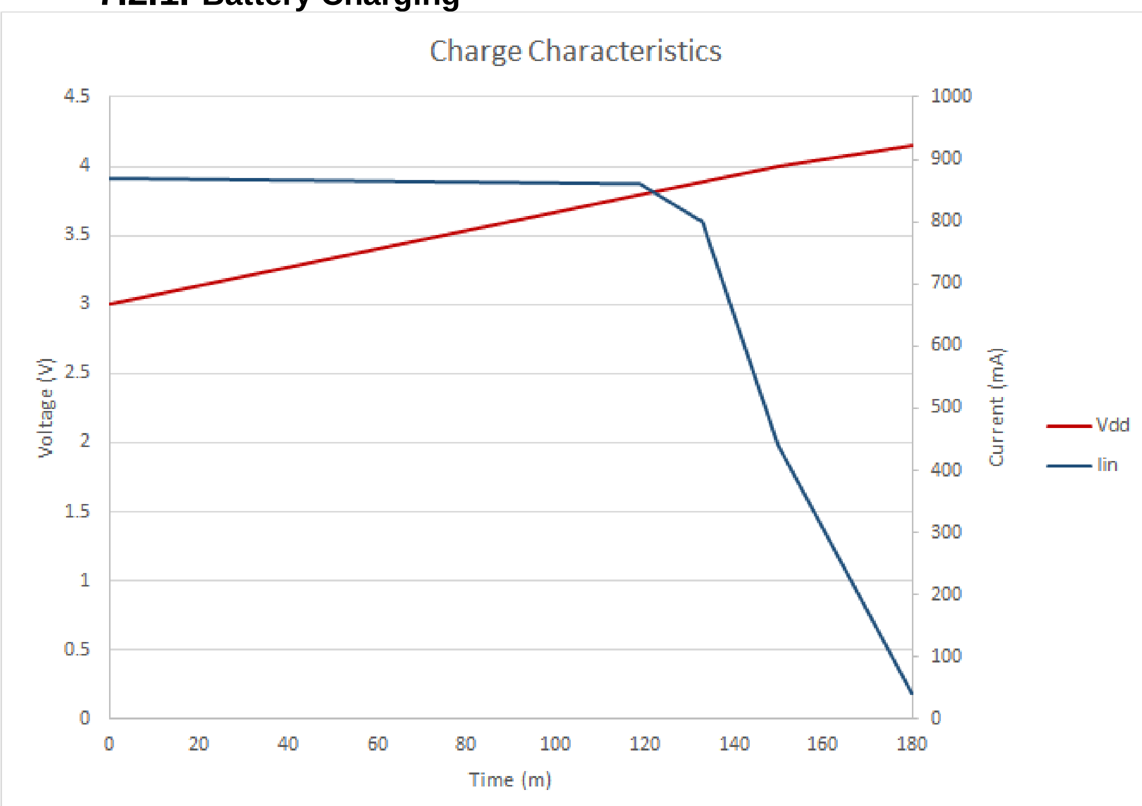


Figure 53: Battery Charge Characteristics

The battery charging circuit is tested to confirm that it works in both constant current and constant voltage mode. The output current into the battery cell along with the voltage from positive to negative terminal of the battery cell is monitored over time by utilizing two multimeters with logging capabilities. The time it takes to go from 10% “charge” to 90% “charge” according to the cell’s data sheet is recorded and plotted as a function of time. The charging circuit is also tested to detect over-temperature conditions where the battery cell crosses the recommended limit in the data sheet. This is done by monitoring the output current and voltage while applying a controlled temperature to the thermistor probe on the charging IC. When the temperature crosses the threshold, the output current should drop to zero amps until temperature drops below the threshold again. The circuit is also be tested for under voltage and overvoltage characteristics by attempting to charge off a 3.3V power source and a 12V power source. The following table outlines the testing procedure for the battery charging circuit.

Table 34: Charging Circuit Testing Procedure

Testing Process	Procedure
Step 1	Construct battery recharging circuit described in schematics above
Step 2	Connect multimeters to measure charging current and charging voltage simultaneously
Step 3	Apply 5V power to input of circuit
Step 4	Record Multimeter measurements until red light turns on indicating battery charge cycle complete
Step 5	Apply hot air to thermistor and ensure circuit turns off when temperature crosses 70 C threshold
Step 6	Adjust input voltage to 3.3V and ensure circuit turns off when the 4.5V threshold is reached
Step 7	Adjust input voltage to 12V and observe circuit turns off

7.2.2. Power Regulation

The power regulation circuit is tested to confirm all power rails provide regulated voltage and adequate current characteristics under operating conditions. The ripple voltage is tested by analyzing the signal with an oscilloscope and measuring the peak-to-peak voltage which is overlaid on the DC offset. Each power rail is tested for maximum current value using a multimeter and potentiometer to control the load. Each power rail is also tested for short circuit characteristics by shorting the rail to ground while monitoring current and voltage. The table below highlights the testing process performed on all power rails.

Table 35: Power Regulation Testing Procedure

Testing Process	Procedure
Step 1	Configure power regulation circuit with approximately 500 mA load on 7V rail, 50 mA load on 3.3V rail, and 100 mA load on 5V rail.
Step 2	Apply simulated 3.6V nominal battery voltage to input of power regulation circuit.
Step 3	Measure voltage and current of each rail simultaneously with digital multimeters.
Step 4	Analyze output waveform with oscilloscope, using the Volts-peak-to-peak measurement to analyze total ripple.

7.2.3. Display

The display needs to be of adequate brightness, size, and resolution to make standalone operation of the device possible and simple. The brightness is tested with a lux-meter which returns the brightness in lumens per sq. meter. This can be directly converted to nits, which is a measure of candela per sq. meter. Nits are commonly used to test personal technology and a value of 200 nits is seen as the minimum for indoor usage. The resolution is measured in pixels per square inch (ppi), where a value of 50 ppi is considered adequate for short spans of time. Menus are simulated on the display to test their readability.

Table 36: OLED Testing Procedure

Testing Process	Procedure
Step 1	Construct OLED display to MCU circuit
Step 2	Apply power to both MCU and OLED display
Step 3	Upload sample sketch which sends various lines of text and symbols to display
Step 4	Measure light output intensity using Lux meter

7.2.4. Buttons

Each of the buttons is a momentary switch which sends a signal to the MCU. Buttons are placed in a pull up configuration, where the pin reads normally high. When the switch is pressed, the signal at the pin pulls low and an interrupt is triggered. This interrupt samples the pin over a short period of time to confirm the button is pressed, after which is returns to the desired function for the button press. The switches need to be consistently debounced to prevent unwanted input to the program. The debounce methods are tested until a 99% success rate is achieved. The table below outlines the testing procedure for the debounce circuit.

Table 37: Button Testing Procedure

Testing Process	Procedure
Step 1	Construct debounce circuit described in schematics above
Step 2	Apply power to circuit
Step 3	Upload sample sketch which blinks LED when button is pressed
Step 4	Press button 100 times and record any instances of multiple blinks or failure to blink.

7.2.5. Sound Feedback

Table 38: Microphone Test Procedure

Testing Process	Procedure
Step 1	Construct the Circuit with Vcc set to 2.5 V
Step 2	Hook up the oscilloscope to read the voltage over the resistor
Step 3	Use a function generator hooked up to a speaker to test if the microphone is working.
Step 4	Observe the oscilloscope to see if the signal could be seen.
Step 5	Turn function generator off and strum one of the guitar strings. Be sure to have the guitar close to the microphone.
Step 6	Observe closely to see if the microphone is picking up the guitar.

The microphone was first tested by itself using the configuration seen in the figure “Microphone with Low Pass Filter” in section 5.4.1. This was done to ensure that the microphone was working as well that it could pick up the sound from a guitar being played. The reason we used a speak to ensure the microphone was working was to have a pure sine wave with a steady amplitude.

After the microphone was tested, we then tested the amplifier and filter design. This was to verify that the circuit would behave as the simulation stated it would. Using the circuit schematic found in section 5.4.3 entitled “Active Filter Design”, the circuit was tested to see the output voltage and the frequency response of the designed filter.

Table 39: Active Filter Test Procedure

Testing Process	Procedure
Step 1	Construct the Circuit with Vcc set to 5 V
Step 2	Hook up the oscilloscope to read the voltage from “Output” to Ground.
Step 3	Using a function generator, as PS1, provide the circuit with 5mV peak to peak with a 2.5V offset.
Step 4	Observe the oscilloscope readings and ensure it has an AC signal of about 500mV peak to peak signal with a 2.5V offset.
Step 5	Now set the function generator to do a sweep of frequencies from 1 Hz to 10k Hz
Step 6	Observe the oscilloscope readings and see if the output voltage attenuates when outside of the designed frequencies.

The Analog to Digital converter was also tested to ensure it would work as intended. It should be constructed as the figure in section 5.4.4. The purpose of this testing is to ensure that the I2C serial communication configuration works as well that the sampling rate is configured correctly.

The microphone is tested alongside the filter and amplification subsystem. The microphone is tested by playing a sinusoidal frequency at 1 kHz through a speaker and analyzing the output waveform on the microphone. The filter is tested by inputting a sinusoidal signal to the input and sweeping from 1 Hz to 20 kHz to ensure chosen cutoff frequencies are present. The amplifier is tested by inputting a 1 kHz sinusoid signal with a Vcc/2 DC offset and analyzing the total gain of the circuit. The gain needs to be adequate to boost the signal to readable levels, while also not peaking at Vcc and ground levels. The system was also tested using an acoustic guitar to fine tune the amplifier and filters as necessary. This will result in a more accurate system later.

Table 40: ADC Test Procedure

Testing Process	Procedure
Step 1	Construct the Circuit with Vcc set to 5 V
Step 2	Program the microcontroller to communicate with the analog to digital converter.
Step 3	Ensure that the microcontroller is communicating with the analog to digital converter
Step 4	Provide the analog to digital converter an AC signal, PS1, of 500mV peak to peak signal with a 2.5V offset.
Step 5	Check to see if the analog to digital converter is successfully converting the signal while preserving the waveform

7.2.6. Bluetooth Module

The Bluetooth module is tested to confirm it can successfully communicate with an MCU over the UART connection as well as wirelessly communicate with a phone application. The module is connected to the Teensy 2.0++ UART pins and a sample sketch provided by Adafruit is run to confirm UART communication. There is also a sample phone application which can be used to test the wireless capability of the device. The following table lists the testing procedure for this circuit.

Table 41: Bluetooth module Testing Procedure

Testing Process	Procedure
Step 1	Connect Bluetooth module to Teensy 2.0++ in normal operating configuration.
Step 2	Apply power to testing setup
Step 3	Run sample sketch to confirm UART communication and check output using Serial port monitor.
Step 4	Use sample application to confirm wireless communication working adequately at range of 5m.

7.2.7. Motor and Control

The motor is tested for its rotational speed and torque characteristics. A supply of 6V is applied to the terminals and the number of rotations is counted for 15 seconds. The motor torque is verified by creating a lever arm of 12 inches, and placing a 5.3 oz (150 gram) weight at the end, creating a 4 in*lbf torque. The motor stall torque is measured by increasing this weight until stall, where the torque is calculated by performing the calculation weight * 12 inches. The following table shows the testing procedure for the Continuous Rotation Servo circuit.

Table 42: Servo Testing Procedure

Testing Process	Procedure
Step 1	Connect servo to power source and microcontroller as shown in schematic.
Step 2	Run sample sketch and test clockwise and counterclockwise operation modes.
Step 3	Use Lever arm method to test torque output according to datasheet.

The H-bridge circuit is tested by applying power at the correct input pins and measuring the output at the output pins. The control pins are manipulated similar to how they will be controlled in the full circuit to enable forward and reverse polarity. The H-bridge is then connected to the motor to ensure the motor can turn in both directions, which is important for narrowing in on the correct output frequency of the guitar. The following table shows the testing process for the uxcell standard gearmotor.

Table 43: H-bridge/Gearmotor Testing Procedure

Testing Process	Procedure
Step 1	Connect H-bridge circuit similar to circuit shown in schematics.
Step 2	Manipulate power supply to pins in order to test control of clockwise and counterclockwise operation.
Step 3	Use Lever arm method to test torque output according to the datasheet.

7.3. Software Testing

Software testing of the microcontroller, Bluetooth chip, Android application follows the traditional IEEE model of dynamic testing and will occur after all hardware has been tested and proven functional. First, all code must go through unit testing, once passed integration testing occurs, followed by final verification and validation testing.

When unit testing, the microcontroller and Android application are tested separately as individual units to ensure each device's software functions on its' own. Both units must run various test cases, passes and fails are logged, debugging occurs, and testing begins again.

During the unit testing phase, the microcontroller must pass test cases to verify the software is prepared to accept and send data to and from the Bluetooth module, accept the DC voltages that were converted from the sampled signal, send out a voltage signal to the motor, allow manual control via button input, and send data via UART to the tuner's LED display. The Android application must be prepared to accept and receive data from the Bluetooth module, and all UI and functional requirements must be met.

Once unit testing passes completely, the Integration phase of testing commences. This means all individual units are hooked up and must function together as a complete system. This is done through various tiers of system expansion. The core of the system is the microcontroller and various subsystem units are added to the core system.

The first of these units is the ADC, the test case that the MCU is receiving the DC signal must pass. Next, the physical buttons and motor are added to the system. When the on button is depressed, the MCU must recognize this and start the motor, when the button is released, the MCU should stop the motor. Next, the LED display is connected and tested. The display does not need to output all final data at this stage, but must show some output received from the MCU. Finally, the Bluetooth module is integrated. Once again, the finalized data input and output is not necessary, but accurate recognition and communication between the MCU and Android application via the Bluetooth module is verified.

Finally, verification and validation testing commences. During this last phase of testing, the software must meet all of the tuner system's functional specifications. The system must not only meet these specifications, but must continue to meet these specifications over multiple test trials.

The LED display outputs all pre-specified data such as the current frequency the system is recognizing and the current state of the system. The Android application receives and outputs similar data to the LED display, it must also accurately send the target tuning to the MCU, which will then recognize that tuning along with the current input from the ADC, then send a signal to the motor for physical tuning. Last, and most important, the signal sent to the motor must be accurate to how off the current tune is from the final target tune. If the current tune is very close to the target, the motor should activate in shorter bursts until the current frequency is either surpassed, or achieved. If surpassed, the motor should operate in the opposite direction until the goal is achieved.

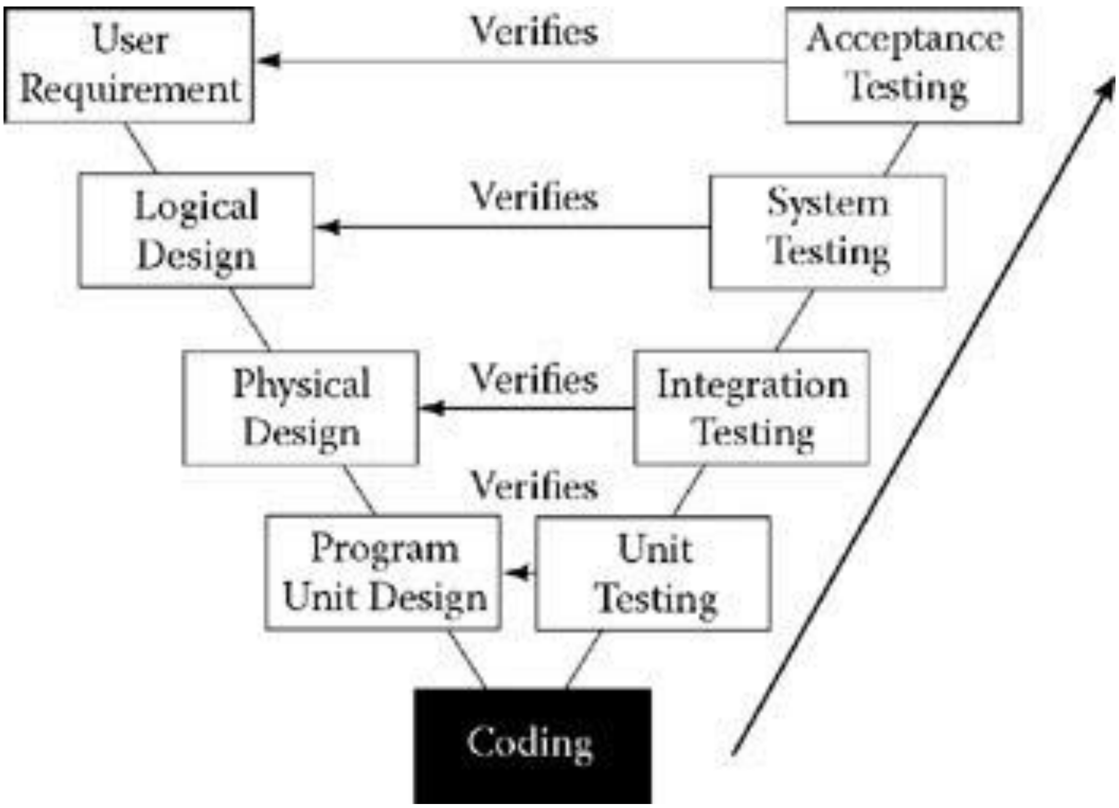


Figure 54: Verification Flow Chart

8. Administrative Content

The following section is an overview of the non-technical details of the project. This includes information such as the project's schedule, financial information, and how the workload is divided within the team. The section concludes with a forward describing the future of the project, including its intentions and goals.

8.1. Milestone Discussion

All major project milestones are displayed in the table below. Senior Design I milestones are defined by assignment deadlines put in place by the instructor. Senior Design II milestones are purely estimations as of this moment in time. The table will continuously update as future deadlines are made clear.

In addition to the major project milestones shown below, smaller less concrete milestones are made within the team on a regular basis and on the fly. Examples include researching one model of a particular component and finishing the write-up of that component within a short time period, usually a day or two.

Column2	Column3	Column4	Column5
Task	**Start**	**End**	**Status**
Ideas	1/9/2017	1/13/2017	Complete
Project Selection & Role Assignments	1/16/2017	2/7/2017	Complete
Project Report			
Initial Document - Divide & Conquer	1/16/2017	2/3/2017	Complete
Update Divide & Conquer document	2/4/2017	2/17/2017	Complete
Table of Contents	3/13/2017	3/24/2017	Complete
30 Pages	2/20/2017	3/10/2017	Complete
60 Page Draft	2/20/2017	3/31/2017	Complete
90 Pages	2/20/2017	4/14/2017	Complete
Final Document(120 pages)	2/20/2017	4/27/2017	Complete
Research, Documentation & Design	2/3/2017	4/27/2017	Complete
Final Hardware Decisions	2/20/2017	3/24/2017	Complete
Order and test parts	8/21/2017	8/28/2017	
Build Prototype	8/21/2017	9/4/2017	
Testing & Redesign	9/4/2017	9/30/2017	
Successful tests & Finalized design	10/1/2017	10/31/2017	
Peer presentation	TBA	TBA	
Final Report	TBA	TBA	
Presentation	TBA	TBA	
Senior Design Showcase	TBA	TBA	

Figure 55: Milestones

8.2. CAD Software

With the increasing processing power of computers, designers have started to rely more and more on computer aided design (CAD) when it comes to creating both electrical and mechanical assemblies. For the purpose of this project, three main CAD packages were used for signal simulation, electrical design, and mechanical design. The program, Multisim, was used to simulate the analog filter/amplifier design. DipTrace was used to create the electrical schematics. Solidworks was used to create the mechanical assembly mockup. Slic3r is used to create the G-code which is placed on a 3D printer for rapid manufacturing.

8.2.1. Multisim

Multisim is an electronic schematic capture and simulation program. Though it is not free, the electronic classes at the University of Central Florida that have a laboratory component to them utilize Multisim for the schematic capture and simulation of the circuits needed to be designed for the particular lab being worked on. Thus, utilizing an electronics lab at the University of Central Florida during open hours, Multisim can be used for free, while an option to utilize Multisim away from campus would be to download a free trial of it. The easy of use of Multisim, as well as the familiarity with the program made Multisim an excellent choice to utilize for simulating the analog filter/amplifier designed circuit for our project.

In order to simulate the analog filter and amplifier design. We modeled the circuit within Multisim. Modeling the circuit was extremely easy due to the extensive libraries and functionality of Multisim. It had the exact operational amplifier used in this project already within one of its libraries. Using the AC analysis functionality of Multisim we were able to see the frequency response of the designed circuit. We used this graph within our paper to show how the designed system would respond to different frequencies. This allowed us to tune the design in such a way that allowed to maximum filtering of the unwanted signals. We also used the built in oscilloscope functionality of Multisim to see how each stage of the filter and amplifier attenuated or amplified the signal. This allow us to be able to diagnose issues with the designed schematic as well as optimize it. While Multisim is still just a simulation, it provided a quick and easy to use solution in order to design the analog filter and amplifier the best that it could be.

8.2.2. Diptrace

DipTrace is a schematic and printed circuit board design software. There are multiple tiers of the program that can be purchased, however for this design we used a free nonprofit version, DipTrace 3.0 Freeware. This program is very intuitive, allowing the students to easily utilize the program to fulfill their needs. This program allows for easy to construct and easy to read circuit designs. While it does not simulate the circuits, it provides a very clean look. This is the tool used in all of our schematic diagrams. This tool is also especially useful when constructing a printed circuit board. DipTrace has the ability to export the circuits into model that can easily be sent to a manufacturer. This ability will increase efficiency later down the line when it comes time to order the printed circuit boards.

Some of the features DipTrace offers is over 50 libraries, sorted by manufacturer, of chips. While these libraries are not a complete list of all the chips used by all the manufacturers, it provides a solid foundation to start with. However, some of the chips used in this design were not found within the existing libraries. For some software, this could be a major obstacle. However DipTrace allows for the users to create custom libraries. This allowed us to create nine chips that were not included in the default libraries. These custom parts can be configured in multiple ways, including what the symbol looks like, the pin layout, and what the actual configuration of the chip is. This customization allows for the custom parts to be able to be used in all the different applications described above. By using this one program multiple efficiencies were found and will be used for the rest of the project's development.

8.2.3. CAD/CAM Software

Solidworks was chosen because UCF provides the student package for free, and one team member was most familiar with its workings. Solidworks is a powerful parametric CAD software which allows for parts to be created by using a combination of simple 2D sketches and 3D extrudes/cuts. These parts can then be placed together in an assembly and the "Mate" function is used to constrain them together. There are also many other functions which can be used to intelligently create screw holes and mounting points, especially utilizing the 3D file export from DipTrace when the PCBs are fully designed in Senior Design II. These Solidworks parts can be directly exported to .stl files which are the de facto standard for 3D printing programs which slice the mesh file into G-code which can then be exported to a 3D printer for rapid prototyping. Once the initial print is tested and all flaws are noted, features can be quickly modified and the model reprinted within the same day.

Slic3r was chosen due to its familiarity with one team member and pre-existing profiles for personal 3D printer, a Rostock Max V2. This software is an open source slicer, which is a program that takes a digital 3D model and converts it into instructions for a 3D printer to follow. The model is oriented within the program's 3D model viewer, then cut into individual layers in the Z direction. There are options within the program's menus which allow modification of many print features, such as layer height, number of perimeters, infill percentage, etc. As such, this program will be heavily used in conjunction with Solidworks to cover all mechanical CAD/CAM development which will be a priority in Senior Design II. The goal is to create an ergonomic device which fits into the average user's hand and is easy to maneuver into position.

8.3. PCB Vendor

Finding a reputable PCB vendor who can deliver a quality product, in a short amount of time and cheaply is difficult. The first PCB manufacturer we looked at was OSHPark. They manufacture PCB in the United States and have free shipping. The way they work is by collecting multiple PCB designs from multiple people and put them together onto a panel. They then will send it out to be fabricated. This practice allows for cheaper fabrication. The second manufacturer that was considered was Advanced Circuits. They also manufacture PCB in the United States, but they do not have free shipping. They offer a student discount that makes their price comparable to OSHPark.

OSHPark's two-layer board come with FR4 substrate, purple mask over bare copper and ENIG finish, while their four-layer board comes with FR408 substrate, purple mask over bare copper and ENIG finish. Their board thickness is 1.6mm for both types of layers.

Advanced Circuits boards come with FR4 substrate, green mask over bare copper and a lead-free HAL finish. They also state that if the lead-free HAL finish is temporarily unavailable, a free upgrade to silver or ENIG will be given.

Table 44: PCB Manufacturer Comparison

Manufacturer	2 Layer PCB	4 Layer PCB	Shipping	Shipping Time
OSHPark	$5/sq in, includes 3 copies	$10/sq in, includes 3 copies	Free	12 Day for 2 layer 2 weeks from Fabrication for 4 layer
Advanced Circuits	$33 for 60 sq in	$66 for 60 sq in	$10 Handling Fee	Standard 5 Days

OSH Park Requirements for Two Layer Board:

1. 6 mil trace clearance
2. 6 mil trace width
3. 13 mil drill size
4. 7 mil annular ring

OSH Park Requirements for Four Layer Board:

1. 5 mil trace clearance
2. 5 mil trace width
3. 10 mil drill size
4. 4 mil annular ring

Advance Circuits Requirements

1. 5 mil trace clearance
2. 5 mil trace width
3. 10 mil drill size
4. 5 mil annular ring

While Advance Circuits is cheaper per sq. in. at the full size, OSH Park is going to be cheaper for the PCB needed for this product, especially since OSH Park provides 3 copies of the PCB for their price. Advanced Circuits does have a much shorter and more dependable lead time however. This could be critical should we have issues late in the design. Their design requirements are also very similar. In terms of technical support, OSH Park has an email that you can send questions to, while Advanced Circuits has a 24 hour 7 days a week technical support email and phone numbers for representatives across America.

For this product, OSH Park was picked as the manufacturer of choice. For this design, they are going to be significantly cheaper to use, at $5 per sq. in and free shipping, since the PCB is not going to be anywhere close to 60 sq. in Advance Circuits provides. In addition, Advance Circuits price was a special for students. If this product goes to market, we could not use their pricing as a guide to see how much it would cost to fabricate. The multiple copies of the PCB also help mitigate the risk of the longer lead time in case there are issues with one of the PCBs they deliver.

8.4. Budget and Finance Discussion

Component	Cost
Board Development Cost	$50
Servo	$40
Casing	$10
Microphone	$20
Rechargeable Battery	$20
Battery Charging Circuit	$20
Charger	$5
Unforeseen Costs	$150
OLED Display	$10
Total Development Budget	$325

8.5. Project Design Problems

Overall, this project has come together very well with successful subsystem tests and clean designs. There were, however, a few difficulties which impeded progress in the design stage. Automatically tuning a guitar string is a complicated task which crosses over many different areas of study in both electrical engineering and musical instruments. This team had to learn the language of musicians, study analog processing and digital communication circuits, and learn the intricacies of embedded processing along with high level Android phone applications.

The first difficulty came in the form of a barrier to entry that all musicians face when starting to play an instrument for the first time. Nobody on the team can actually play guitar, so research was initially difficult due to lack of vocabulary knowledge. As each team member researched their own subsystem and learned the required components and specifications, we also researched how the musician would approach the problem. Things which may be common knowledge to a musician, like the difference between an octave and a cent, were learned. There is still more to be figured out, such as the applications of up-tuning and down-tuning and how they will affect the ending algorithms which will control the frequency adjustment.

This project contains a large amount of both analog and digital signal processing. The classes given at UCF have focused mainly on the theory behind these subject areas and only give a broad look at implementation. For this reason, this group needed to learn the intricacies of filter and amplifier design, inter-device communication, and wireless communication. The filter and amplifier design were learned in the Electronics II class, though there were some design restraints which added difficulty. The amplifier circuit needed to work from a single supply, which after much research, led to the conclusion that a Vcc/2 input would be needed in place of a ground reference, which added confusion when dealing with DC filter capacitors. Inter-device communication is dealt with by usage of both synchronous I2C serial protocol, and universal asynchronous receiver/transmitter (UART) serial protocol. This played into MCU choice greatly, as the MSP4302553g contains pins for each of these protocols and a large amount of support to achieve a working circuit. Finally, wireless communication was heavily researched, as it is not covered in practice in class. The final conclusion to use Bluetooth Low Energy (BLE) was decided and a module was chosen. Due to the high frequencies of the radio signal, it would have been impossible to test our custom circuit on a breadboard, so only the breakout board was tested.

Lasly, the project has an interesting combination of both low-level embedded programming in C, and high level programming and Java. These two languages contrast greatly and require a different approach of thinking. Overall, it will prove an interesting challenge to develop the codebase required to achieve the end usage goals as this group progresses to Senior Design II in the Fall 2017 semester.

8.6. Project Roles

The project has been divided up into four sections. Each member of the team is responsible for a section of the project in it’s entirety. This includes both research and implementation. All members do some work in every section and assist in each section as needed.

8.6.1. Adam Harmon (EE)

I am responsible for the Bluetooth communication between the Automatic Guitar Tuner and the cell phone app that is being made for this project. This includes choosing the Bluetooth module to use for this project. I am also responsible for any button input that is done on the device. This includes the design and implementation of the momentary switch.

8.6.2. Bryan Casey (CpE)

I am responsible for all software dependent components of the project. This means primarily the microcontroller and the Android application. I am also responsible for making sure the software communicates with and, if applicable, properly controls other components of the system. This includes control over the Servo motor via manual button and automatic inputs, Bluetooth communication between the Android application and the physical tuning device, and informative output to the LED screen.

8.6.3. John Geiger (EE)

I am responsible for the battery charging, power regulation, and display components of this project. The battery charging and power regulation responsibility has proven to be a great challenge in learning the intricacies of battery circuits and how to power multiple devices from a single source. The display section has also given some great insight on the human factor of a handheld device and how it can make or break the usability of a product.

8.6.4. Jason Lupo (EE)

I am responsible for the audio signal processing components, including the microphone, amplifiers, filters, and analog to digital converters. Also, I am responsible for the servo control. This has been a significant challenge in term of the amount of research need fully understand how the sound from instruments could be captured and processed. The filter and amplifier design also provided some difficulties.

8.7. Looking forward

The integration of all parts of this project will be performed in the Fall 2017 semester at the University of Central Florida as part of the Senior Design II class. At the culmination of this class, this group will have completed a full design project, from initial project brainstorm, to full implementation at a functional level. This project will be judged by professionals in academia and industry for its technical achievement and professional appearance/function.

9. Appendix A - Works Cited

[1]: "Boston Globe Online / Business / USB deserves more support." Boston Globe Online / Business / USB deserves more support. Boston Globe Online, 31 Dec. 1995. Web. 30 Mar. 2017 [Online] https://simson.net/clips/1999/99.Globe.05-20.USB_deserves_more_support+.shtml

[2]:"Wi-Fi Direct." Wi-Fi Alliance. Wi-Fi Alliance , 30 Mar. 2017. Web. 30 Mar. 2017 [Online] http://www.wi-fi.org/discover-wi-fi/wi-fi-direct

[3]: "How it works | Bluetooth Technology Website." Bluetooth. Bluetooth , n.d. Web. 30 Mar. 2017. [Online] https://www.bluetooth.com/what-is-bluetooth-technology/how-it-works

[4]:"Bluetooth Architecture Overview" Microsoft, Mar 2011, 11 Apr. 2017 [Online] http://download.microsoft.com/download/2/4/A/24A36661-A629-4CE6-A615-6B2910A1367A/Bluetooth%20Architecture%20Overview.pdf

[5]: "Host Controller Interface (HCI)" Bluetooth, 13 Apr. 2017 [Online] http://www.amd.e-technik.uni-rostock.de/ma/gol/lectures/wirlec/bluetooth_info/hci.html#RS232 Transport Layer

[6] UM10204 I2C Bus Specification And User Manual. (2014, April 04). Retrieved April 11, 2017, from http://www.nxp.com/documents/user_manual/UM10204.pdf

[7]: Cambridge Silicon Radio BC417 2.4 GHz Bluetooth Radio chip datasheet https://cdn.sparkfun.com/datasheets/Wireless/Bluetooth/CSR-BC417-datasheet.pdf

[8]:HC-05 information source http://ijsetr.org/wp-content/uploads/2016/04/IJSETR-VOL-5-ISSUE-4-869-872.pdf

[9]: HC-05 information source https://arduino-info.wikispaces.com/BlueTooth-HC05-HC06-Modules-How-To

[10]: Adafruit's Bluefruit LE nRF8001 Breakout https://cdn-learn.adafruit.com/downloads/pdf/getting-started-with-the-nrf8001-bluefruit-le-breakout.pdf

[11]: nRF8001 product specification sheet https://www.nordicsemi.com/kor/content/download/2981/.../nRF8001_PS_v1.3.pdf

[12]: Bluefruit Bluetooth LE UART Friend info https://learn.adafruit.com/introducing-the-adafruit-bluefruit-le-uart-friend?view=all

[13]:nRF51822 module info https://www.nordicsemi.com/eng/Products/Bluetooth-low-energy/nRF51822

[14]: RS232 Standard information http://www.omega.com/techref/pdf/RS-232.pdf

[15]: “How Do Microphones Work” Media College 1995-2012, 20, Apr. 2017 [Online] http://www.mediacollege.com/audio/microphones/how-microphones-work.html

[16] “Introduction to Integrated Circuit Temperature Sensors” Omega 2003-2017, 17, Apr. 2017 [Online] http://www.omega.com/prodinfo/Integrated-Circuit-Sensors.html

[17] “Preventing Switch Bounce in Low-Profile Keypad Assemblies” Epec 2017, 24 Apr. 2017 [Online] http://blog.epectec.com/preventing-switch-bounce-in-low-profile-keypad-assemblies

[18] “Schmitt Trigger Circuit Tutorial” Lammerbies Apr 2015, 23 Apr. 2017 [Online] https://www.lammertbies.nl/comm/info/Schmitt-trigger.html

[19] “String Tension Guide” Violin String Review, 19 Apr. 2017 [Online] http://www.violinstringreview.com/tension-chart.html

[20] “USB3.1: Physical, Link, and Protocol Layer Changes” Synopsys 2017, 15 Apr. 2017 [Online] https://www.synopsys.com/designware-ip/technical-bulletin/protocol-layer-changes.html

[21] “Bluetooth Security” eTutorials 2008-2017, 05 Apr. 2017, [Online] http://etutorials.org/Microsoft+Products/windows+xp+unwired/Chapter+6.+Bluetooth/6.7+Bluetooth+Security/

[22] “Java Platform Standard Edition 8 Documentation” Oracle 1993-2016, 17 Apr 2017, [Online] http://docs.oracle.com/javase/8/docs/index.html

[23] “Android Distribution Numbers” Droid-Life 2009-2017, 09 Apr. 2017, [Online] http://www.droid-life.com/2016/04/05/android-distribution-update-for-april-2016/

10. Appendix B - Copyright Permissions

John Geiger
Today, 2:39 PM
band@bandindustries.com

Reply all

To whom it may concern,

I am currently writing documentation for a University of Central Florida Senior Design project which references your product for comparison. Will you let me use an image of your Roadie Tuner product?

Thank you,

John Geiger
University of Central Florida EECS Student
jgeiger@knights.ucf.edu

Senior Design - Copyrighted Information Permission

John Geiger
Today, 3:00 PM
copyrightcounsel@list.ti.com

Reply all

To whom it may concern,

I am currently writing documentation for a University of Central Florida Senior Design project which contains two Functional Block Diagrams for the MSP430G2xx series and MSP430x5xx Series for MCU comparison. This information is being used for a proof of concept Automatic Guitar Tuner design in an Academic environment and falls under the classroom usage scenario. Both figures are noted as "Courtesy of Texas Instruments".

Thank you,

John Geiger
University of Central Florida EECS
Senior Design – Automatic Guitar Tuner
jgeiger@knights.ucf.edu

JG John Geiger
Today, 3:08 PM
copyrightpermission@intel.com

Reply all

To whom it may concern,

I am currently writing documentation for a University of Central Florida Senior Design project which references your product for comparison. Will you let me use a figure of the Intel Quark D1000 Functional Block Diagram for purposes of MCU comparison? This project is strictly academic, with the goal of creating a proof of concept Automatic Guitar Tuner. This figure will be reproduced with a note saying "Courtesy of Intel".

Thank you,

John Geiger
University of Central Florida EECS Student
jgeiger@knights.ucf.edu

Educational and Non-Profit Use of Copyrighted Material: If you use Microchip copyrighted material solely for educational (non-profit) purposes falling under the “fair use” exception of the U.S. Copyright Act of 1976 then you do not need Microchip’s written permission. For example, Microchip’s permission is not required when using copyrighted material in: (1) an academic report, thesis, or dissertation; (2) classroom handouts or textbook; or (3) a presentation or article that is solely educational in nature (e.g., technical article published in a magazine). Please note that offering Microchip copyrighted material at a trade show or industry conference for the purpose of promoting product sales does require Microchip’s permission.

http://www.microchip.com/about-us/legal-information/copyright-usage-guidelines