eGuitar

Electronic Guitar Assistant

ABSTRACT

eGuitar aims to be an all-encompassing guitar assistant. Whether there is a beginner learning their first chords or an experienced guitarist playing their own songs, the eGuitar system has benefits for all users.

BY

Brandon Berk, William Remington, Eric Sorokowsky

EEL 4915

Spring 2015

Group 7

Table of Contents

1 Executive Summary

The eGuitar is a portable guitar assistant that aids both beginner and expert players. There are millions of people around the world who actively play the guitar, and they have learned to play in a fairly analog way. A typical guitar lesson includes opening up a book and reading the notes written for the beginner songs. The beginner learns to read the tablature provided and then learns the finger placements that correspond to each note. The process of learning the finger placements can be challenging for certain beginners. For many, there is only the feedback of hearing the wrong note, but no indication of which finger is placed incorrectly. The eGuitar system helps beginners by visually showing where their fingers need to be placed, and actively tracks what they are playing to determine how they are progressing. The eGuitar system has the ability to enhance the learning process for beginning players.

After beginning players start to learn the basics, they can start using other guitar products to fine-tune their guitar playing. For instance, a metronome can be purchased to help the player stay on beat. Another example is a guitar tuner which helps keep the player’s guitar in tune. The eGuitar system incorporates not only LED indicators, but also a metronome, guitar tuner, and active digital signal processing of what the user is playing. With the built-in digital signal processing functions of the eGuitar system, the user is given the tools to improve their guitar experience without the addition of any complex hardware. All of the extra features can be accessed using the eGuitar’s intuitive user interface in the form of an OLED screen with a knob to select menu options.

For expert players, the LED fretboard matrix may not be the most useful feature, since the player already knows the proper finger placements and chords. However, it is when expert players use the full functionality of the DSP features that the true power of the eGuitar system shows. With a hexaphonic pickup connected to a windows PC, expert guitar players have the ability to write tablature without ever picking up a pen. The eGuitar system is also able to actively record the notes played by the user and create tablature on a host PC. With this feature, guitarists can create their own songs quickly and easily and share them with the world.

This document details the research, design, and prototyping work that went into the creation of the eGuitar system. Some of the fields that were researched include communication methods between subsystems, printed circuit board design and materials, and digital signal processing. Some design considerations were the form factors of the LED fretboard matrix, whether they should lie on the top of the neck of the guitar or underneath the strings. Once the subsystems have been designed, the different components were formed into prototypes that undergo varies test procedures. The testing helped fix issues within the designs to ultimately bring the best experience to the user. Finally, the full design was implemented on a custom PCB and windows application.

2 Project Description

2.1 Contributors and Sponsors

Other than the properly cited usage of open source software, the group members were the only contributors to this project. While the group members attempted to procure sample parts to save money, this project was funded "out-of-pocket" with no official sponsors.

2.2 Project Motivation

This project idea was conceived and motivated by the struggle that many would-be musicians undergo when trying to familiarize themselves with a guitar for the first time. There is a significant learning curve involved with any new instrument but the aim of the eGuitar is to minimize this learning curve and make guitar tablature more tactile and natural. A person who decides they want to learn guitar should be able to learn quickly and receive feedback about their performance without the definitive need for a guitar instructor. With the help of the eGuitar, honing of coordination and speed should be accelerated beyond the standard experience of learning an instrument.

2.3 Objectives and Goals

The eGuitar system should be more affordable than a paid instructor and the sum of the leading individual products whose feature set is included in the system. It should also have unique features that make it attractive to a learning guitarist. The eGuitar ecosystem should provide the user with a convenient means of learning alongside recorded tablature. The guitarist should have visual cues for which frets need to be played, audible feedback for the tempo of the song (metronome), accompaniment, and/or the current pitch to be played, and an on-board chromatic tuner (with audible reference pitches) for tuning their guitar. In terms of special features, the guitarist will have the ability to loop certain portions of the song with increasing tempo (speed-training). The guitarist will also receive constructive feedback metrics about their performance and accuracy in playing the right notes at the right times.

2.4 Project Requirements and Specifications

2.4.1 LED Indicator Specifications

The following list details the specifications laid out for the LED Fretboard Matrix. There are two individual design ideas for the placement of the LED matrix, therefore two separate specifications have been detailed.

- For LED placement under each string
    - Height of LEDs must be less than 1 mm.

    - Width of LEDs must be less than 3 mm.
    - Combined resin enclosure for wires and LEDs must remain under 1 mm.
    - Height of PCB must be less than 1 mm.
    - Width of PCB must be less than 20 mm.
    - Length of PCB must be less than 42 mm.
    - Common ground line against top of guitar neck.
- For LED placement on top of guitar neck
    - LEDs are embedded in line on custom PCB.
        - Height of PCB must be less than 4 mm.
        - Width of PCB must be less than 20 mm.
        - Length of PCB must be at least 150 mm long.
    - For 6 frets, total of 36 LEDs must be included.
    - Custom printed enclosure is used to easily read which numbered LED is on, as well as hide connections.
        - Height of enclosure should be less than 8 mm.
        - Width must be less than 20 mm.
        - Length must be at least 150 mm long.
    - All data lines are connected to the main control board via ribbon cable to prevent large bundles of wires.
        - Thickness of ribbon cables must be less than 5 mm.
        - Length of ribbon cables must be at least 2 ft. long to connect to Control Board at the base of the guitar.

2.4.2DSP paired with Hexaphonic Pickup

The following list details some simple requirements for the eGuitar's integrated guitar pickup and DSP.

- DSP chip must be able to handle 6 individual signal inputs from polyphonic pickup
    - 6 individual magnetic pickups that encode note played into a signal.
    - Each individual pickup must be sensitive enough to not receive "data" from other strings being played.
    - Support frequencies between 50 Hz and 2000 Hz with accuracy of 0.1Hz.
- DSP chip would also be source of LED control.
    - I2C I/O Expanders for control of at least 36 LEDs.
- DSP must be able to communicate via RS232 /USB to host PC.
- DSP can communicate over I2C to external display.
    - 128x64 LCD.
- Potentiometers used for user input.
    - Rotary potentiometer – 10k Ohm, Linear.
- Push buttons used for user input.
- Size of PCB with DSP chip and other components must be no larger than 100x100 mm.

2.4.3Power for Control Board

The following list details some simple requirements for powering the control board.

- Rechargeable 5V battery.
  - Powers control board and all I/O connections.
  - Rechargeable via USB connection to PC or proper wall adapter

2.1 Standards

The following list details some standards utilized by the eGuitar System.

- I^2C
  - I^2C or Inter-Integrated Circuit is a protocol developed by Phillips Semiconductor for communication between peripherals and processors operating at different speeds. It consists of several standards for messaging protocols as well as voltages and number of physical lines. For more information see section 3.2.5.
- SPI
  - SPI or Serial Peripheral interface is a specification for synchronous serial communication over short distances. SPI was developed by Motorola and has evolved to be the go to standard for communication between peripherals in embedded systems. For more information see section 3.2.4
- RS-232
  - RS-232 is a standard created by The Electronic Industries Association in 1969. This standard defines the characteristics of the electronic signals such as voltages, signaling rate and timing. In addition it defines what each wire of the interface is responsible for. The standard does not however define the format that the data is transmitted in. Elements such as character encoding, framing and error correction are up to the implementers.

3 Research

3.1 Existing Products

In order for the eGuitar team to create and innovative and novel solution to the problem of guitar learning it is important to analyze products currently on the market. These products range from conventional books and videos such as Gibson's Learn & Master Guitar, to augmented guitar systems such as Rocksmith and FretLight. Each of these takes a different approach to teaching someone the basics of playing a guitar according to their target audience.

3.1.1Gibson's Learn & Master Guitar

Gibson's Learn & Master Guitar is an attempt at teaching guitar in a more classical sense. It utilizes a set of DVD videos and written lessons as well as play along audio. This method requires the user to spend weeks to months of independent study to learn the basics of playing a guitar. It can be viewed similarly to home schooling. The learner is to follow each DVD and follow along in the provided book. Along the way they are coached through new techniques and tested to see if they can advance to the next section. It has the added advantage of containing a comprehensive book that details remedial lessons if the user wishes to brush up on one aspect of their guitar playing or if the user feels as though they are not ready to proceed to the next section.

3.1.2Rocksmith

Rocksmith is an attempt to teach the basics of playing a guitar as well as teaching specific songs using a video game to facilitate learning. This product allows the user to use their personal guitar and connect it directly to a gaming system. The game software then teaches the user by walking them through the basic cords of the guitar as well as finger position. In order to motivate the user they are tested along the way by playing songs and receiving a score depending on how close they were to the correct notes. Since the user is using a standard guitar the game software features the ability to decode the mono output from the guitar into cords and notes that it can compare against. This creates a welcoming and effective learning environment however since Rocksmith is using a standard gaming system it lacks a dedicated DSP. This causes the software to misinterpret certain notes and cords thus giving false positives to the user when they may have played the incorrect cord.

3.1.3FretLight

FretLight is an attempt to teach the basics of playing a guitar by using a modified guitar. This system comes in many varieties from acoustic to bass guitars and electric guitars. Each of these versions features a modified fretboard that houses LED’s under each string. They vary from only supporting the first five frets for beginners to supporting the entire fretboard for the more advanced users. Although this system uses a modified standard guitar it requires the user to purchase the guitar made by FretLight since the system is not designed to be added to a guitar after initial production. In addition this system contains no way to measure the accuracy of the user since it does not analyze finger placement or the produced audio. This makes this system excellent for learning the basic finger positions but it quickly loses its usefulness once the learner has advanced beyond learning basic finger positioning.

3.2 Communication

Communication between devices is an integral component to this projects success. Data will need to be relayed between numerous embedded processors, in order to facilitate a division of labor between all components. As well as communication between embedded devices, relatively large amounts of data will need to be transferred back and forth between a Personal Computer (PC) and the embedded components. Although audio input is not typically thought of as communication, embedded devices will treat it as though it is an asynchronous data stream being fed in.

3.2.1UART

Universal Asynchronous Receiver/Transmitter (UART) is a piece of hardware commonly included in embedded platforms to facilitate serial communication. The UART chip allows for simplex (only one direction), half duplex (devices trade of transmitting and receiving) and full duplex (both devices can send and receive at the same time) communication which allows compatibility with a large range of serial devices and serial applications. To support different serial protocols UART allows the user to configure several settings such as voltage, number of parity and data bits, and data transfer rate in the form of symbols per second. In general UART converts individual characters into bits which are then combined with several control bits that are required by the protocol. These bits are then sent to another device, using a variety of standards such as RS-232 or RS-423, then the receiving device decodes the data into a format that it is able to understand. Although UART is common across embedded applications and facilitates serial communication, it is not compatible with the serial style communication used by the USB standard. In order to communicate with USB devices a separate conversion chip known as an FTDI is required.

- Allows voltages of 1.8V, 2.5V, 3.3V, 5V, or 12V
- Supports simplex, half duplex and full duplex serial communication
- Uses FIFO for sending and receiving of data
- Buffers data to be sent or received
- Supports sleep and low-power modes

3.2.2USB

Universal Serial Bus (USB) was developed in 1996 to create an industry standard for communication and connectivity. USB is available on almost all computers today and allows for data transfer as well as transfer of power to an external device. Although many different standards exist for use in this project USB 2.0 is the focus since it is standard on most devices today. USB 2.0 is capable of transfer 280 Mbit/s or 35 Mbit/s which enable rapid transfer of large data packets. As well as communicating data USB 2.0 allows us to deliver power to the device at 5 volts for operation or charging of devices. USB is based upon serial communication at its core and allows for computer applications to communicate by use of virtual communication ports. This feature enables the user to rapidly create applications that can communicate with embedded devices.

- Operates using 5 volts
- Transmits at 280Mbit/s and receives at 35Mbit/s
- Serial style communication standard

- Can charge/power devices while transmitting data.

3.2.3FTDI

Future Technology Devices International (FTDI) is a company founded in March of 1992, which specializes in creating devices to interface between RS-232 and USB devices. Although the company manufactures numerous devices there most common product, a USB to RS-232 serial adapter, is referred to as an FTDI board. This chip allows the user to easily connect a UART RS-232 device to a USB 2.0 cable for communication via virtual communication ports. As well as facilitating data transfer this chip allows power to be sent to supported devices for use in operation or battery charging.

- Operates using 5 volt USB input
- Translates USB 2.0 to serial communication
- Allows programing and data transfer to embedded devices

3.2.4SPI

Serial Peripheral Interface (SPI) was created by Motorola to facilitate synchronous serial communication between devices over short distances. SPI is favored in systems involving numerous sensors and embedded systems due to its simple implementation and reliability. SPI utilizes a master slave design to facilitate digital synchronous communication between two or more devices at a time. The master device generates the clock signal, usually less than 5 MHZ, for use by all connected devices and has to instruct each device when it can transmit data. Four communication lines are utilized in SPI along with a ground line which earned it the names four-wire serial bus and Synchronous Serial Interface (SSI). Communication data is sent between devices across the Master Output, Slave Input (MOSI) and Master Input, Slave Output (MISO). The remaining two lines, Slave Select (SS) and Serial Clock (SCLK), allow the master device to specify which slave to communicate with and transmit the global master clock to all connected devices to ensure each device stays in sync.

The design of SPI provides support for full duplex communication, which means devices can send and receive data between the master and slave device during the same clock cycle. Full duplex compatibility leads to faster transfer rates since data can be sent in parallel instead of waiting for the next clock cycle to transmit data. SPI has one negative as no official standard has been published. This enables each manufacturer to create their own version of SPI to suite their specific needs, which can lead to errors occurring when interfacing devices created by different manufactures.

- Master/slave style communication
- Utilizes 4 wires to communicate between multiple devices
- Unified clock across all devices

- Full duplex communication
- Utilizes Interrupts to notify processor data has been received

3.2.5 I^2C

Inter IC (I^2C) is a synchronous serial communication bus created by Phillips Semiconductor in 1982 to attach peripherals to personal computer motherboards. I^2C is commonly used to connect devices on a single circuit board and devices that are connected via cables. Originally the maximum data transfer speed for I^2C was 100 Kbit with a "fastmode" that allowed a 400 Kbit transfer rate when required by certain devices. In 1998 I^2C was upgraded to support speeds up to 3.4 Mbit to facilitate transfer of larger data sizes.

The design of I^2C requires the connection to be half-duplex meaning communication data can only be sent in one direction at a time. Similarly to SPI I2C utilizes a master slave relationship however it only operates using two data lines and a single grounding wire. The first data line is the Serial Clock Line (SCL) which transmits the master clock to all slave devices. The second data line is the Serial Data Line (SDA) which is used to send data in one direction between slave and master devices. I^2C can operate over 5 volts or 3.3 volts under normal conditions although other voltages are permitted by the standard. The I^2C standard allows multiple slave devices to connect to a single master device; however there is no way to specify which device to transmit data to. This functionality can be added however with the addition of another control line or in software with packet headers.

- Operates at 3.3V or 5V
- Master/slave style relationship
- Can be extended to support multiple slave devices
- Uniform clock signal across devices
- Half-duplex support only
- Used for low-speed communication between devices

3.2.6 uPP

Universal Parallel Port (uPP) is a communication port that is included with numerous Texas Instruments DSP processors to facilitate high speed parallel data transfer. This device utilizes two independent data buses and minimal control lines. This enables data transfer in both directions in parallel with other data transfers significantly reducing transfer time of large or complex data sets. To further increase data transfer speeds the protocol allows data to be sent at the rise and fall of a clock cycle which can double the data transfer rate. In order to minimize the overhead on the CPU uPP incorporates a Direct Memory Access (DMA) controller into the hardware. This feature enables the uPP controller to fetch information directly from memory, this frees up the CPU to perform other tasks instead of waiting for relatively slow memory accesses to be completed.

uPP will enable use of multiple DSP chips as well as Field-Programmable Gate Array (FPGA) devices to be utilized in parallel for the multiple complex calculations required to analyze multi-channel audio data.

- Utilizes two separate data buses for parallel data transfer
- Can transmit and receive at the same time
- Utilizes built in DMA for faster data transfer
- Contains its own controller to free up CPU

3.2.7EMIF16

External Memory Interface Peripheral (EMIF16) is a module provided on a variety of Texas Instruments DSP processors utilizing a 16 bit data bus. This bus module is designed specifically to interface with multiple types of external asynchronous memory devices. The EMIF16 design allows access to 256MB of memory utilizing four chip select lines. Each chip select line is capable of accessing 64Mbytes of memory each. Multiple types of asynchronous memory devices such as Asynchronous Static Random Memory (ASRM), NOR gate style memory and NAND gate style memory. EMIF16 is incompatible with several synchronous devices such as Dynamic Random-Access Memory (DRAM), Synchronous Dynamic Random-Access Memory (SDRAM) and Single Data Rate Synchronous Dynamic Random-Access Memory (SDR-SDRAM). This module available on Texas Instruments DSP chips will allow access to more high speed memory since the memory included with the processor is limited.

- Designed to interface with asynchronous memory devices
- Can access up to 256MB of memory
- Transmits and receives data using a 16 bit bus
- Supports Big and Little endian operation

3.2.8SRIO

Serial Rapid I/O (SRIO) is a module included on Texas Instruments DSP processors and other computing devices to facilitate non-proprietary and high-bandwidth system level interconnects. This interface provides gigabyte per second data transfer rates for use in chip to chip and board to board communication. SRIO is primarily used to connect microprocessors, large memory devices, memory mapped I/O devices and in networking equipment.

In today’s world SRIO is used primarily for large scale or high speed computer networks. The most recognizable of these is cellular infrastructure and hardware Ethernet networks. The number of devices that can be interconnected by SRIO is only limited by the number of address that are allowed by the routing infrastructure. This standard also allows numerous different types of devices to be interconnected without requiring special hardware for translation or conversion.

SRIO utilizes packet switching and a three-layer architecture commonly seen in networking. The Physical layer is the first of these layers and provides a hardware device level interface for error management, flow control, and electrical specifications. The transport layer is responsible for managing addressing schemes and routing information packets within the system. The logical layer is the final layer in this specification and provides all the necessary protocols and software interfaces required to ensure data reaches the proper endpoint intact.

Several advantages are provided by SRIO beyond high speed. It provides low power consumption, low latency and a low overall pin count all while keeping processor overhead to a minimum. SRIO is based upon serial communication where there are four physical wires, power, ground and a twisted pair of Tx, Rx lines. The latest specification for SRIO provides max bandwidths of 1.25, 2.5, 3.125 and 5 Gbps across each pair of I/O signals. Although due to encoding overhead the effective bandwidths are reduced to 1.0, 2.0, 2.5 and 4 Gbps. The bandwidths listed previously are for a single twisted pair of Tx, Rx wires, however the protocol does allow for up to four twisted pairs which can boost bandwidth to 16 Gbps.

Along with high data rates SRIO provides several features to take the burden off of the CPU. It has an integrated clock that it uses for processing and to communicate with other devices. Since the chip contains its own clock cycle it integrates hardware level error handling and Cyclic Redundancy Check (CRC) circuitry. Once the device has received the data intact it utilizes interrupts to return data for processing.

- Allows peer-to-peer communication
- Robust error-detection features built in
- Operations are built into the hardware instead of software
- Supports speeds from 1 to 4 GBPS
- Utilizes 4 pins for communication
- Lower power usage
- Low latency protocol

3.2.9McBSP

Multichannel Buffered Serial Port (McBSP) provides and interface between Texas Instruments DSP processors, as well as digital codecs and other devices contained in a system. McBSP is used primarily to provide an interface for audio devices with embedded processors. The audio formats that McBSP deals with are Audio Codec ’97 (AC’97) and Integrated Inter-chip Sound (IIS). According to the Texas Instrument documentation this port can be programmed for other serial formats but it is not recommended as this interface was designed specifically for audio.

McDSP includes several improvements over standard serial ports to make it ideal for audio input for processing. It allows direct access to numerous industry-standard codecs for encode or decode an audio signal as well as Analog Interface Chips (AIC's), Analog-to-Digital (A/D) and Digital-to-Analog (D/A) devices. The McBSP port also provides double buffered registers. This feature is critical for audio input as it allows us to read data from the buffer while new data is constantly being fed into them. This ensures that no part of the audio signal is lost due to a buffer being full or the processor being busy.

- Built in AC'97 and IIS audio codecs
- Dual input buffers allows data to be read while being received
- Can be programmed to support serial formats such as RS-232
- Clock is separate from CPU

3.2.10 GPIO

General Purpose Input/Output (GPIO) is a style of input and output commonly included in integrated circuits that facilitates user controlled communication via generic pins. By default GPIO pins are unused in most systems and are left available to system integrators to add additional features after production. These pins usually output voltage a 3.3 or 5 volts to enable or disable external devices, as well as having the ability to detect an input voltage and report it to the embedded processor. GPIO pins can be used for control of complex devices when paired with a PWM chip.

- Supports 3.3V and 5V operation
- Individual pins can be programed for input or output
- Can be pared with a PWM to facilitate communication

3.2.11 PWM

Pulse Width Modulation (PWM) or Pulse Duration Modulation (PDM) is a method of controlling the output of an electrical signal into a series of pulses. An easy way to visualize this is to think of PWM as Morse code. Mores code breaks information down into a series of dots and dashes or highs and lows. PWM accomplishes this by taking a standard 3.3 or 5 volt output and enabling or disabling it according to a specific frequency. The result is that devices such as GPIO pins can be utilized for complex input and output such as serial communication.

- Supports 3.3V and 5V operation
- Can be used to facilitate communication
- Useful for dimming LEDs and reading waveforms

3.3 Control Unit Printed Circuit Board (PCB)

For eGuitar, the brains of the system will reside at the base of the guitar. This will house the digital signal processing (DSP) chipset and LED microcontroller, as well as all I/O ports, which include the external LCD, LED fretboard matrix, power supply, and communication ports to the host PC. These will all be condensed into either one or two custom PCBs. In order to perform the design and manufacturing work, there needed to be layer considerations for the main control unit as well as a determination of software was to be used to create the PCB layout.

3.3.1Layer Consideration

A printed circuit board is made up of several different layers. These layers can include conductive layers, non-conductive substrate layers, and even silkscreen layers for clear labeling of the PCB parts. When it comes to PCB manufacturing, only the conductive layers are counted towards the layer count. So when a 2-layer PCB is created, it has two conductive layers as well as non-conductive layers in between as well as a silkscreen on top [1]. Typically, a simple circuit design only requires 2 or less layers, while complex circuits need more layers for all of the connections being made. Components are connected by copper tracks, which is a conductive path between two points. If two paths have to intersect, then another layer is used to send one path below the other. For the eGuitar's LED control circuit board, only 2-layers are needed since the circuit layout does not require any complex routing or intricate ICs.

3.3.2Design Software

3.3.2.1EagleCAD

EagleCAD is a PCB and schematic capture software provided by Cad Soft USA. A light version is available for free with given limitations. Those restrictions include only two layers of PCB can be used for design work, and the PCB designs can only have an area of 100 x 80 mm. For the purposes of the LED fretboard matrix microcontroller board, EagleCAD provides sufficient software. However, for the main control unit, which features the DSP chip, the light version of EagleCAD will not meet the design requirements. Many components being used for the eGuitar system are manufactured by Texas Instruments, who provide Eagle files for each of their products. Another benefit of EagleCAD is it's large community base, which can help ease the design process when complications arise.

3.3.2.2Altium Designer

Altium Designer is another PCB design and schematic capture tool offered by Altium. Their system includes schematic capture, 3D PCB layout, analysis, and programmable design. The software also allows rigid-flex board design, which adds flexible circuit design to their capabilities. Altium Designer adds convenient tools to boost the user experience. These include complete schematic-PCB design integrity, which is a system that automatically updates schematics when changes to the PCB design are made. Like EagleCAD, Altium also offers an intelligent guided interactive routing system, which keeps design requirements in check without accidentally routing two wires together. The only drawback to Altium Designer is that it requires a subscription service and does not offer a free option like EagleCAD's light version.

3.3.3Reference Designs and Layouts

The PCB designed for the eGuitar system will feature a microcontroller chip, I/O expander, input output connections, and a charging unit for the battery attachment. In this section, a number of schematics are shown as reference for the final PCB design.

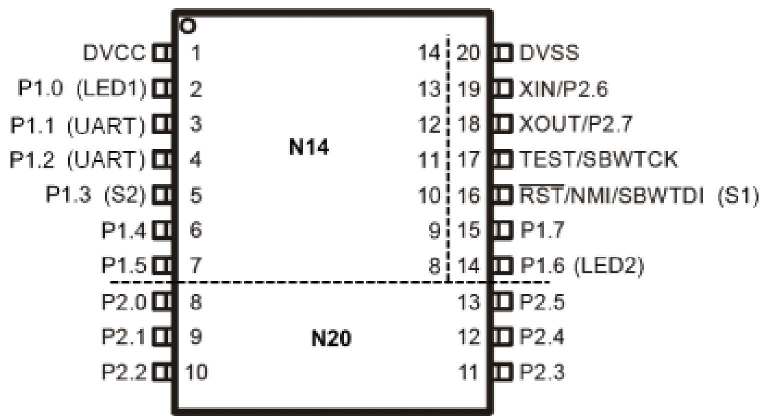


Figure 1 - MSP430G2553 N20 Pin Diagram
Reprinted with permission from Texas Instruments

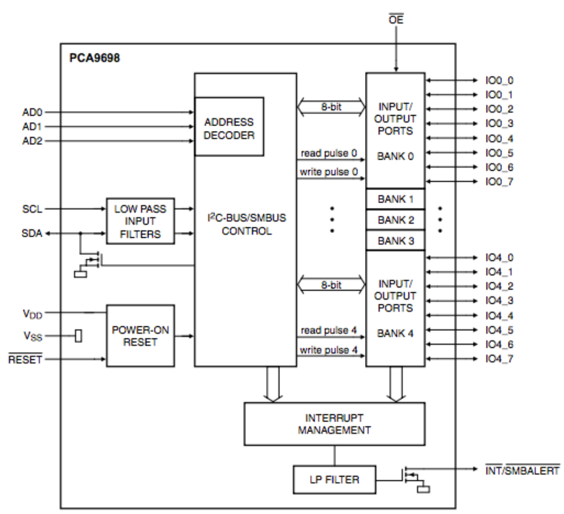


Figure 2 - I/O Expander Block Diagram
Reprinted with permission from NXP Semiconductors

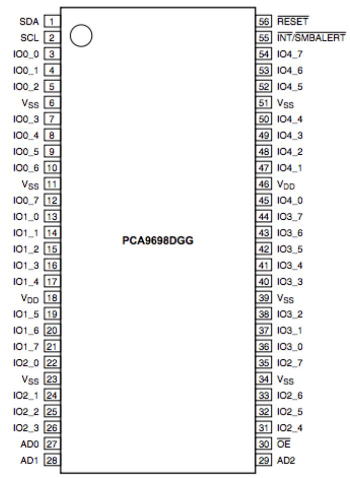


Figure 3 - I/O Expander Pin Diagram
Reprinted with permission from NXP Semiconductors

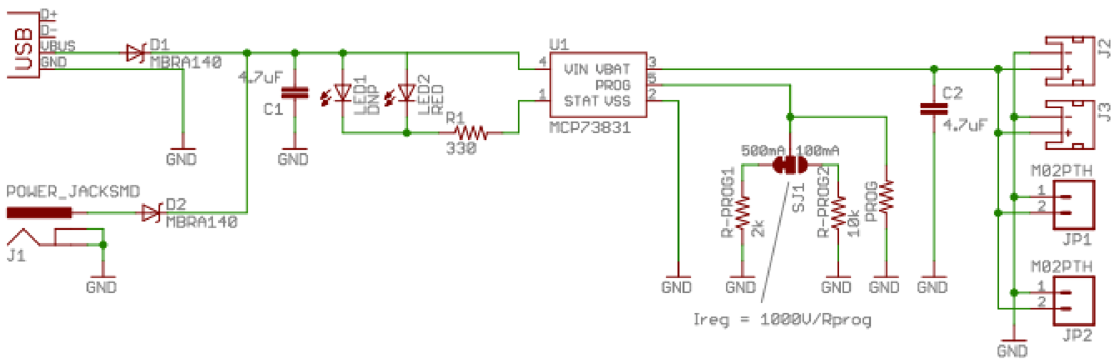


Figure 4 - USB Charger Schematic
Reprinted with permission from Sparkfun under the Creative Commons License

The PCB designed for the eGuitar LED fretboard matrix control is based off of the TI MSP430 LaunchPad. This board features the MSP430G2553N20 chip, which will meet the requirements of this project. The following are the schematics for the MSP430 LaunchPad, which is used as reference for the custom PCB of the eGuitar system.

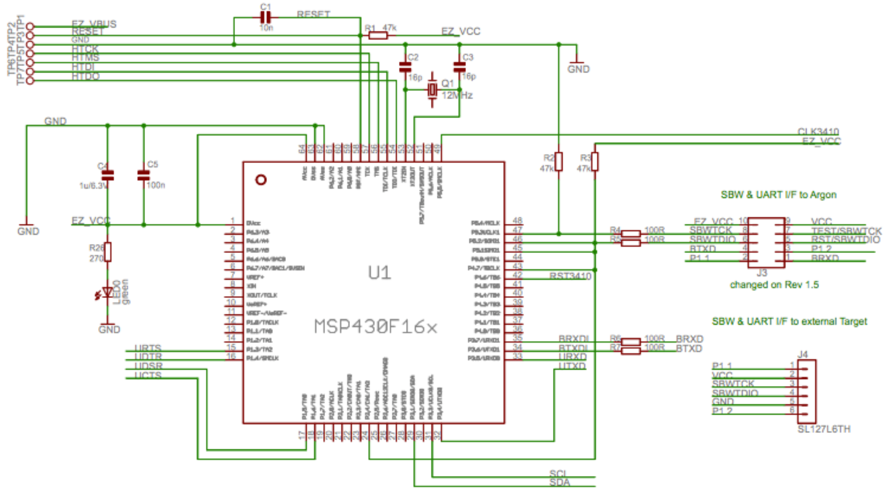


Figure 5 - MSP430 Emulator Schematic
Reprinted with permission from Texas Instruments

The main component in this schematic is the MSP430F16x. This processor is not the main microcontroller that a user programs, but is in charge of programming the MSP430G2553 through Texas Instrument's Spy-Bi-Wire Interface (SBW). SBW allows the use of only two connections for serial communication rather than four used in a JTAG interface. The unit at J3 takes messages from the F16x and sends them to the UART pins of the G2553 at P1.1 and P1.2, as well as sends VCC to power the G2553 chip. The SL1276TH component at the bottom right allows the LaunchPad to connect externally via pins.

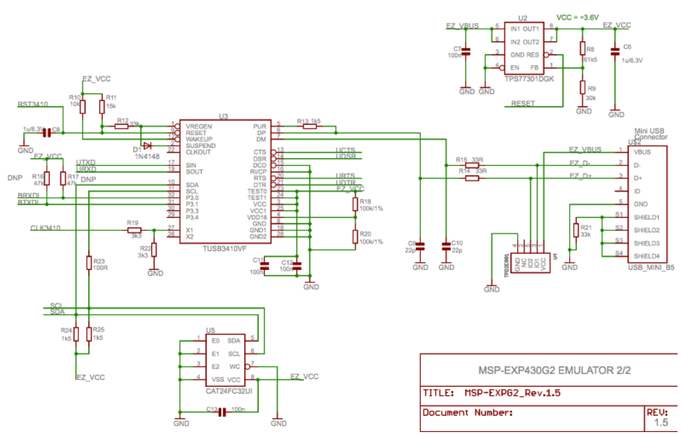


Figure 6 - MSP430 Emulator Schematic
Reprinted with permission from Texas Instruments

The figure on the right side of the schematic is the mini USB connector that the MSP430 supports. The input voltage from the mini USB is sent through the TPS77301DGK, which is a low dropout regulator (LDO). LDOs regulate output voltage from a higher input voltage. The TPS77301DGK can supply 250 mA of output current to the system and suppresses the input voltage to what the system needs (3.6V). The input voltage is also connected to a TPD2E001. This component is a two-channel transient voltage suppressor. The TPD2E001 dissipates electrostatic discharge (ESD) that might affect the board. These three components make up the power regulation section of the MSP430 LaunchPad. For the eGuitar system, power regulation is key to not damage the various subsystems. The TUSB3410 controller is featured on the left side of this schematic. It acts as a bridge between a USB port and a UART serial port. Basically, the TUSB3410 takes in the data sent from the host PC via USB and converts it to serial data. Data travels from the PC to the TUSB3410 using USB out commands. The data is then sent out of the TUSB3410 using the SOUT line to the MSP430. Lastly, the CAT24FC32UI component at the bottom of the schematic allows backchannel UART functions to the USB/UART bridge.

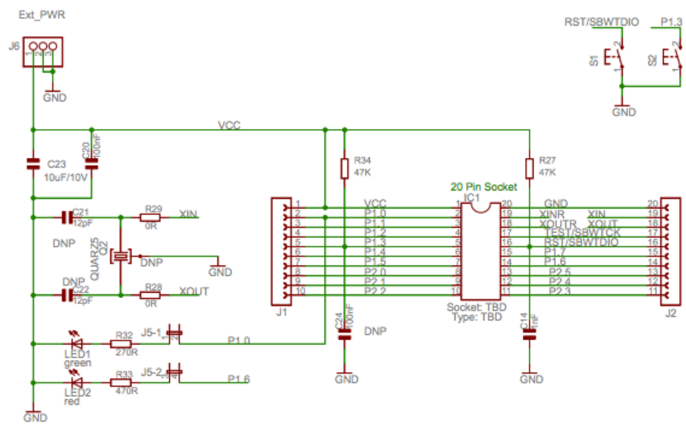


Figure 7 - MSP430G2553 Schematic
Reprinted with permission from Texas Instruments

The last schematic of the MSP430 LaunchPad features the socket connector for the G2553 chip. The connections to the socket lead to various parts of the LaunchPad, but ultimately, the G2553 can operate from an input voltage at VCC, a ground connection at GND, and programming signal at P1.1 and P1.2. The top left of the schematic shows that the LaunchPad could be powered externally rather than through the USB mini B port.

3.3.4Sourcing

Due to the requirements of both the main control DSP unit and LED fretboard matrix designs, the manufacturer of the PCBs must be discussed. The University of Central Florida has special promotions with 4 PCB manufacturers, and they recommend the Senior Design teams to contact these companies and use their services. These companies include PCB-Pool, OSH Park, Express PCB, and 4PCB. Below is a brief description of what each company offers.

PCB-Pool uses standard FR4 PCB material at either 1.6mm or 1.0mm in thickness. They offer circuit boards with up to 6 layers. They provide the most flexibility when it comes to options with the PCB. For instance, soldermasks, silkscreen, surface type, and even layout specifications are customizable. A standard 4-layer board runs about $125. One customization field that is desired is PCB material selection, but PCB-Pool does not offer anything other than FR4.

OSH Park offers high quality, lead free boards that are manufactured in the United States. They also provide free shipping anywhere in the US. 2 layer boards from OSH Park are $5 per square inch, and that includes 3 copies of the board ordered. OSH Park also has a 4-layer board option, which is only $10. All boards use FR4 PCB material and have ENIG (gold) finish, which allows easier soldering and resistance to the environment. The boards are 1.6mm thick. OSH Park provides an affordable and easy option for PCB manufacturing, but unfortunately, if the first design option for the LED fretboard matrix is chosen, then OSH Park does not have the correct materials for the project's needs. OSH Park would need to have a flex board material to satisfy the requirements.

Express PCB has the unique functionality of having their own PCB design software. Users can draw their schematics and design the PCB and immediately get a quote from the company. The design software allows 2 and 4 layer board designs. These board designs are placed in Express PCB's MiniBoard category, which is a 2 or 4 layer board with tin/lead plating and they must be within 3.8 x 2.5 inches. The MiniBoard is $51 for 3 2-layer boards or $98 for three 4-layer boards. These boards include silkscreen and solder mask layers, and take anywhere from 1 to 10 business days to manufacture. The thickness of the boards is not specified, but it looks as though they are made of FR4, which is typically around 1.6mm thick. Express PCB runs into the same problem as OSH Park, where they do not offer any flex board materials for manufacturing.

4PCB, also known as Advanced Circuits, has special student programs for their 2-layer and 4 layer PCBs. For the 2 layer full spec PCB, they offer one board for $33, while the 4-layer board is only $66. 4PCB also has its own design software, which can be used to easily send in PCB designs to the company. What is unique to 4PCB is the variety of PCB materials offered. They have the industry standard FR4 material as well as other RoHS compliant materials. One in particular is a polyimide material, which is commonly found in flex boards. This inclusion makes 4PCB a leading candidate for creating the PCBs needed for the LED fretboard matrix as well as the main control unit circuit board.

3.4 Light Emitting Diode (LED) Fretboard Matrix

The eGuitar system is comprised of a base control unit that processes user input and an LED Fretboard Matrix. The purpose of the LED Fretboard Matrix is to indicate to the user the notes that need to be played at the correct time. When a song is selected, the base control unit sends out signals to the LED matrix that correspond to the finger placement to play certain chords and notes on the guitar. The main goal of the LED system is to have a fluid user experience that lets the user easily play the guitar, regardless of skill level. There are two design ideas that will are researched in following sections. The first is LED placement underneath the strings of the guitar, while the second is LED placement on the side of the neck of the guitar. Both options were researched and designed, with one choice being selected for final implementation.

3.4.1Requirements and Specifications

A great user experience is the foundation for the requirements of this eGuitar system and more specifically, the LED system. The LED fretboard matrix must be easily visible to the user and unobtrusive while playing the guitar.

3.4.1.1Controller

The eGuitar system includes both digital signal processing needs as well as LED control. The DSP requirements are controlled a processor detailed in Section 3.6 - Digital Signal Processing (DSP). In order to not overload the processor that is computing the user output signals, a separate controller is used to command the LED fretboard matrix.

The controller must be able to command up to 36 LEDs at once. Typical microcontrollers, like Texas Instrument's MSP430, do not have enough general purpose input output (GPIO) pins to control as many LEDs as the matrix requires. However, a GPIO expander can be used to increase the amount of I/O ports available to the microcontroller. The LED controller will need to be able to send out signals using I^2C or SPI. More details can be found on the microcontroller and I/O expanders in Section 3.4.4 - Controller Hardware.

3.4.1.2Form Factor

This section is broken down into two parts for each form factor of the LED matrix. The first being considered is LEDs under the guitar strings, and the second being considered is the LEDs lying on the side of the neck.

For the idea of placing the LEDs underneath the guitar strings, the LEDs must be small enough to not interrupt the playing of the guitar. The way a note is typically played on a guitar is when a string contacts a fret on the neck and the user strikes that string to create a sound. There needs to be good contact between string and fret for the note produced to sound accurate. Therefore, if there is a strip of LEDs underneath the strings, they must not interfere with the contact between the string and corresponding fret. The typical height of a fret is around 1 mm, which means the height of the LED must at least be less than 1 mm. If this requirement is not met, the user experience can be compromised. The height requirement immediately restricts which types of LEDs the eGuitar system can implement.

Since there are 6 strings per fret, the LED matrix is comprised of 6 LEDs per fret. The LEDs will need to be wired together, but maintain the thinness requirement. Due to the specifications that are shown below in Table 1, the LEDs must either be wired together with very thin wire, or be mounted onto a printed circuit board (PCB) that meets the requirements. If a PCB were used, the only LEDs that can be implemented would be surface mount LEDs. PCB research can be found in Section 3.4.3.

Parameter	Specification
Height of LED	Less than 1 mm
Width of LED	Less than 3 mm
Combined resin enclosure and LED Height	Less than 1 mm
Height of PCB	Less than 1mm
Width of PCB	Less than 20 mm
Length of PCB	Less than 42 mm

Table 1 – Under-String LED and PCB Specifications

The second design idea is to place the LED matrix on the top of the neck of the guitar. The system must not interfere with the user's ability to play the guitar. Typically, guitar players hold the neck of the guitar in the palm of their hand with their fingers coming from beneath the guitar neck and wrapping around. Their thumb wraps up the neck and rests on the back or the top of the guitar neck. It is from this standard hand positioning that the eGuitar LED fretboard matrix can lie on the top of the guitar neck without major interference in how a player would typically use their guitar. Beginning guitarists look down on the frets as they play to maintain correct finger positioning. So with this second iteration, the LEDs would be arranged in a line that would be facing the user as they are playing. The guitarist using the eGuitar system would look down at their finger positioning and see the LEDs lit up that correspond to the fret and string that need to be pressed to play the correct note.

Since there needs to be 6 LEDs per fret, the width of the LEDs must be small enough to fit 6 within the length of a fret. Also, the frets on a guitar become smaller the further up the neck they go. This means that not only will the 6 LEDs have to fit into a space as long as the first fret, but also be able to fit on the 5th fret and so on. A PCB is manufactured for the strip of LEDs and an enclosure is created to house the PCB. The enclosure will serve two purposes for the fretboard matrix. The first is protection of the circuit board. Users will need to grip the guitar, which could damage an exposed circuit board. The second purpose is to clarify to the user which string and fret should be played. This is achieved by having the numbers 1 through 6 cut out at each fret. When the 5th string of the 3rd fret needs to be played, the corresponding LED will light up, which will illuminate the number 5 on the enclosure.

The size restrictions for this second idea are slightly more lenient as compared to the LEDs that would lie underneath the strings. The PCB that is on the neck of the guitar needs to have a height that is less than 4 mm. The width of the PCB can only be 20 mm at most, while the length of the PCB can be as long as 150 mm to cover up to 6 frets on the neck of the guitar. The enclosure must follow the same parameters as the PCB but be have enough height to fit on top of the circuit board, according the specifications below in Table 2.

Parameter	Specification
Height of LED	Less than 2 mm
Width of LED	Less than 3 mm
Number of LEDs	36
Height of PCB	Less than 4 mm
Width of PCB	Less than 20 mm
Length of PCB	At least 150 mm
Height of Enclosure	Less than 8 mm
Width of Enclosure	Less than 20 mm
Length of Enclosure	At least 150 mm

Table 2 - Side Neck LED and PCB Specifications

Both system ideas would be connected to the main control unit by a ribbon cable to prevent large bundles of wires. The thickness of the ribbon cable must be less than 5 mm, and the length of the cable must be at least 2 ft. long to connect all the way down the neck of the guitar to the base where the main control unit will lie.

3.4.1.3Color Support

Since the LED fretboard matrix is giving direct feedback and information to the user in real time, the correct use of color is essential. The colors used must be visible at a quick glance but also must not be distracting to the eye when playing. Also, as another requirement for the use of color, the LED must be visible even if the user is covering portions of the other LEDs when they are playing. In order for the user to play a song correctly, they must know what notes to play on time and what notes are played next. There should be a very small lag time between each note played, and the color of the LED can be used to assist the user in finding the correct notes quickly. Below is an explanation of color psychology and the perception’s people have for each color.

Blue has been noted as a mentally soothing color, which calms the mind and helps concentration. In an article written by Colour Affects in London states that, “[Blue] is the colour of clear communication.” They continue that blue objects typically do not appear to be as close physically as, say, red objects [2]. Playing the guitar can be a relaxing and rewarding activity, so having blue indicator lights on the LED fretboard matrix may maintain that feeling. However, blue is not the most eye-catching color, which could lead to players missing notes when they are using the eGuitar system.

Red is the color of power and objects appear closer when they are red. This means that red things tend to grab people's attention first when compared to other colors. However, red is not the most visible color. This color also stimulates and raises pulse rates in some cases. While blue is a more calming color, red can grab user's attention better. This would help reduce missed notes and also may alert players more when a note is hidden behind a finger.

The most visible and strongest color psychologically is yellow. Yellow has the potential to raise self-esteem as well as bringing confidence and optimism to a person. Alternatively, the wrong shade of yellow can bring about anxiety. While the visibility aspect of yellow is very appealing for the project, the dependency of tone can affect the eGuitar system in unintentional ways.

The last color that is focused on is green. Colour Affects continues in their article that, "[Green] is the colour of balance". Green is a restful color, but can also be portrayed as a boring or bland color. The positive aspects of green relate to guitar playing in the sense that it can bring peace and harmony to a person. This color can also be easy on the eye as compared to yellow.

With the ability to use multi-color LEDs for the eGuitar system, a combination of colors can be implemented for a better user experience. For example, the color red can be displayed when a note needs to be pressed and after the user successfully does so, the LED can then display the color green. This allows the user to easily judge whether they have their fingers in the correct position. Ultimately, the use of color can provide useful feedback to the player using the eGuitar system.

3.4.1.4Luminosity

Brightness of the LEDs in the lighting matrix is another parameter that must be considered for the eGuitar system. The brightness needs to be high enough that the user can easily view the correct LED, but not bright enough where the user gets eye strain from looking at the fretboard. LED visual output can be measured in terms of power and intensity. LED power is called luminous flux and is measured in lumen (lm), and LED intensity or brightness is called luminous intensity and is measured in candela (cd) [3]. The light output of an LED depends on many different variables including type of chip, wafer type, and enclosure type. Currently, there is no industry standard for LED brightness, but typically most manufactured LEDs do not have a high enough brightness to cause eyestrain. The intensity is dependent on the amount of current supplied to the LED, which means that if an LED is too bright, a restriction in current can help solve the problem [4].

3.4.2LED Comparison

Due to the rapid development and integration of LEDs in all areas of life, the typical LED is no longer just the indicator light for various electronics. The types of LEDs vary from car headlights to sports team scoreboards to home light bulbs.

For the eGuitar system, the focus of types of LEDs are on bi-color, RGB, and SMD LEDs.

The bi-color and RGB LEDs are the lights that can be typically found in a beginner's electronics kit. The bi-color LED contains two light-emitting dies within its enclosure, but only one color can be displayed at a time. RGB LEDs have the ability to display millions of colors by using different combinations of red, green, and blue colors [5]. Both bi-color and RGB LEDs have a typical height of around 5 mm.

SMD LEDs or surface mount device LEDs are usually components that are mounted to a circuit board. These types of LEDs are usually a single color, but there are bi-color and RGB variants for SMD LEDs. The size of these lights can vary from 0.5mm to 0.8mm in height, but their width and length are dependent on which type of LED it is. For example, a bi-color SMD LED manufactured by Kingbright has a length of 2mm and width of 1.25mm [6], while an RGB SMD LED made by Yetda Industry Ltd has a length and width around 5.4mm [7].

3.4.3Mounting Solutions

3.4.3.1PCB

The LED fretboard matrix requires a printed circuit board for both design ideas detailed in Section 3.4.1.2 - Form Factor. For the first design idea of placing strips of LEDs underneath the strings of the guitar, a circuit board would need to be less than 1 mm thick with its components mounted on it. For the second design idea of placing the LEDs on the top of the neck, the PCB does not need to adhere to the thickness restrictions of the first idea. Below is a detailed look at the types of PCB materials that can be used for circuit manufacturing.

The standard PCB material is FR4. This material is a glass fiber epoxy laminate and is a fire retardant material. Typically, FR4 is around 1.6mm thick and uses 8 layers of glass fiber material to form the board. It is capable of forming multi-layer boards, which can be used for direct chip attachments, automotive, and wireless communications. FR4 is noted for its consistency and predictability when being fabricated. Ultimately, this material is easy to use and is a reliable option for circuit design.

PCB materials also include Flex and Rigid-Flex applications. These options are circuit boards that are non-rigid, meaning they can fold and wrap around areas that most circuit boards cannot reach. The Rigid-Flex options add some sturdiness in the flex material so the material cannot break as easily [8]. The material used is called Kapton, which is manufactured by Dupont Corporation. The Kapton materials are in the Pyralux product line for Dupont. These flex materials are acrylic-based copper clad laminates. They offer high bond strength as well as high thermal resistance. The thickness of the material can range from 0.01mm in single layer variants to 0.13mm with multi-layer variants. Other PCB fabricators offer a polyimide material, which has the same properties as Dupont's Pyralux products [8].

While the folding and bending properties of the flex materials will not be needed for the eGuitar LED matrix, the thinness is highly desired for the first design idea. With this material, it is possible for the circuit board to easily fit underneath the strings of the guitar without interference. If the FR4 material was used, it would not meet the specifications required to fit underneath the strings, however FR4 may be the right option for the second design idea. By placing the LED strip on the top of the neck of the guitar, the thickness requirement is not as strict. FR4 could be a solid option as it can be easily produced and is a reliable option for this project.

3.4.3.2Custom Housing

The second design idea for the LED fretboard matrix has a strip of LEDs positioned at the top of the neck. This design requires a custom housing which will protect the PCB that contains the LED strip. There are many available options for manufacturing the enclosure. The most easily accessible and affordable would be the route of 3D printing. 3D printing heats and extrudes a plastic that forms a CAD designed object. There are two prominent materials used for this process and is detailed below.

ABS (Acrylonitrile Butadiene Styrene) and PLA (Polylactic Acid) are thermoplastics that 3D printers use to create custom objects. Alex English from ProtoParadigm gives a good description of what the requirements are for materials that are used for 3D printing. He details three tests that need to be passed for each material considered. The first being the ability to form into a plastic filament, which is the form factor used in the printers. The second test is that the materials look nice when printed as well as assure physical accuracy. The last requirement for 3D printed materials is that they can offer the desired property for each application, meaning the correct strength, gloss, durability, and many other qualities. Both ABS and PLA pass the three tests detailed by English [9]. ABS offers high strength, flexibility, and machinability. However, ABS requires a heated print bed to reduce curling on the base edges of the object being printed, which some 3D printers do not offer. PLA has a better feel than ABS printed objects and can also offer a range of colors and glossy finishes that ABS cannot have. Table 3 below details the differences between ABS and PLA plastic.

ABS	PLA
Extrude at ~225 DEGREE C	Extrude at ~180-200 DEGREE C
Requires heated bed	Benefits from heated bed
Works reasonably well without cooling	Benefits greatly from cooling while printing
Adheres best to polyimide tape	Adheres well to a variety of surfaces

Filament tolerances are usually tighter	Finer feature detail possible on a well calibrated machine
Prone to cracking, delamination, and warping	Prone to curling of corners and overhangs
More flexible	More brittle
Can be bonded using adhesives or solvents (Acetone or MEK)	Can be bonded using adhesives
Fumes are unpleasant in enclosed areas	More pleasant smell when extruded
Oil Based	Plant Based

Table 3 – ABS vs. PLA

For the eGuitar enclosure, the plastic used must be able to withstand the constant interaction with the user's hand, which will come in contact with the enclosure at almost all times. Both plastics would work well for the eGuitar's enclosure needs, but the detail that ABS plastics are more prone to cracking and delamination makes it a less desirable material. Also, the PLA material is noted as feeling better to the touch, which would also remove any possible distractions with the feel of the enclosure.

3.4.4 Controller Hardware

3.4.4.1 Microcontrollers

For the purposes of the LED fretboard matrix, the microcontroller, which is the serial master for 36 LEDs, does not need to be complex. Based on the requirements specified above, the microcontroller only needs to be able to communicate using I^2C or SPI to an I/O Expander, which will control the LEDs individually.

The ATmega328P is a low-power CMOS 8-bit microcontroller [10]. With a 2-wire Serial Interface and an SPI port, the ATmega328P has the necessary functions for LED control. Its development tools are user friendly, and the microcontroller does have large community support.

Another option for the LED controller is Texas Instrument's MSP430 microcontroller. Like the ATmega328P, the MSP430 also offers synchronous SPI and I^2C communications, which is contained in their Universal Serial Communication Interface (USCI). The MSP430 offers a low-power microcontroller solution, which meets the requirements for the LED matrix controller. Details of each microcontroller can be found in Table 4 – Microcontroller Comparison.

	ATmega328P	**MSP430**
Voltage Range	1.8V to 5.5V	1.8V to 3.6V

Architecture	8-Bit RISC	16-Bit RISC
Frequencies	20 MHz	16 MHz Internal Frequency
Communications	PWM Serial USART SPI Serial Interface 2-wire Serial (I^2C)	USCI: UART IrDA Encoder/Decoder Synchronous SPI I^2C
Memory	32kb flash memory 1kb EEPROM 2kb Internal SRAM	32kb flash memory 1kb RAM

Table 4 – Microcontroller Comparison

3.4.4.2I/O Expander

For the LED fretboard matrix, there are at least 36 LEDs that need to be individually controlled. LED control does not need a high-powered processor, but just a large amount of I/O pins. From the two choices of microcontrollers above, neither offers more than 20 GPIO pins, which means an I/O expander must be implemented for the LED fretboard matrix control. There are two options for general I/O expansion, shift registers and I/O expanders.

A shift register is a form of sequential logic. Meaning, it is a combination of flip-flops, which share the same clock. Shift registers can take in signals and “shift” the data in the array down the register. This means that if an 8-Bit shift register is used for I/O expansion, the microcontroller would only have to send out 1 signal to the shift register and it has the potential to control 8 different I/O ports [11]. If more I/O is needed, then shift registers can be daisy chained together, or linked together serially. Ultimately, only 3 lines are needed from the microcontroller to command over 64 I/O pins [12]. An example of a shift register is Fairchild Semiconductor’s MM74HC595 8-Bit Shift Register with Output Latches [13]. This component has an 8-bit serial-in, parallel-out, shift register. It operates from 2V to 6V and can be cascaded with other shift registers.

An I/O expander acts in a similar way of shift registers, but they communicate to the microcontroller differently as well as allow more I/O expansion per component. An example of an I/O expander is the PCA9698 by NXP Semiconductors. This specific expander offers 40-bit parallel I/O with I^2C. Any of the 40 I/O ports can be configured as an input or output. The PCA9698 requires 5V supply voltage with a total load of 1A, which is able to drive 40 LEDs at once. I/O expander’s only use 3 address pins from the microcontroller, and in the case of the PCA9698, can have 64 programmable slave addresses. A more complicated I/O expander can be found in Cypress Perform’s CY8C95xxA family of I/O expanders. This expander is the same as the PCA9698, but also has internal EEPROM and PWM functionality. For the purpose of the eGuitar, these additional functions are not needed.

3.5 User Interface

The group values an intuitive, functional user interface (UI). Generally speaking, the user interface is the mode by which a user can interact with the system. The eGuitar project, the group researched user interface for both the embedded (embedded) components and the personal computer.

3.5.1 Embedded UI Hardware

The embedded portion of the eGuitar system requires an array of hardware controls for interfacing with the user. A relatively compact but functional set of physical controls will manipulate and navigate through various logical states, indicated by the display, the fretboard led matrix, or audible feedback.

3.5.1.1 Display

In researching small displays for electronics projects, two display technologies stand out as affordable options: Organic Light Emitting Diode (OLED), and Liquid Crystal Display (LCD). For the purposes of the eGuitar system, the embedded display was chosen primarily on the basis of its affordability and readability. In choosing between LCD and OLED screens, an understanding of the underlying technology is important. LCD technology involves small crystals embedded into the display panel. A backside light source shines through the screen, allowing the crystals to alter the color of the light. OLED screen technology has small, self-illuminating, color-producing elements embedded in the screen. With no need for a backlight, OLED screens have more specific control over each pixel's luminosity.

Picture quality is one of the defining issues when considering OLED vs. LCD technology. LCDs offer vivid colors that can be optically discerned even in brightly lit environments, but the backlight often washes out the portions of the screen that are supposed to be black. OLEDs exhibit vibrant colors and, because they can turn off the components of the screen that need to be black, those blacks are deeper than those of their LCD counterparts. This is consequential not only for displaying true blacks, but additionally because colors appear sharper against darker backgrounds. There is a stark difference in the quality of viewing angles between these two technologies. LCDs struggle with the loss of picture quality when viewed at an angle. OLED screens do not suffer any loss of picture quality at extreme angles.

Power consumption is final major area of contrast with OLED vs. LCD. LCDs generally consume more power than OLEDs due to their always-on backlight. OLED technology only uses electricity for the parts of the screen that need color; parts of the screen that remain black use no power, making OLEDs significantly more energy-efficient. Considering that the embedded eGuitar system is battery-powered, power consumption is of great concern. In this regard and by consideration of all of the previously mentioned differences between LCD and

OLED screens, the eGuitar system will likely opt for OLED technology for its embedded display.

3.5.1.2Fretboard LED Matrix

The eGuitar system will equip the chosen guitar with an LED matrix for attachment to either the fretboard either under the strings or on the side of the neck. Pending hardware constraints, the goal of the LED matrix is to provide the user with a visual feedback of where their fingers should be places to play the set of notes dictated by the current point in the tablature being played. As it is so integral of a component of the eGuitar system's user interface experience, the fretboard LED matrix research is covered separately in Section 3.4 – LED Fretboard Matrix.

3.5.1.3User Controls

In using the eGuitar embedded system, the user will require a means of interacting with the embedded logic. There exists a wide selection of hardware for user input such as potentiometers, buttons, switches, sliders, etc. For the eGuitar system specifically, a subset of these are necessary for user interaction.

3.5.1.3.1 Potentiometers

A potentiometer's purpose is simple—to convert physical user input into varying resistance. As such, a potentiometer is an electro-mechanical transducer. By converting a change in linear or rotary position into a change of resistance, potentiometers are useful for a variety of applications where the user needs to adjust a parameter on a scale. Potentiometers vary greatly in their specifications and usage, but the most important variables with respect to the eGuitar system are a potentiometer's total resistance and taper.

Potentiometers are rated by their total resistance, or the resistance when the potentiometer is at the high extreme of its linear or rotary position. This rating makes for an effective change in sensitivity while turning or sliding the potentiometer. For the purposes of the eGuitar system, potentiometer ratings are chosen on the basis of their ability to handle operating voltages appropriately.

A potentiometer's taper describes the function that governs the amount of varying resistance with linear or rotary movement. While the potentiometer might rotate or translate in a smooth or linear fashion, the taper determines the output resistance value. In the case of linear potentiometers, the change in output resistance is directly proportional to the change in movement of the potentiometer mechanism. In the case of other tapers like logarithmic or antilogarithmic, the resistance varies with linear change in a fashion that approximates the taper name. This is very practical for circumstances like audio volume control; because human perception of loudness is on the logarithmic decibel scale, a volume control potentiometer should have a matching appropriate logarithmic taper so that a linear change in potentiometer position or rotation results in the perception of matching linear loudness. For the purposes of the eGuitar system, both logarithmic and linear taper potentiometers are used--for volume controls and standard controls, respectively.

3.5.1.3.2 Switches

Switches are perhaps the most basic hardware-level user interaction used in the eGuitar system. Switches vary in form factor and aesthetics, but all serve the same purpose of selectively allowing voltage and current to pass through them by means of an open or closed circuit. Switches work by selectively allowing voltage to pass through them. In the open (off) state, the switch physically disconnects its two pass-through pins. In the closed (on) state, the switch will complete the circuit by physically connecting its two pass-through pins. This allows current and voltage to pass through. Due to their simplicity, hardware switches are selected for the eGuitar system only on the basis of their form factor. They are used for toggling power and possibly other binary user choices.

3.5.1.3.3 Pushbuttons

Pushbuttons are a special-purpose switches that are mechanically stable in only one position. Buttons are available in varying shapes and sizes, but all serve the same purpose of reporting a user's press to the software system. Like switches, buttons work by selectively allowing voltage to pass through them. Depending on the button design, the reading from the button is either high or low when not pressed but will reverse its state when depressed, by opening or closing a circuit.

When using buttons, it is important to reduce noise in-- or "debounce" --the input. Without doing so, a low quality of either the user's press or the button itself could result in a single press being interpreted as a series of presses. Noise filtering can be achieved at either the hardware level by use of resistors/capacitors, or at the software level by treating button press events within a small time frame as only one press. For the purposes of the eGuitar system, buttons are chosen on the basis of their form factor and will likely be used for simple user confirmation actions.

3.5.1.3.4 Rotary Encoders

Like potentiometers, rotary encoders are electro-mechanical transducers. Unlike potentiometers, they support unlimited rotation making them useful for applications where settings are shared across a single dial. Rotary encoders are more complex than potentiometers in that they convert their mechanical position into analog or digital code, usually binary or Gray code. The two main types of rotary encoders are absolute and incremental (relative). Absolute encoders report the current position of the shaft while relative encoders report information about the motion of the shaft. The eGuitar system aims for an intuitive and minimal-hardware embedded control system, for which a rotary encoder is invaluable. User control and interaction will largely occur via a single rotary encoder with a built-in button. Much like familiar car stereo controls, an infinite-rotation knob supporting presses will make for quick and effective menu and information navigation.

3.5.1.3.4.1 Gray Code

Often used by rotary encoders to report absolute shaft angle or relative shaft motion, gray code is a form of binary where increasing values differ in only one bit. The advantage with Gray code over standard binary number representation is that it provides the ability to check errors. The bit total of a new encoder position should differ by only one compared to the previous position's bit total. For more robust error checking, it is also the case that the bit total should alternate between even and odd upon change.

If we analyze a binary rotary encoder with 3 contacts and 8 sectors, its sector position can be reported in binary as seen below in Table 5 - Binary Rotary Encoder States:

Sector	Angle	Contact 1	Contact 2	Contact 3	Binary
0	0° to 45°	off	off	off	000
1	45° to 90°	off	off	on	001
2	90° to 135°	off	on	off	010
3	135° to 180°	off	on	on	011
4	180° to 225°	on	off	off	100
5	225° to 270°	on	off	on	101
6	270° to 315°	on	on	off	110
7	315° to 360°	on	on	on	111

Table 5 - Binary Rotary Encoder States

An equivalent Gray code rotary encoder with 3 contacts and 8 sectors can report its sector position in Gray code as seen below in Table 6 - Gray Code Rotary Encoder States:

Sector	Angle	Contact 1	Contact 2	Contact 3	Gray Code
0	0° to 45°	off	off	off	000
1	45° to 90°	off	off	on	001
2	90° to 135°	off	on	on	011
3	135° to 180°	off	on	off	010
4	180° to 225°	on	on	off	110
5	225° to 270°	on	on	on	111
6	270° to 315°	on	off	on	101
7	315° to 360°	on	off	off	100

Table 6 - Gray Code Rotary Encoder States

3.5.1.4Audio Output

In providing a means for the user of the eGuitar system to receive audible feedback, many viable options exist. Ideally the user should be able to use their own headphones as well as take advantage of an embedded speaker for tasks like reference tones for tuning, metronome clicks, etc. It is standard in the guitar industry to support only mono audio, as the vast majority of guitars themselves output mono. While the eGuitar system will ideally handle 6 or more channels for processing (one channel per string), audio feedback need only be mono.

3.5.1.4.1 Headphone Jack

A headphone jack is an important inclusion for the various audio features of the embedded eGuitar system. In music and audio production the most common form factor for headphone jacks is 1/4 inch (6.35 mm) class, but for other (standard) applications, 1/8 inch (3.5 mm) class audio jacks and plugs prevail. As the eGuitar system is innately a music and audio production utility, it stands to reason that the 1/4 inch (6.35 mm) class audio hardware should be supported. For users of the eGuitar system with headphones or audio equipment of the 1/8 inch (3.5 mm) class, there exist cheap and readily available 1/8 inch (3.5 mm) class to 1/4 inch (6.35 mm) class adapters (pictured below) that will patch audio (in this case mono) without issue.

3.5.1.4.2 Embedded speaker

If the user opts not to use the headphone jack, audible reference tones and metronome clicks are rerouted to the embedded speaker. While the embedded speaker does not need to be high-fidelity, it does need to be fairly loud and support the reference frequencies for a reasonable set 6-string guitar tunings. The standard guitar string frequencies that the chosen speaker must support are listed below in Table 7 - Standard Tuning Frequencies. Affordable and readily available speakers of small form factors usually support frequencies from 0 – 20 kHz, which well exceeds the minimum response range for the purposes of reference notes for tuning.

String (Pitch)	Frequency (Hz)
1 (E4)	329.63
2 (B3)	246.94
3 (G3)	196.00
4 (D3)	146.83
5 (A2)	110.00
6 (E2)	82.41

Table 7 - Standard Tuning Frequencies

3.5.2Embedded UI Software

The eGuitar system's embedded software for user input will involve controlling the display based on hardware controls. Reading and storing button, potentiometer, switch, and encoder states will allow for logical control over the feedback that the user receives. From the UI hardware research it is likely that a rotary encoder with shaft button is used for the primary user navigation. The language used to develop the UI software is dictated by the chosen embedded processors. The development of the UI software will occur on either the MSP430 or the TMDX5535EZDSP chip using Code Composer Studio (CCS) from Texas Instruments.

3.5.3PC UI Hardware

The eGuitar system's PC hardware for user input will adhere to the classical input methods of keyboard and mouse for tablature editing. In the case of tablature recording, the user's guitar rigged with the embedded eGuitar system is the user's method of input. The user's playing of notes are transcoded into digital guitar tablature in real-time.

3.5.4PC UI Software

As pitch detection is handled entirely by the embedded DSP chip, the PC portion of the eGuitar system merely exists as a user interface for recording and editing tablature. As such, the research regarding PC UI software consists of a comparison of the UI capabilities of various coding languages discussed here. Due to time constraints and a reasonable amount of past exposure to PC languages for UI development, only languages with which the project members have previous experience are considered for development of the eGuitar system's PC software.

3.5.4.1Java

As a language for PC user interface development, java has several advantages. Java runs on the JVM, a virtual machine that has been ported to most major platforms. Code written in java will need little if any changes across platforms to support serial communication from the embedded hardware to the PC. Java also has an expansive following and a multitude of public code samples, libraries and cross-compatible APIs and toolkits like Abstract Window Toolkit (AWT) and Swing. With drag-and-drop UI designers like the one included with the NetBeans Integrated Development Environment (IDE), the overhead of making an aesthetically appealing window of user controls is drastically reduced.

3.5.4.2C#

C# has important considerations compared against Java as a language for the eGuitar system's PC user interface. Compatibility with the Visual Studio IDE is arguably the greatest advantage-- Visual Studio is a very powerful and mature

IDE with nearly endless features. While it is not as involved as the NetBeans' drag-and-drop system for Java UI development, Visual Studio does have a drag-and-drop toolkit for the basic setup of either a Windows Forms or Windows Presentation Foundation (WPF) UI. It is also worth considering that a C# application would only support a Windows operating system, thus excluding Mac and Linux users.

3.5.4.3C++

C++ has largely the same disadvantages and advantages as C# when compared against Java. Additionally, C++ has the distinct advantage of being the language in which the partially open-source PowerTab system was written. PowerTab supports ASCII tablature exports and provides its code for its tablature parsing system. As its primary disadvantage over the other languages, C++ is a lower-level than C# or Java, generally considered to be more tedious to develop with.

3.6 Digital Signal Processing (DSP)

The portions of the eGuitar system that require information about the user's playing the guitar itself require Digital Signal Processing (DSP). In general, the project will require pitch detection (ideally from multiple channels) and pitch generation, both of which should be handled by a special-purpose DSP chip.

3.6.1Requirements and Specifications

This section addresses the requirements and specifications for the performance of the eGuitar system. The DSP chip must perform responsively when tuning or potentially recording tablature from user playing. The DSP chip must support polyphony and utilize an effective audio sampling rate and audio word size.

3.6.1.1Polyphony

Polyphony simply means that multiple pitches can be played (or in this case detected) simultaneously. If the eGuitar is to be able to go beyond single-note detection for applications like tuning, the DSP chip must be able to handle multiple inputs simultaneously. Whether the DSP chip supports multiple inputs by design or if a main input line must be switched between string outputs rapidly, ultimately the DSP chip is splitting its pitch detection time between 6 channels by time division. This means that for a 6 string guitar, the response time of a single-channel DSP chip is effectively reduced to 1/6 of its original speed.

3.6.1.2Sample Rate

For digital audio processing, values are continuously sampled from a single analog input and are stored at certain intervals based of the sampling frequency of the system. If the DSP chip is time dividing, a shared sampling frequency must be accounted for. To reliably interpret an analog signal digitally, the analog signal must be sampled at twice its maximum frequency by the Nyquist sampling theorem.

Assuming a standard tuning 6 string guitar with 24 frets (two full octaves), the highest standard fret pitch is 2 octaves above the highest reference note E4. If the highest standard fret pitch is an E6 and the user adds an octave by playing a pinch harmonic, the user will have achieved a maximum pitch of E7, or 2637.02 Hz. To finish out the octave and support some non-standard tunings, the eGuitar system should support a maximum achievable frequency of C8, or 4186.01 Hz. By the Nyquist sampling theorem, the required detection frequency is 2*4186.01 Hz = 8372.02 Hz. For reference, human hearing ranges from approximately 20 Hz to 20 kHz. As such, common audio sampling rates are about twice that, at 44.1 kHz.

In theory, time division between sampled channels should not affect required sampling rate. Splitting processing time on the DSP chip between channels results in an effective pitch detection refresh rate of 1/6 the chip's clock rate. As the DSP chips researched approach 100 MHz clock speeds, audio processing speed should not be an issue.

3.6.1.3Audio Word Size

In audio processing, the audio word size (number of bits that represents a single audio wave) limits the quality of the signal being sampled. This also directly governs the output signal-to-noise ratio. With the most commonly used audio word size of 16 bits. The DSP implementation for the eGuitar system will attempt to adhere to at least CD quality audio word size, which allows for a signal-to-noise ratio of approximately 96 decibels.

3.6.2Programming

The development of the digital signal processing applications for the TMDX5535EZDSP chip will take place on a PC using Code Composer Studio (CCS) from Texas Instruments for compilation and debugging purposes. It is important to consider publicly available and reusable software and theory for pitch detection purposes.

3.6.2.1Available Software

Texas Instruments provides the TMS320C55x Connected Audio Framework -- a software framework which will allow the DSP chip to operate as a USB Audio peripheral with audio processing during record or playback. Texas Instruments also provides DSPLIB, a collection of common DPS functions that will help to implement the various audio features of the eGuitar system. DSPLIB is marketed as being highly optimized and supporting fixed and floating point computations. The source code for these functions is provided and functionality includes adaptive filtering, correlation, fast Fourier transform, filtering, convolution, and matrix computations.

3.6.3 Pitch Detection Theory

Much of the theory regarding audio DSP is readily available online. The eGuitar system primarily requires pitch detection capabilities for both the chromatic tuner and the tablature recording features. A pitch detection algorithm serves the purpose of estimating the pitch or fundamental frequency of a periodic signal. Existing methods of pitch detection algorithms utilize the time domain, the frequency domain, or both.

3.6.3.1 Time-Domain Pitch Detection

The general goal of a time-domain pitch detection algorithm is to estimate the period of the input signal and convert that period to a frequency, yielding pitch. Some common time-domain pitch detection algorithms are discussed below.

3.6.3.1.1 Zero-Crossing Rate

One of the more simplistic approaches to time-domain pitch detection measures the *zero-crossing rate*. This is the time between points of the input signal that cross the y = 0 line. With a sufficiently high sampling rate, zero-crossings are reliably measured when the value of the waveform changes its sign (from negative to positive or from positive to negative). To understand the determination of frequency from zero-crossings, observe below Figure 8 - Sinusoidal Frequencies of 1 and 2 Hz.

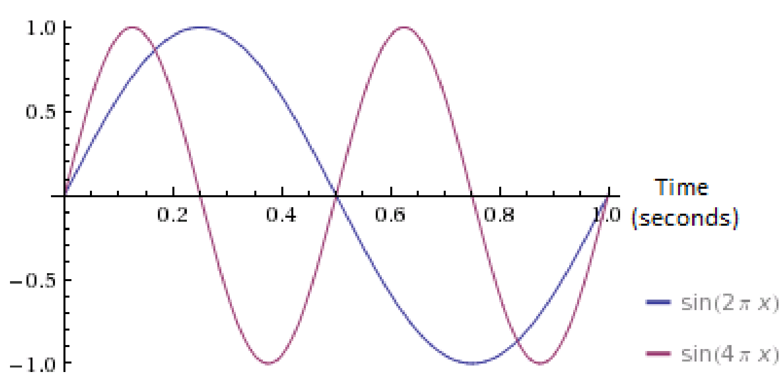


Figure 8 - Sinusoidal Frequencies of 1 and 2 Hz.
Reprinted with permission from Wolfram Alpha [18]

Using an analysis interval of 1 second, the blue waveform, crosses *y = 0* three times: at times 0, 0.5, and 1 seconds. Over the same analysis interval, the purple crosses *y = 0* five times: at times 0, 0.25, 0.5, 0.75, and 1 seconds. For the blue waveform, a zero-crossing every 0.5 seconds yields a cycle period of 1 second by nature of a sinusoid. For the purple waveform, a zero-crossing every 0.25 seconds yields a cycle period of 0.5 seconds. Given the period, *T*, of each waveform, the frequency, *F*, can be calculated from the reciprocal by the fundamental Frequency-Period equation: $F = \frac{1}{T}$

Thus, the frequency of the blue and purple waveforms are 1 and 2 Hz, respectively. The advantage of the zero-crossing rate pitch detection algorithm is its conceptual and computational simplicity. Unfortunately, this algorithm is inadequate in handling complex input signals with noise or multiple base waves.

3.6.3.1.2 Autocorrelation

More sophisticated and capable time-domain pitch detection algorithms use autocorrelation, in which a time interval of the input signal is correlated against itself at a different time. Classic examples of autocorrelation algorithms include Average Magnitude Difference Function (AMDF) and Average Squared Mean Difference Function (ASMDF). Both of these algorithms provide accurate results for highly periodic signals, but can have octave issues where the detected pitch might be the correct note, but the incorrect octave. Modern and commonly used autocorrelation algorithms build upon AMDF and ASMDF, aiming to reduce error and in some cases support even support polyphony. As these algorithms are very complex and computationally expensive, the eGuitar system will attempt to rely on individual signals from each string via a custom pickup.

3.6.3.2Frequency-Domain Pitch Detection

The general goal of a frequency-domain pitch detection algorithm is to estimate the spectral density of the input signal and convert that to a frequency, yielding pitch. Frequency-domain pitch detection makes possible polyphonic note detection at relatively large computational cost. Here many common frequency-domain pitch detection algorithms are discussed and compared, the basis for most being the fast Fourier transform (FFT) algorithm.

3.6.3.2.1 Fast Fourier Transform

The fast Fourier transform (FFT) algorithm aims to compute the discrete Fourier transform (DFT) and the inverse as efficiently as possible. FFT is a highly optimized form of Fourier analysis-- the study of the way in which complex functions can be approximated by sums of simpler trigonometric functions. FFT rapidly performs converts between time and frequency through a specific and efficient process; it computes its transformations by decomposing the discrete matrix into a manageable number of simple terms.

In understanding the speed gain off FFT over the standard DFT algorithm, consider a set of N inputs. A direct evaluation of the DFT requires as many as N^2 operations. The FFT algorithm has the quality of being able to produce the exact same output as the standard DFT, with only (N * log (N)) operations. Observe the extreme disparity in required operations across merely 25 input terms below in Figure 9 - Operational Complexity of Standard DFT and FFT.

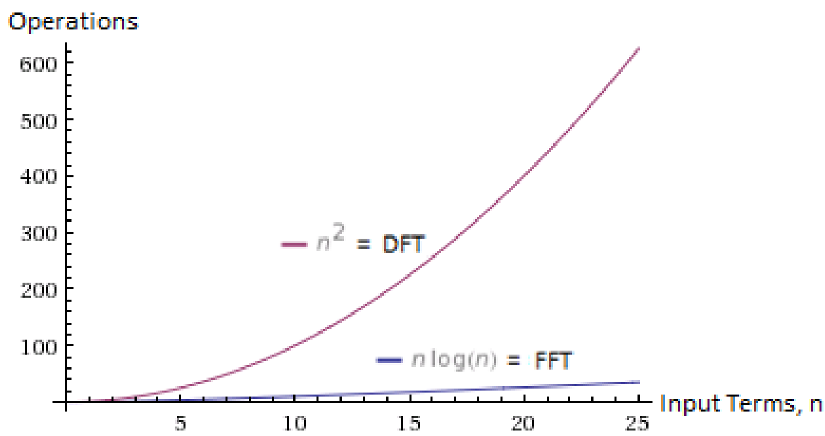


Figure 9 - Operational Complexity of Standard DFT and FFT
Reprinted with permission from Wolfram Alpha [19]

At higher input *N*, the contrast between FFT and standard DFT is truly staggering. Observe that with increasing input size, the relative speed of the FFT algorithm increases over the standard DFT algorithm below in Table 8 - FFT vs. Standard DFT Performance.

Input size	FFT operations	Standard DFT operations	FFT relative speed
N = 10	10	100	10 times faster
N = 100	200	10,000	50 times faster
N = 1000	3,000	1,000,000	333 times faster

Table 8 - FFT vs. Standard DFT Performance

The FFT algorithm is the basis for many of the modern frequency-domain pitch detection approaches considered here. Without it, DFT reliant frequency analysis would be nonviable on an affordable embedded digital signal processor.

3.6.3.2.2 Periodogram

The periodogram is a way to estimate the spectral density of an input signal. The periodogram of an input signal is often computed from a finite series of terms using FFT. The raw periodogram suffers from two serious but solvable issues: spectral bias and frequency variance. Spectral bias is caused by the sharp convergence of the underlying sequence and ultimately reduces the periodogram accuracy. This can be combatted by applying special functions to spread out the rate at which the underlying sequence truncates. The frequency variance issue is caused by the fact that the variance at a given frequency does not decrease with as the number of signal samples increases. In combatting frequency variance, one constructs a spectral plot-- is a smoothed periodogram whose signal noise has been reduced by techniques of varying complexity. One common method for constructing the spectral plot is Bartlett's method, which attempts to average several periodograms by dividing the samples into smaller groups of equal size. By computing the squared DFT of each group and averaging all of the groups, the standard deviation of the periodogram is reduced by $1/\sqrt{G}$, where G is the number of smaller groups.

3.6.3.2.3 Maximum Likelihood

A popular but specific-case approach to frequency-domain pitch detection involves the use of maximum likelihood. This algorithm attempts to match the frequency domain characteristics against predetermined frequency maps. This is perhaps the most accurate approach when solving for well-understood pitches like that of a musical instrument, but innately requires reference data. As such, this approach is suitable only when pre-recorded, in-tune notes are available for analysis and the computer performing the pitch detection algorithm can reference resultant maps from this predetermined analysis. While the accuracy of maximum likelihood is desirable for the eGuitar system, the construction of a reference data set is not viable.

3.6.3.3Hybrid Pitch Detection

It is possible to combine both time-domain and frequency-domain analysis into a hybrid temporal/spectral pitch detection approach. The layering of the approaches can in many cases account for the shortcomings of the complimentary analysis. Hybrid pitch detection are innately more computationally expensive, relying on both a time-domain autocorrelation function and a frequency-domain spectral analysis to determine pitch. The candidate pitches from both domains are ranked and the reported pitch determined by the highest ranking candidate.

3.7 Guitar Pickup

In order to analyze the notes coming from someone guitar we will first need to capture the raw data from each string. The most common way of doing this is to utilize a magnetic style pickup or a piezoelectric style pickup for each string. Each of these pickups utilizes a different method to convert the mechanical energy of the strings into electrical energy. This electrical energy can then be amplified and filtered to create audio waveform that our ears can pick up. The eGuitar System will interface directly with this audio waveform and analyze it to produce musical notes that can read by the user.

3.7.1Requirements and Specifications

3.7.1.1Polyphony

Polyphony by definition is the process of combining two or more sources in order to create a richer more complex output. Almost all string based musical instruments depend upon the property of polyphony in order to create complex audio waveforms that our ears can pick up and our brains find favorable. Since the eGuitar System deals specifically with guitars of all types' we will define polyphony as combining the individual notes, created by each string of the guitar, into a complex and harmonized melody. The output from the guitar that creates the polyphonic output can be seen in Figure 10. The resultant polyphonic output from the six individual audio signals can be seen in Figure 11.

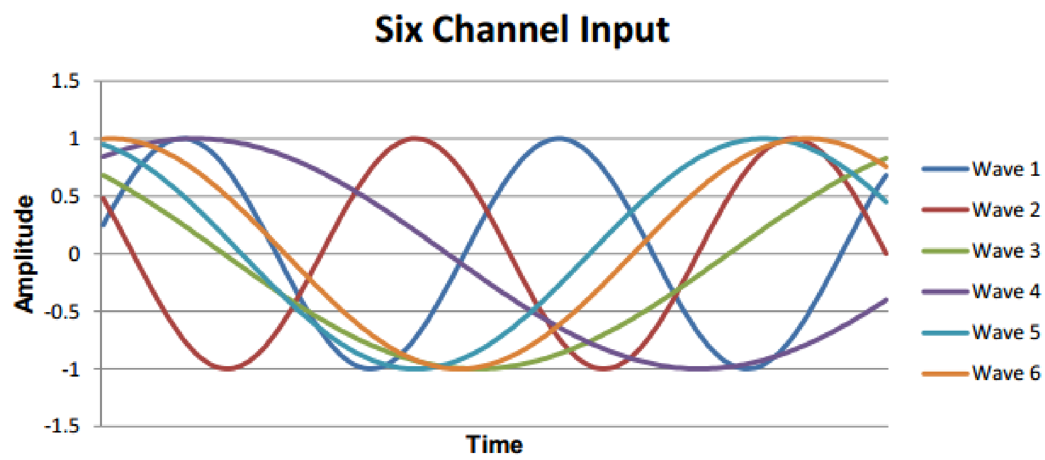


Figure 10 – Sample Six Channel Input

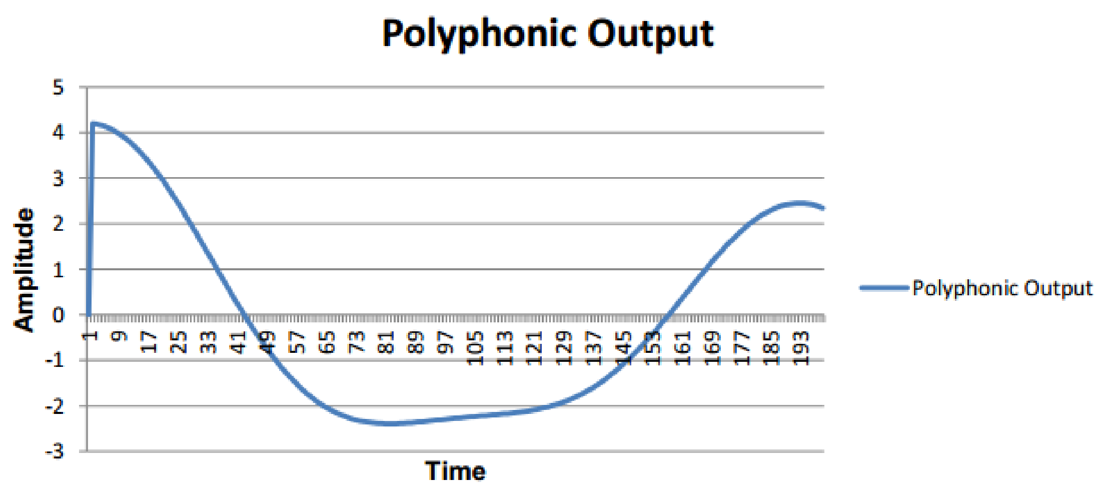


Figure 11 - Sample Polyphonic Output

Due to the complex resultant wave that polyphony generates, seen in Figure 11, the eGuitar System will endeavor to process the signal from each individual string. This will enable the eGuitar System to achieve a high level of accuracy and minimize complex waveform analysis, thus reducing power consumption and output delay.

3.7.1.2Frequency Response

Frequency Response is a measure of the total output spectrum of a device in response to an input or stimulus. This is often used to describe the operational characteristics of an electrical component or audio device. Frequency Response is usually calculated as a function relating the output frequency to that of the input wave. Several types of Frequency Response systems exist.

The first of which is a linear response system. In a linear system the output signal only varies from the input signal in relation to phase angle, the frequency and amplitude will remain unmodified. The Second type of system is a time-variant signal. This system allows for the output signal to change with respect to time even if the input signals magnitude remains constant. A simple example of this system is one where as time goes on a charge builds up to be released as one large signal. The final type of system is a time-invariant system. In this system does not change in respect to time. The best way to visualize this is a system where as time goes on the output signal stays constant. This system allows for changes in magnitude of the output wave as well as a phase shift.

The eGuitar System will focus primarily on the audio application of Frequency Response. Audio Applications of Frequency Response focus primarily on reproducing the input signal with little to no distortion. A Frequency Response that is too low will cause the output audio wave to distort and the sound produced will begin to screech or be cutoff. This is often heard in older style magnetic electric guitars and through speakers when the audios amplitude is greater than they were designed to handle.

Although the audio signals that the eGuitar System will analyze are not going to be used for playback, it is critical that distortion is minimized so that each note can be analyzed correctly. Distortion would limit the analysis that could be done on the waveform as the data is incomplete and contain significant error. On average guitars produce frequencies from 82Hz to 1319Hz, causing the required Frequency Response to be high, in order to minimize distortion.

3.7.1.3Signal Conditioning

Signal Conditioning is the processes of modifying or protecting a signal in order to make it suitable for post-processing. This process can contain, but is not limited to, amplification, converting, filtering and isolation. Each of these processes is designed to make the resultant signal easier to be interpreted and analyzed by another device. Depending on conditions and the form of the signal few or no Signal Conditioning processes may need to be applied to it.

3.7.1.3.1 Signal Isolation

Signal Isolation is the process of preventing distortion from other devices to interfere with a sensor or data as it is transmitted between the sensor and the analysis device. Isolating can be accomplished with magnetic shielding, converting the signal to another analog, or removing physical connections. In the eGuitar System, isolation is required at the pickup. Isolation is required there to ensure that the signal from each string is not influenced by the one next to it. This will ensure that we receive a clean signal for analysis and minimal cleanup will have to be performed on the waveform.

3.7.1.3.2 Signal Filtering

Signal Filtering is the process of removing noise in the signals frequency spectrum produced though amplification or poor Signal Isolation. In the eGuitar System, due to the positioning of each guitar string, noise ' generated on each guitar string's pickup from the surrounding guitar strings. Because of this phenomenon each input signal will need to be filtered to remove low frequency noise and waveforms that match those of the guitar strings around that particular input. Preforming these two processes will effectively remove noise in the signal and allow accurate note decoding.

3.7.2Magnetic Pickup

A Magnetic Pickup is a style of transducer used to capture mechanical vibrations and convert them into electrical impulses on stringed instruments. This style of pickup relies on the principles of Electromagnetic Induction, the process of generating a voltage and current by moving a wire around in a static magnetic field. The pickup is composed of a permanent magnet, referred to as a pole as seen in Figure 12 Left. These poles are then wrapped with several thousand strands of small copper wire as seen in Figure 12 Right. The permanent magnet then creates a small magnetic field around the copper coil and the guitar string. The movement of the guitar string then disturbs the magnetic field which changes the magnetic flux around the copper coil, thus generating a voltage and current in the copper coil.

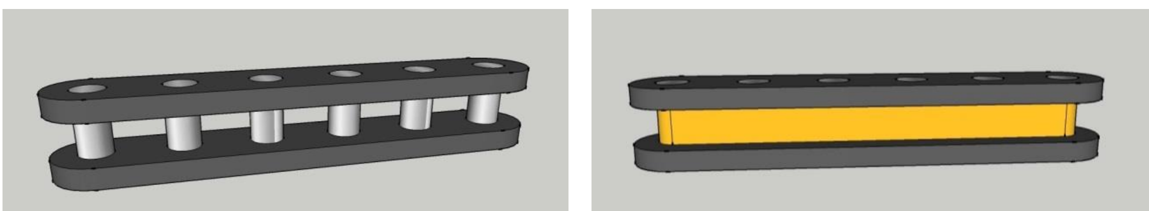


Figure 12 - Standard Guitar Pickup

The voltage that is generated in the magnetic coil will not however be constant, but will instead be a modulated waveform that represents the movement of the individual guitar string. Through amplification this signal can be recorded or played back though a set of speakers to create audio that our ears can interpret. Before amplification the voltage produced by a standard guitar will vary between 100 millivolts and 1 volt per string. The maximum voltage produced by strumming all 6 guitar strings is typically between negative and positive 5 volts.

Magnetic Pickups are simplistic to construct but possess several disadvantages. Depending on the number of times the copper wire was wrapped around the magnet there Frequency Response will change. The greater the number of copper turns around the magnet will produce a higher voltage but will also increase the resistance of the output. This will in turn cause distortion with high frequencies if the proper buffer amplifier is not in place. The second disadvantage is that the magnetic field will pull on each guitar string. This will dampen there movement and cause there oscillation to diminish much faster. The result will change the sound of the guitar on long notes. The final disadvantage is created by winding wire around the permanent magnet. This process creates a capacitance in the component that can then create resonance when connected to a cable that also contains a capacitance. The result is certain frequencies being accentuated and the overall sound to shift tones slightly.

3.7.3Piezoelectric Pickup

A Piezoelectric Pickup is a style of transducer used to capture mechanical vibrations and convert them into electrical impulses on stringed instruments. This style of pickup is commonly found in acoustic and semi-acoustic guitars and occasionally electric guitars. It produces a sound far different from a magnetic pickup on a typical electronic guitar. Instead the sound is similar to that of a non-electrified acoustic guitar. Piezoelectric Pickups are also commonly found in other stringed instruments such as violin, cello and double basses.

Piezoelectric Pickups rely upon the principles of the piezoelectric effect to generate a voltage. This effect is the process of converting mechanical energy into electrical energy through the deformation of crystalline materials that have non-inverse symmetry. This process can operate in both directions where electrical energy is created through mechanical movement or mechanical energy is generated by an induced voltage.

Since Piezoelectric Pickups do not rely upon a magnetic field for operation they avoid the disadvantages of the magnetic pickups. They do however require an amplification to produce an audible sound. Usually a buffer amplifier is used to avoid clipping and distortion of the output audio signal. Buffer amplifiers however require a relatively large voltage to avoid distortion of the audio signal. In most cases a nine volt power supply is used. If a buffer amplifier is not used then a preamp single-FET amplifier will need to be used. Single-FET amplifiers have the advantage of softer clipping of the audio wave but can distort sooner than buffer amplifiers.

Piezoelectric Pickups that are paired with a buffer amplifier provide a much greater frequency response than magnetic pickups. Because of this they are compatible with the wide range of frequencies that are produced by electric, acoustic and base guitars. This style of pickup also has the advantage of only requiring a single piezoelectric device to be attached to the guitar instead of one device per string.

3.7.4Optical Pickup

Optical Pickups are a recent development in the audio industry. They work by transmitting a beam of light towards the string. If the string is moving then this beam is picked up by a photodiode or phototransistor. The frequency at which the beam is cutoff can then be translated into an audio wave.

This style of pickup provides significant advantages over magnetic and piezoelectric style pickups. Since the audio signal is generated by the computer chip, according to the oscillation of the string across the beam of light, it provides a broad and flat frequency response that has no distortion or clipping. In addition this style of pickup does not affect the motion of the strings since no magnetic field is being generated and light does not provide any force on the strings. Finally noise due to the surrounding strings cannot be generated because only one string is capable of passing through each individual light beam.

Although this style of pickup provides significant advantages over the other types it has some major drawbacks. The cost of this style of pickup is five to ten times the price of a standard pickup. In addition to cost, these pickups currently require the guitar to be designed for their use. This is because of the unique size of the pickup and how it has to have a sensor on one side of the string and a laser on the other.

3.7.5Existing Solutions

In order to avoid modifications to the user's guitar the eGuitar system aims to simply add another pickup to the guitar through non-permanent means. In order to accomplish this, the eGuitar system needs to utilize a thin, cheap and accurate pickup. The eGuitar System could also utilize existing interfaces such as MIDI or analyzing the mono audio output directly.

3.7.5.1MIDI

Musical Instrument Digital Interface (MIDI) was created in 1983 by music industry representatives to standardize audio recording and transmission. MIDI is a standard that details a digital interface, connections and protocols for communicating with electronic musical instruments. The MIDI standard allows for a single cable to carry sixteen different audio channels. Each one of these channels can be used to carry information as a single note or string from that instrument or to carry the entire polyphonic output. It achieves this by sending audio as a set of messages. Each message can contain information about pitch, velocity, volume and vibrato. Along with each of these messages MIDI utilizes a unified clock signal across all devices.

Beyond transmission MIDI provides standards for audio storage and several other useful features. Since a MIDI signal is composed of simple messages, such as pitch, volume and frequency, it is a simple step to storing this data for use at a later time. Audio files that are created and stored using the MIDI standard are compact, often taking only a few kilobytes, and are only comprised of a few hundred lines of code. Since the raw information about each audio channel is stored individually modification of the audio file after it has been created can be easily achieved with no lose in quality. Along with minimal quality loss MIDI allows for the user to change the musical instrument that the file represents using a simple musical instrument synthesizer.

Although MIDI provides a powerful interface and transmission layer for audio it is seldom included on standard instruments. In most cases MIDI is used in electronic keyboards, audio sampling equipment and audio synthesis equipment. In particular guitars that contain MIDI outputs are rare and usually more expensive than a standard guitar thus making them uncommon by guitar players and beginners learning guitar.

In order to use a standard guitar with a MIDI interface an extra processing unit is required. These units accept standard mono input from the guitar, and then convert it into six separate audio channels, one for each string on the guitar. One of these devices often costs only a few hundred United States dollars and can be used with almost any instrument. This universal compatibility would extend the eGuitar's abilities from only guitars to electronic violins, cello and other similar instruments. The draw backs of a mono to MIDI converter however are substantial. These devices only allow simple audio signals such as individual notes to be encoded into MIDI. This means that complex cords cannot be converted. Since cords are required in most musical compositions this one drawback alone makes MIDI an unappealing standard to use.

3.7.5.2Additional Pickup

Another way for the eGuitar system to achieve multichannel audio sampling would be to utilize an additional pickup that is dedicated to sending audio to its DSP processor. This would enable the system to capture each individual string of the guitar while avoiding having to worry about compatibility with multiple types of pickups. This does however provide issues when it comes to the placement of the pickup, the size of the additional pickup and proper amplification of the signal before it can be processed.

3.7.5.2.1 Placement

Due to the way guitars are designed the strings are not parallel to each other as they travel from the nut to the bridge. This in turn means that the spacing between each string changes down the length of the guitar. Since the placement of the pickup along these strings will determine whether each string is over a single magnetic sensor or between two sensors, the eGuitar system will have to pick a specific spacing standard that can be used on common guitar configurations. The end user will also need to be able to easily place the pickup without extensive calibration. If the placement of the pickup is off significantly from its intended position it may be impossible for the eGuitar system to properly decode the audio signal into individual notes.

3.7.5.2.2 Pickup Size

Since the eGuitar system endeavors to maximize compatibility and minimize the setup by the end user, it is imperative that the guitar pickup, used for audio sampling, fits between the guitar strings and body. On average there is one inch of space between the body of a guitar and the strings between which we will need to fit our guitar pickup. Since piezoelectric pickups can attach to a variety of places on a guitar body and they are often extremely thin devices, usually no larger than one millimeter, size of these devices would not be an issue. Magnetic pickups on the other hand vary from one-half inch to one inch. This means that the eGuitar system will need to use a thin pickup to maintain compatibility with all common guitar types.

3.7.5.3Mono Audio Sampling

For the eGuitar system to avoid placing additional pickups on the guitar body it would need to be capable of connecting to the built in one-quarter inch mono audio output on the guitar. Since the waveform coming from this output is polyphonic and not individual channels, the eGuitar system will have to decode the signal into its individual components. To achieve this eGuitar system will have to utilize Digital Signal Processing (DSP) to analyze the output waveform in real-time so that tablature can be created for the user.

3.7.6Custom Build

In order to ensure all the project requirements are met the eGuitar system team may decide to utilize an in house custom guitar pickup. Since piezoelectric pickups cannot be made simply at home and optical systems are not fit for this project due to their size, complexity and cost, the eGuitar system team will have to create a magnetic pickup to meet there needs.

3.7.6.1Basic Design

In order for the eGuitar team to construct a pickup, it is important to understand the basics of how one works. As explained previously, electromagnetic pickups operate by voltage generated due to disturbances in an electromagnetic field. In order to measure the disturbance or flux of this magnetic field due to each string a set of six transducers will need to be used.

A transducer is a relatively simple device constructed with a combination of copper wire and permanent magnets will generate a voltage when there magnetic field is disturbed. In order to construct one the eGuitar team will need to acquire a small permanent magnet such as magnetized iron, nickel or cobalt. Other permanent magnets such as neodymium can be used they are not required and usually have an increased cost due to their rarity. The next step to creating a transducer is to wrap it with thin copper wire. Based upon the number of times the copper wire was wrapped around the magnetic this will produce a voltage from 100 millivolts to 5 volts which can then be amplified or processed to determine what notes were played on the guitar.

3.7.6.1.1 Magnet Types

The type of magnet used by the eGuitar team will determine the sound produced by the pickup and how noisy the output audio wave is. Since there are a number of permanent magnets, such as magnetized iron, nickel or cobalt as well as rare-earth magnets such as neodymium, that can be utilized to construct the guitar pickup it is important for the team to understand the differences between them. Rare-earth magnets often produce stronger and therefor larger magnetic fields. These magnets will thus generate a greater voltage when used however they will pull harder on the strings which will dampen the sound. In addition to dampening the sound the larger magnetic field will allow more than one string to influence the surrounding magnetic fields thus creating unwanted noise in the resultant signal. Permanent magnets that are not constructed from rare-earth elements however do not create such a strong magnetic field and thus will not be prone to producing a noisy output signal.

3.7.6.1.2 Coiling methods

The number of coils and the method of coiling the wire around a permanent magnet will significantly influence the sound produced. In order to increase the output voltage from the permanent magnet a greater number of coils can be utilized. This will allow the team to overcome the reduced voltage produced by not using rare-earth magnets. Increasing the number of coils however can cause the resistance of the wire to increase. This will cause high frequencies to distort quickly and will need to be properly handled. To diminish the distortion in the high frequency spectrum a buffer amplifier can be added between the guitar and the processing input.

3.7.6.1.3 Reuse of existing coils

In order to free up the eGuitar team to work on other aspects of the project, as well as reduce overall cost, existing coils can be reused to create the eGuitar's pickup. Relay switches are often used to enable low voltage embedded systems to control high voltage system such as home wiring. In order to achieve this they utilize a transducer in the form of an electromagnet. The relay would usually send a low voltage such as five volts through the relay which would then cause a piece of metal to move connecting the input and output of the high voltage source. The transducer used in relays is constructed by wrapping a permanent magnet with copper wire just as in a guitar's pickup transducer. Because of this the eGuitar team could utilized a set of these to create their pickup.

3.8 Embedded Memory

The tablature recording and playback portion of the eGuitar system will require embedded memory for tablature storage. The eGuitar system will also need to save user preferences to memory that won't be erased after a power cycle. While the relatively low space-cost user settings may be writable to the processor's built-in memory, a Secure Digital (SD) card will likely be needed for storage of tablature.

3.8.1Memory On-Chip

All researched DSP chips have at least some small amount of non-volatile on-chip memory that persists after power cycling. This type of memory is ideal for saving small persistent data like user preferences. The eGuitar system is likely to require saved user preferences for safe headset volume, recent tablature, most recent tuning and metronome tempo, etc.

3.8.2SD Card

Users of the eGuitar system has two options in transferring tablature to and from the PC to the embedded system-- physical SD card transfer and USB pass-through. These two options have relative advantages and disadvantages to both the group as developers and the future users of the eGuitar system.

3.8.2.1SD Card Transfer

In the first case of tablature transfer, physical transfer of the SD card from the embedded slot to an SD slot on the PC side will expose the SD card as a storage device, allowing the host operating system to access its contents. So long as the embedded system and PC can agree on a common supported SD card format, this method will require no software development is the more familiar method to users as a means of accessing and modifying the contents of an SD card. Unfortunately this method requires either the user's PC to have an SD card slot or the purchase of a separate SD-to-USB adapter. As not all users have this hardware and it is the prerogative of the eGuitar system to be as accessible and affordable as possible, a USB pass-through option is presented alongside the classic physical SD card transfer.

3.8.2.2USB Pass-Through

In the second case of tablature transfer, the eGuitar system will support USB pass-through, where the SD card can remain plugged into the embedded system as the user attaches the main embedded chip to the PC via a USB cable. This method adds a level of convenience for the end-user—plugging the embedded system into the PC will avoid physical SD card transfer and PC-side hardware restrictions, while also helping to charge the onboard battery over USB. In this scenario, the embedded chip responsible for USB communications will interface with the PC software via serial over a Communication (COM) port. Unfortunately, this software interface for serial data transfer on both the embedded and PC side of the eGuitar system would require additional development time to merely contribute an alternative method to a preexisting functionality.

3.8.2.3SD Variants

SD cards are available in a variety of families, form-factors and speed classes. These are described below in Table 9, Table 10, and Table 11 respectively.

As popularity and space demands have developed, so have the families of the SD format evolved. The three storage-focused SD card families are compared in regards to their capacities and preferred file system formats below in Table 9 - Secure Digital (SD) Card Families.

Family	Full Name	Capacity	File System Format
SDSC	Secure Digital Standard-Capacity	Up to 2 GB	FAT12, FAT16
SDHC	Secure Digital High-Capacity	2 - 32 GB	FAT32
SDXC	Secure Digital eXtended-Capacity	32 GB - 2 TB	exFAT

Table 9 - Secure Digital (SD) Card Families

SD card form-factors have adapted to meet the consumer demand for lighter, smaller devices, providing iteratively smaller dimensions and weights over time. The three SD card form factors are compared in regards to their dimensions and approximate weights below in Table 10 - Secure Digital (SD) Card Form-Factors.

Form-factor	Dimensions	Approximate Weight
Original size	24 x 32 mm	2 g
Mini size	20 x 21.5 mm	1 g
Micro size	11 x 15 mm	0.5 g

Table 10 - Secure Digital (SD) Card Form-Factors

As desktop and mobile consumer and enterprise computers have grown in their processing capabilities and data storage and manipulation demands, iteratively more performant SD speed classes have emerged. The various speed classes among SDHC and ultra-high-speed (UHS) standards are compared in regards to their minimum performance ratings and appropriate applications below in Table 11 - Secure Digital (SD) Card Speed Classes.

Speed Class	Min. Performance	Common Application
SDHC 2	2 MB/s	Standard-definition video recording
SDHC 4	4 MB/s	HD video recording (720p-1080p)
SDHC 6	6 MB/s	
SDHC 10	10 MB/s	1080p video recording
UHS 1 (U1)	10 MB/s	Real-time large videos
UHS 3 (U3)	30 MB/s	4K (2160p) video files

Table 11 - Secure Digital (SD) Card Speed Classes

The DSP board used for the eGuitar requires the micro SD format, while a passive micro-to-standard SD adapter is used for accessing the SD card on a PC with a standard-sized slot.

3.9 Power

In order for all of the various components to work for the eGuitar system, the correct power must be used. Incorrect voltage and current levels can either not turn on the corresponding parts or even damage them. The following sections will detail what power is required as well as the sources of power that were considered.

3.9.1Requirements

The eGuitar system implements a main control DSP unit, a microcontrolled LED fretboard matrix, and external input output such as LCD display. For each of these components, an individual power requirement is needed. The main control DSP unit is a Texas Instrument C5535 ezDSP developer board. This product does not require an external power supply, and draws its power from the USB connection made to the host computer. Therefore, the main control unit uses a 5V USB connection for power. The LED fretboard matrix requires enough voltage to power all 36 LEDs as well as the microcontroller that controls them. For a bi-color SMD LED, the typical forward voltage is 2.2V and a DC forward current of around 30 mA. Multiplying the forward current by the number of LEDs finds that there can be a total of 1.08A drawn if all the LEDs are turned on at once. The microcontrollers can operate off of 5V. Table 12 below compares all of these requirements.

	Voltage Req.	**Current Req.**	**Power Source**
C5535 ezDSP	5V Operating	0.7 mA	USB – PC
Bi-color SMD LED	2.2V	30mA	USB/AC/Battery
36 LED matrix	2.2V	1.08A	USB/AC/Battery
Microcontroller - USB	1.8-5.5V	0.2 mA	USB/AC/Battery
Microcontroller – DC jack	7.5V	0.2 mA	AC/Battery

Table 12 – Component Voltage, Current, and Power Requirements

3.9.2USB Power

The Universal Serial Bus (USB) power source offer a unique option in the fact that it is also a data transfer cable. USB power delivery offers a number of features specified on USB.org/. They explain that USB power has increased levels of power up to 100W [14], power direction is not fixed from the host to peripheral, and also allows level management of power between devices. USB cables are comprised of 4 individual cables, V_{bus}, D-, D+, and GND. The V_{bus} line provides +5V, and the D lines represent the data lines for communication. A benefit of USB power is the ability to use one USB cable to connect from the host PC to the main control unit to transfer data and power the C5535 ezDSP at the same time. However, in the final implementation of the eGuitar system, the data transfer may be done once and the system will no longer need to communicate to the base PC. If this is the case, powering through USB may not be a practical method for the eGuitar system.

3.9.3 AC Power

AC Power follows the same principles as the USB power source only without the data transfer capabilities. AC power can be found in wall outlets, which typically provide 15A at 125V. An AC-DC power adapter plug would be needed to use the wall outlet and bring the power level down to 5V for the purposes of the eGuitar components. An example of this type of plug is a switching AC/DC power adapter, which gives a regulated 5V output at up to 2A with an input from 110V to 240V. Using AC power from a wall outlet runs into the same problem that using USB power has. In order for the eGuitar to be a wireless system, there cannot be any external connections, including power source. Therefore, AC power may not be the best option for this system.

3.9.4 Battery

Using batteries to power the eGuitar components brings the benefit of portability to the system. Batteries can be housed next to the component that they are powering, which means that the final eGuitar product does not need any external connections. There are many types of batteries and there is also the issue of charging or replacing the batteries. Batteries can be made up of different chemicals and also come in various sizes like AA or 9V batteries.

3.9.4.1 Types

Batteries can be comprised of many different chemical compounds, all with their own benefits and drawbacks. Certain elements are rechargeable, while others contain less toxic chemicals for the environment. An article written by Battery University asks the question, "What's the Best Battery?" In the article, they detail the differences between each battery type [15]. The types being compared are Nickel Cadmium (NiCd), Nickel-Metal Hydride (NiMH), Lead Acid, Lithium Ion, and Lithium Ion Polymer. Typical AA batteries only use NiCd, NiMH, and Lithium Ion types from the list above, so that is what is focused on here.

Nickel Cadmium (NiCd) is a type of low energy density battery, but they do offer long life and low economic price. Nickel-Metal Hydride (NiMH) is a higher energy density type than NiCd, but its battery life is lower. NiMH has no toxic metals as well. Lastly, Lithium Ion batteries are lightweight and long lasting, at the expense of a higher price for their qualities. Table 13 below shows some specifications of each battery type.

	NiCd	NiMH	Li-ion/Polymer	Reusable Alkaline
Energy Density	45-80	60-120	110-160	80 (initially)
Internal Resistance	100-200 mΩ	200-300 mΩ	150-250 mΩ	200-2000 mΩ
Cycle Life	1500 uses	300-500 uses	500-1000 uses	50 uses
Charge Time	1 hour	2-4 hours	2-4 hours	2-3 hours
Overcharge Tolerance	Moderate	Low	Very low	Moderate
Self-discharge per month	20%	30%	10%	0.3%
Cell Voltage	1.25V	1.25V	3.6V/3.7V	1.5V

Table 13 – Battery Specifications

From the requirements above, the maximum voltage needed for system operation is 7.5V. In order to provide this maximum voltage value, multiple batteries are needed. Six NiCd or NiMH batteries will need to be connected together to offer 7.5V while only 5 reusable alkaline batteries are necessary. For lithium ion batteries, there only needs to be three connected together to provide enough voltage for all components.

3.9.4.2 Sizes

In addition to the types of batteries out there, each type can come in a variety of sizes. These sizes determine the capacity of each battery. Table 14 below shows the battery size, capacity, and typical drain for a number of battery sizes. The second half of the table assumes lithium-ion batteries are used and the numbers reflect 3 batteries being connected together [16].

Battery Size	Capacity (mAh)	Typical Drain (mA)
9V	500 mAh	15 mA
AAA	1000 mAh	10 mA
AA	2000 mAh	50 mA
C	6000 mAh	100 mA
D	12000 mAh	200 mA
Lithium Polymer	2500 mAh	-
3 Li-Ion 9V	1500 mAh	45 mA
3 Li-Ion AAA	3000 mAh	30 mA
3 Li-Ion AA	6000 mAh	150 mA
3 Li-Ion C	18000 mAh	300 mA

3 Li-Ion D	36000 mAh	600 mA

Table 14 – Battery Size, Capacity and Drain

Capacity in mAh measures the discharge rate of a given battery under certain scenarios. For example, a battery with 2000 mAh capacity running at 500 mA will discharge in 4 hours. For the eGuitar system, with all LEDs on, the total current draw from the system would be 1.1A. Table 15 below details how long the different sizes will last at this max current draw and also at average draw. The second half of the table assumes three lithium ion batteries are being used to power all components.

Battery Size	Operating Time Max	Operating Time Average	Hours Playing/Charging Ratio – Li-Ion
9V	27 minutes	4.1 hours	-
AAA	54 minutes	8.26 hours	-
AA	1.8 hours	16.5 hours	-
C	5.45 hours	49.5 hours	-
D	10.9 hours	99 hours	-
Lithium Polymer	2.27 hours	20.6 hours	5.15
3 Li-Ion 9V	1.36 hours	12.4 hours	3.1
3 Li-Ion AAA	2.72 hours	24.8 hours	6.2
3 Li-Ion AA	5.45 hours	49.6 hours	12.4
3 Li-Ion C	16.36 hours	148.8 hours	37.2
3 Li-Ion D	32.73 hours	297.5 hours	74.38

Table 15 – Battery Life

From the results above, most battery options are suitable for the purposes of the eGuitar. However, it is interesting to note the ratio given in the fourth column. For Li-Ion AAA batteries, the user will typically get 6 hours of playtime for every 1-hour charged, and that number doubles for AA batteries.

3.9.4.3Charging System

A preferred feature of the eGuitar system is ability to recharge the battery that runs the components. With this functionality, the user never has to worry about buying and replacing new batteries for their system. The challenge that appears for a charging system is that the battery must be able to charge and be connected to the circuit at the same time. This means that a standard recharging dock for certain batteries will not work unless a lot of modifications are made to the charging dock. Furthermore, a custom PCB is designed to host both the LED matrix controller as well as a charging unit.

Sparkfun manufactures a lithium polymer USB charger and battery, which can charge the lithium polymer battery while still powering the main systems. The charger is a highly advanced linear charge management controller and goes from charge initiation to completion and automatic recharge. Figure 13 is a flowchart detailing this process. For a schematic of the USB Charger, refer to Section 3.3.3 - Reference Designs.

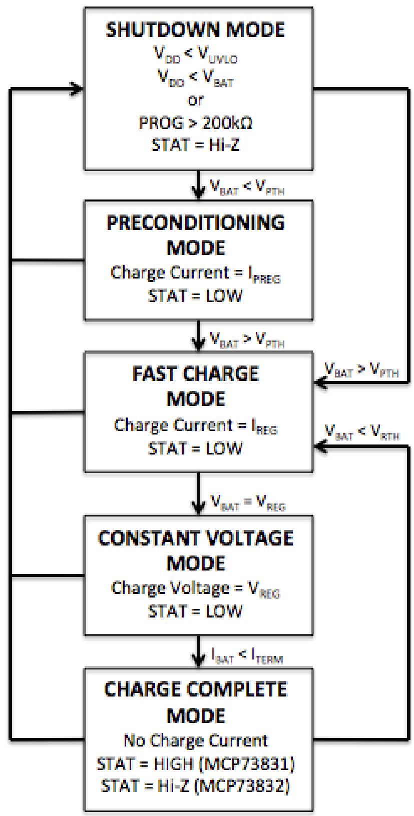


Figure 13 - USB Charger Flowchart

The charger begins in Shutdown mode until the supplied voltage level rises above the undervoltage lockout (UVLO) threshold. The UVLO circuit controls whether the device is in shutdown mode or not. This means that the UVLO can detect when the input voltage drops below the threshold then the system is shutdown. The V_{BAT} pin is used to determine whether a battery is connected or not. If V_{BAT} is less than the precondition voltage threshold ratio (V_{REG}), then the system knows that a battery is present and continues on to checking the charge conditions. For a charge to begin, the PROG pin is checked for the current regulation set and charge control. If the PROG pin is open, then the system shuts down and begins from the top of the flowchart. However, if the PROG pin is not open, then the Trickle Chard mode is activated, which can be found in the preconditioning step. The voltage at V_{BAT} will slowly rise until it reaches the preconditioned threshold. Once this threshold is met, and then the system moves into the Fast Charge mode. This mode is kept on until the regulation voltage (V_{REG}) is reached, and then switched to Constant-Voltage mode. Constant voltage regulation is terminated when the average charge current drops below a certain percentage that is determined by the resistor value connected to the PROG pin. Once this occurs, the charger enters the Charge Complete mode. This mode stays on until the voltage at V_{BAT} goes below the recharge threshold, which then begins the cycle again of recharging the battery. The USB charger also has temperature regulations where the system will shut down if a certain temperature is reached. Charging will resume after the system has cooled down [17].

Texas Instruments makes a similar product in the Fuel Tank BoosterPack. This kit comes with a rechargeable lithium polymer battery and is designed to power TI LaunchPad development kits like the MSP430. The following will detail some of the schematics for this charging system.

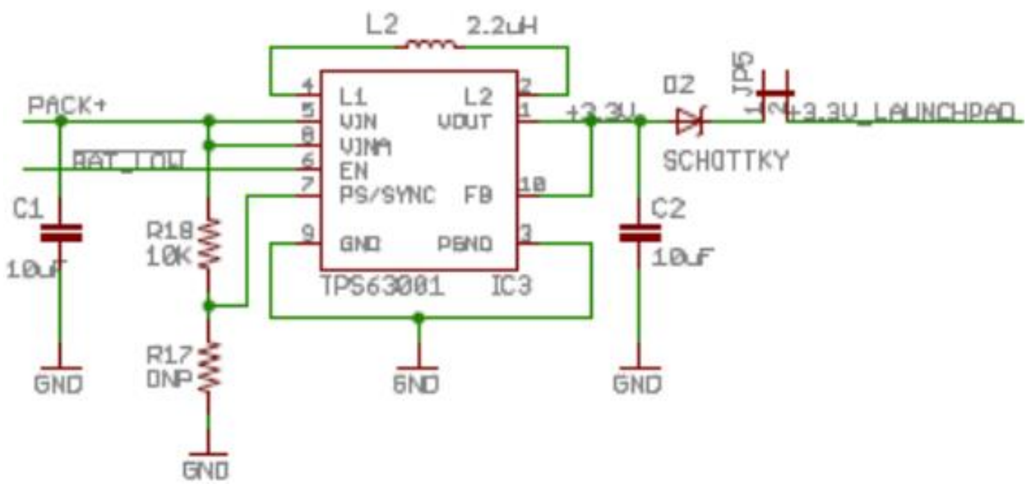


Figure 14 - TPS63001 Schematic
Reprinted with permission from Texas Instruments

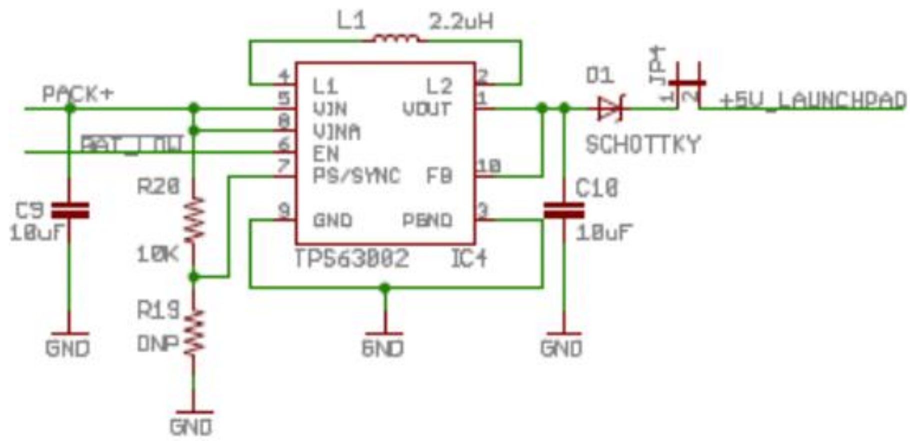


Figure 15 - TPS63002 Schematic
Reprinted with permission from Texas Instruments

The TPS63001 or Buck-Boost Regulator in Figure 14 above is a power converter, which has an output of 3.3V with 1200mA output current. Figure 15 shows the TPS63002, which has an output voltage of 5V with 1200mA output current. Both chips use the Buck-Boost operation that automatically switches from restricting voltage when the input voltage is high to boosting it when the input voltage is lower than the output voltage. High efficiency is achieved with this system due to the fact that the components that are being powered may not require 5V at all times. However, for the eGuitar system, the battery pack is powering both the LED control unit and the DSP processing unit, so the power efficiency may not be as useful.

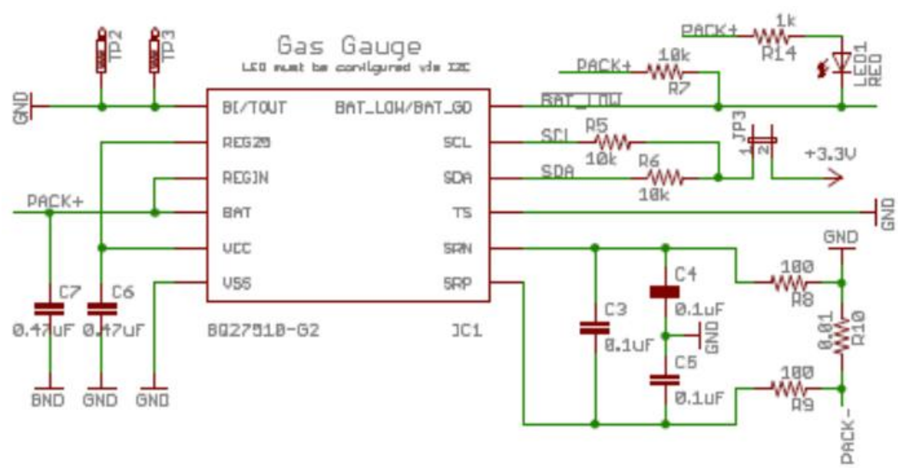


Figure 16 - BQ27510-G2
Reprinted with permission from Texas Instruments

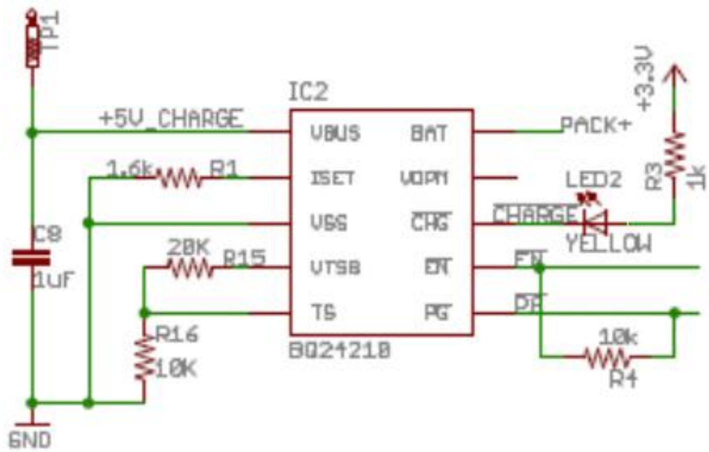


Figure 17 - BQ24210
Reprinted with permission from Texas Instruments

Figure 16 above is of the BQ27510-G2 Li-Ion battery fuel gauge. This device can detect charge percentage, remaining run time, battery voltage, remaining capacity, and temperature. The device communicates over I^2C to the main control unit. Texas Instruments has an Impedance Track™ algorithm that is used to accurate check the battery life of a lithium ion battery pack. When the power coming from the battery at the PACK+ line is low, LED1 light turns on and notifies the BQ27510 of the low battery. For the purposes of the eGuitar system, an indication of low battery would aid in the user experience. The user would visually be told that the battery is low instead of finding out when the system powers down by itself. Figure 17 shows a schematic of the BQ24210, which is used as the charging circuit for the lithium polymer battery. The charger has bi-directional power capabilities and can switch operating modes automatically depending on the battery status. Like the Sparkfun product, the BQ24210 has a pre-charge phase, fast-charge phase, and voltage regulation stage. The main functions of the stages are detailed in the paragraphs above. When the charge stages are in effect, the device sends a signal to LED2 to show the user that the battery is being charged.

4 Design

4.1 PCB

4.1.1 Requirements

From the research performed in Section 3, it is possible to begin design work for the LED control unit PCB. The discovered requirements are the foundation for a functioning prototype of the eGuitar system. The PCB designed for this project includes a microcontroller, input output expander, battery-charging unit, and on-board memory. The following three sections detail the requirements for all of the components.

4.1.1.1 Power Levels

Having the correct operating voltage and current levels is key for proper functionality. For the ATmega32u4 microcontroller that has been selected, an operating voltage of 3.3V is required. The LED matrix is powering upwards of 36 LEDs at once. Each surface mount LED runs off of 5mA, so the total current for the LED strip is 0.18mA. Table 16 below is a summary of the power requirements for each component.

	Voltage Req.	Current Req.	Power Source
ATmega32u4	5V Operating	0.7 mA	Battery
SMD LED	2.2V	5mA	Battery
36 LED matrix	2.2V	0.18mA	Battery

Table 16 – Power Requirements

4.1.1.2 Onboard IC's

The PCB designed for power and LED control feature the following integrated circuits (ICs).

- ATmega32u4 – LED microcontroller
- PCA9698 40-bit I/O port – Input Output Expander for LED fretboard matrix
- TPS63001/2 – Buck-Boost Power Converter
- BQ27510 – Fuel Gauge Detector for battery
- BQ24210 – Lithium Ion Battery Charger
- HJ4050 – Hex Converter

4.1.1.3 PCB Specifications

The designed PCB using the above components are a 2-layer design. A 44-pin QFN package is used for the ATmega32u4 chip, as well as surface mount components for the control board. For the other components, simple copper pads are designed into the PCB for mounting. The final PCB design is shown in Figure 18 below. The left side of the board has USB and SD card connections. The bottom left has a voltage regulator and hex converter for the SD card communications. The bottom right features the ATmega32u4, which connects to the 40-bit I/O expander on the right side. In the top middle of the board, the charging circuit is shown. All connectors are designed for the edges of the board for easy access and plug-and-play capabilities. The red colorization shows the top layer of the board, while the bottom layer is represented by the blue color. The bottom layer also features a polygon pour for the ground layer.

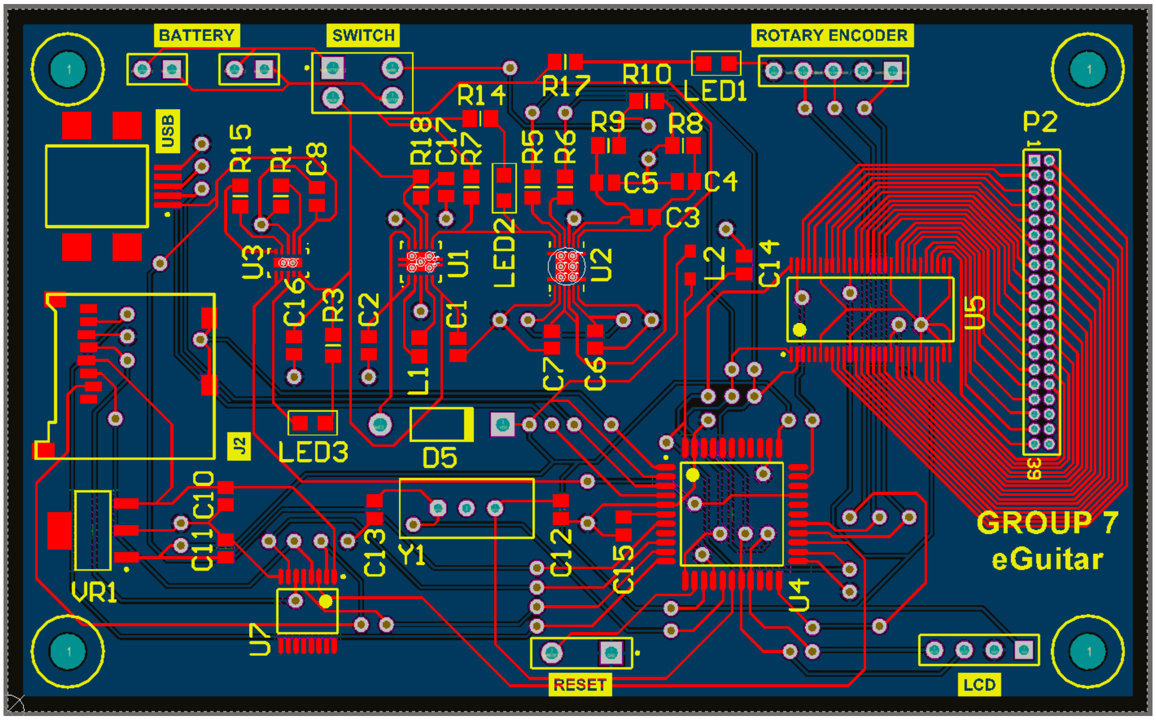


Figure 18 – Final PCB Design

4.1.2 Hardware Configuration

4.1.2.1 Circuit Design Figures

The figure below shows the schematic drawing of the main control board.

The figure below shows the schematic drawing of the main control unit. The unit includes the ATmega32u4 chip, the 40-Bit I/O Expander, OLED and Rotary Encoder connections, full charging circuit, and SD card. The ATmega32u4 sends out signals via I^2C to the 40-Bit I/O Expander to control each LED individually. It also communicates to the SD card via SPI as well as getting USB data via RS232. The board is controlled by a rotary encoder and an OLED that displays the user interface.

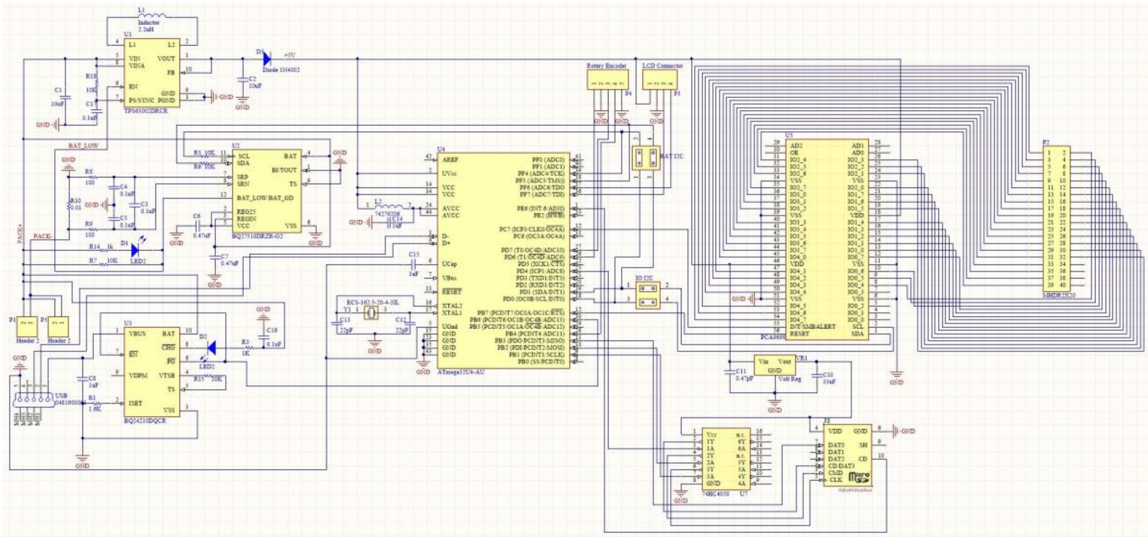


Figure 19 – Control System Schematic

4.2 LED Fretboard Matrix

4.2.1Hardware Configuration

The LED fretboard matrix is comprised of 6 individual rows of 6 LEDs. A 40-Bit I/O Expander, connected to the ATmega32u4, controls each LED. The LEDs each have a required input current of 5mA, and the resistance needed before each LED is 680 Ω. The LEDs and resistors are mounted on a single layer of FR4 material. 6 individual PCBs were created for the 5 supported frets. All six boards are connected to a single 20x2 ribbon cable that leads down to the main control board. A 3D model was created to show the general physical look of the LED matrix underneath the strings. Below, Figure 20 and Figure 21 show the 3D representations. The final PCB design of the LEDs is shown in Figure 22. Figure 23 shows the LED Fretboard Matrix schematic.

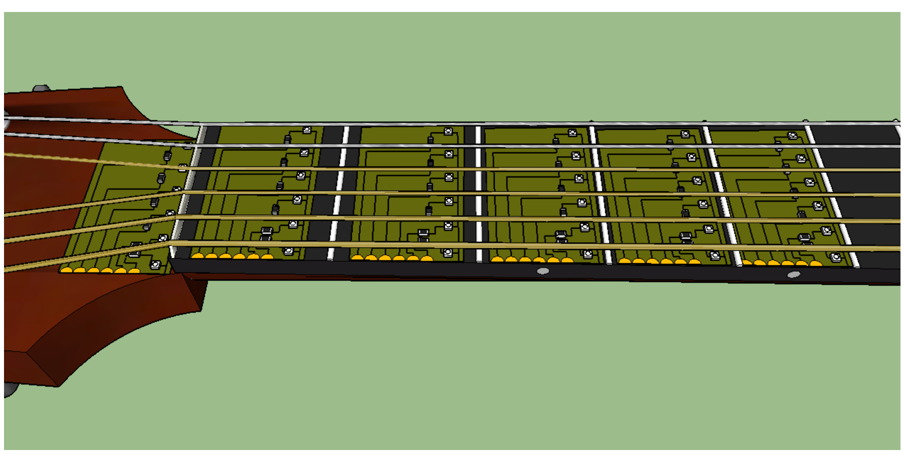


Figure 20 - 3D Representation of LED Matrix

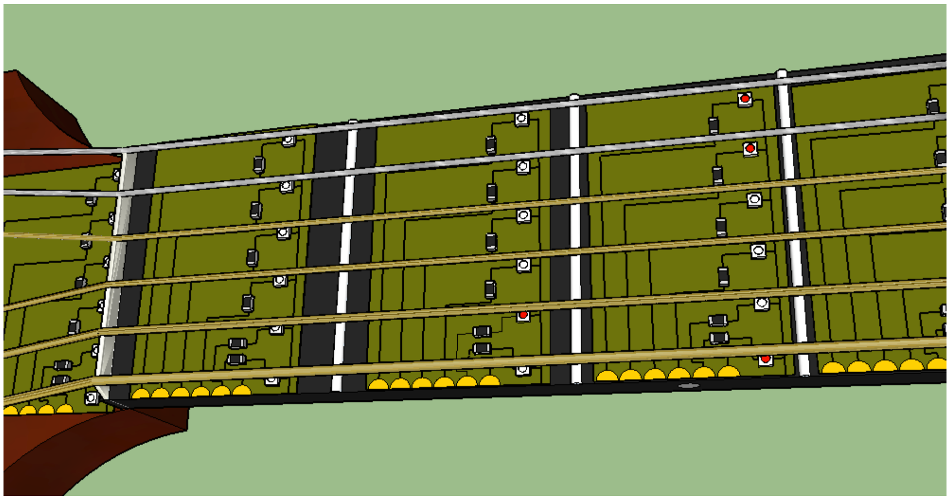


Figure 21 – Visualization of a G Chord

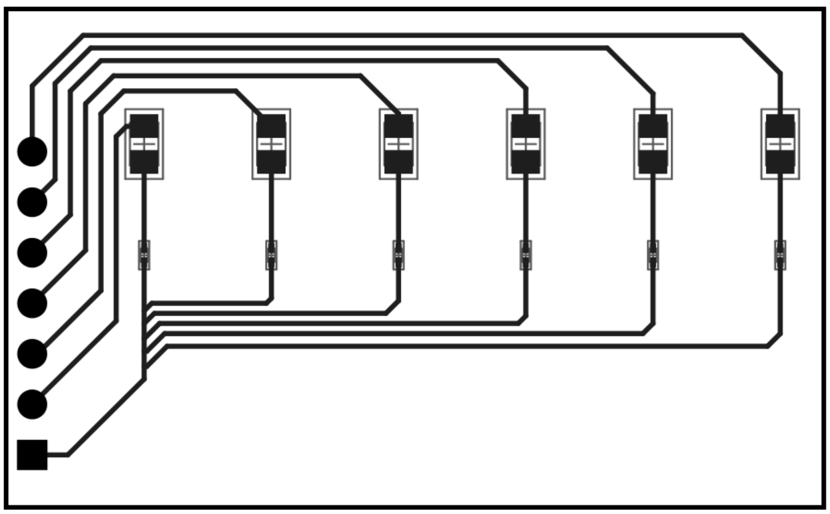


Figure 22 – LED PCB Layout

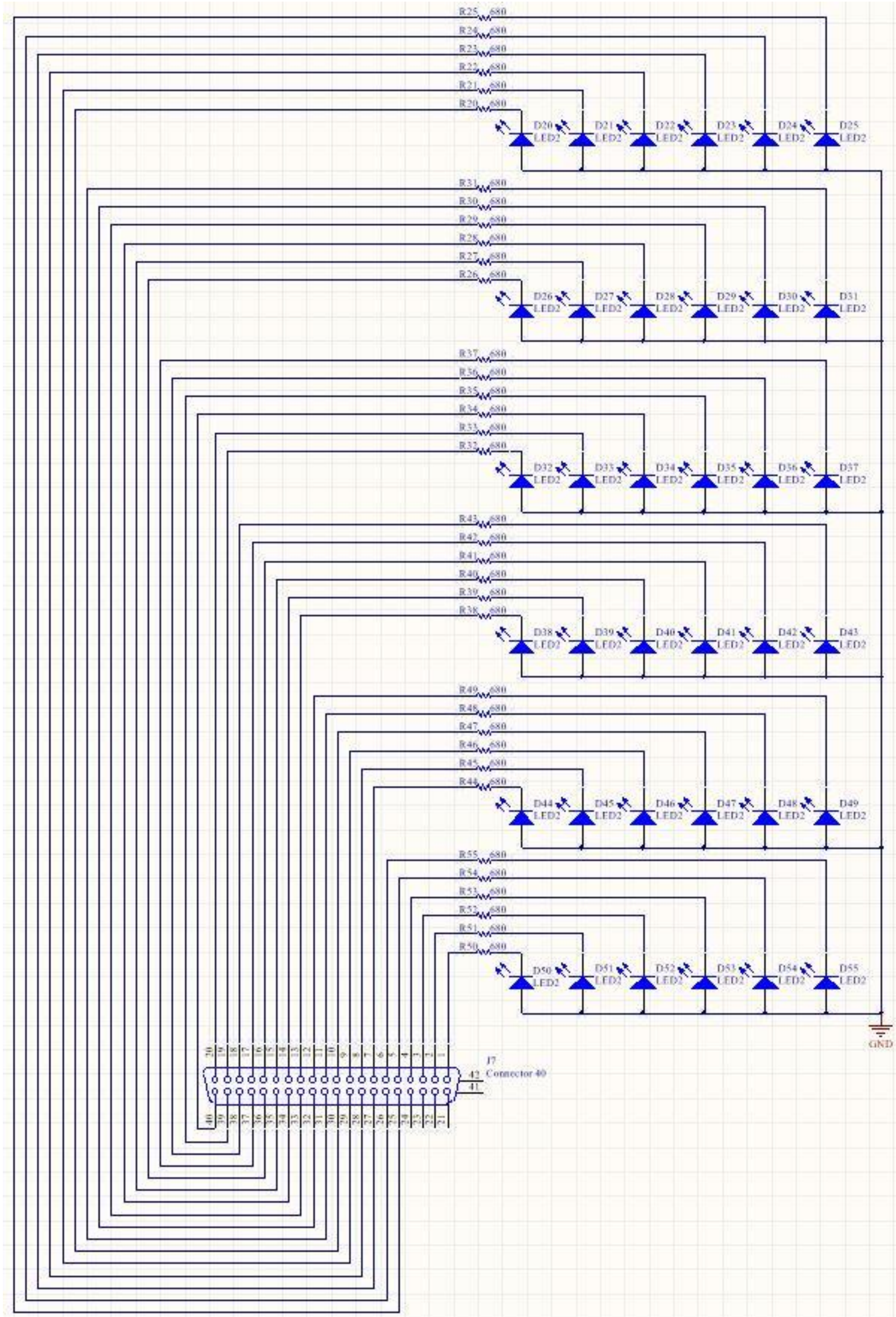


Figure 22 – LED Fretboard Matrix Schematic

4.2.2 Software Communication

Individual LEDs are sent signals from the 40-Bit I/O expander, which receives control signals from the ATmega32u4. The ATmega communicates to the I/O expander via I^2C. Details on this communication method can be found in section 3.1.4. The ATmega also can receive serial pass-through from the PC application.

4.3 User Interface

4.3.1 Embedded UI Hardware

The aim of the embedded eGuitar system is that all user options and functions be navigable with minimal and intuitive hardware controls. For this purpose, a single 3D-printed control box containing the DSP chip and LED matrix control chips also house the display and user controls. The concept design for this control box was modeled by hand specifically for the eGuitar System in Trimble Sketchup and is shown below in Figure 23 - Onboard Enclosure 3D Model.

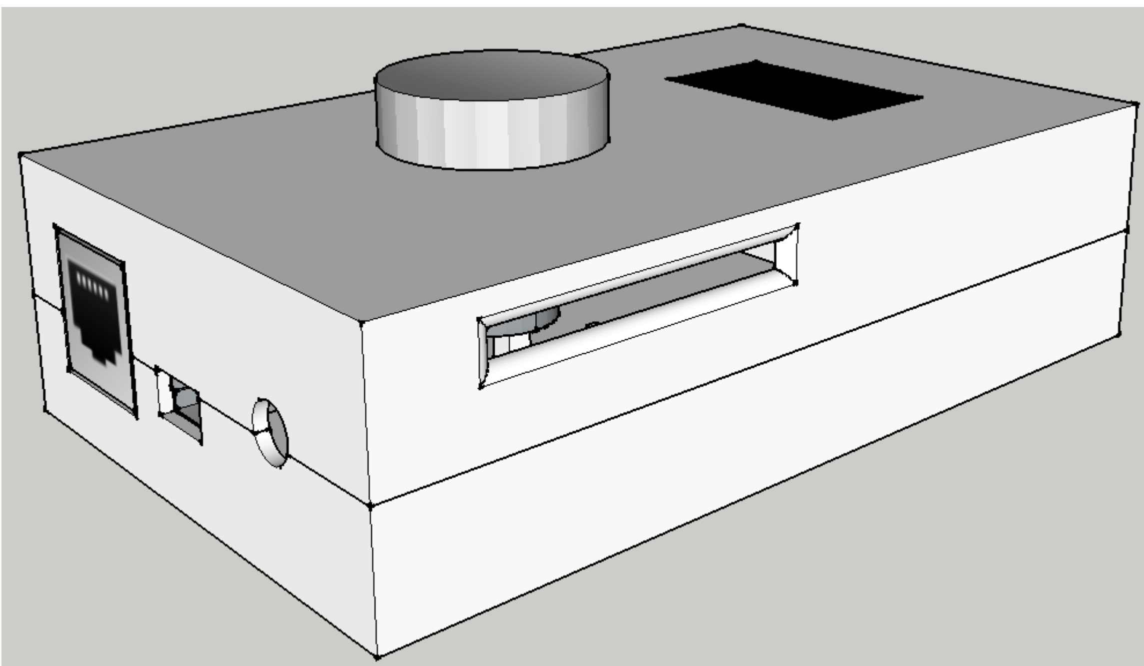


Figure 23 - Onboard Enclosure 3D Model

The primary user-facing controls include the following:

- A 128x64 pixel monochrome OLED display
- A push-button rotary encoder for option navigation and selection
- A power switch to engage or disengage power
- A Mini-B USB female port for tablature transfer and charging
- SD card slot
- Ribbon Cable opening

4.3.1.1 Purchased Components

The 3D printed control box uses purchased components where possible, in an effort to save development time and maximize compatibility between parts. The power switch, OLED display, push-button rotary encoder, audio jack, audio stereo potentiometer, battery and battery plug is sourced from online vendors who supply datasheets that guarantee adherence to the requirements and specification eGuitar system.

4.3.1.1.1 Power Switch

The power switch is perhaps the simplest of the purchased components for the embedded enclosure. Power switches normally only require connection to their power in and out terminals. As a nicety, a power switch with a built in power-indicator LED is chosen for the control box. This type of power switch has a ground line for the built-in LED alongside the standard power I/O terminals. The wiring for such a switch is shown below in Figure 24 - LED Power Switch 3D Model.

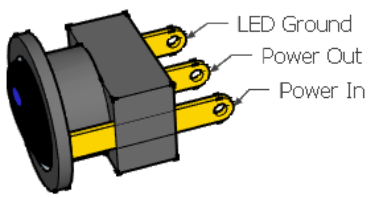


Figure 24 - LED Power Switch 3D Model

4.3.1.1.2 OLED Display

The OLED display selected for the control box supports both the I^2C and SPI standards. The target resolution is at least 128x64 pixels to support a wide array of use cases like waveform visualization or tuner mode. For affordability, the chosen display was monochrome. The display is self-illuminated and power-efficient by nature of OLED technology, with a wide range of operating temperatures and supported input voltages somewhere in the 3-5V DC range. The practicality of the SPI and I2C standards is shown below in Figure 25. Requiring only two control lines in addition to power and ground, a display like this contributes greatly to the simplicity of the eGuitar system design.

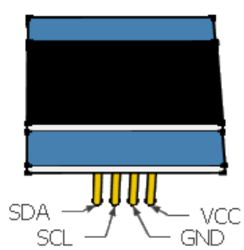


Figure 25 - SPI/I^2C OLED 3D Model

4.3.1.1.3 Push-Button Rotary Encoder

The rotary encoder is the key part in minimizing UI hardware for the embedded control box. The rotary encoder allows for the software to read the absolute rotational position as well as track rotational motion. As a potentiometer offers only the former, the rotary encoder has the advantage of infinite rotation. The rotary encoder selected for the control box includes a push-button that is triggered when the knob shaft is depressed. The Encoder A/B lines are those which are read for binary of Gray code, while the Push A/B lines operate the same as a pushbutton's two pins. The wiring for such a rotary encoder is shown below in Figure 26 - Pushbutton Rotary Encoder 3D Model.

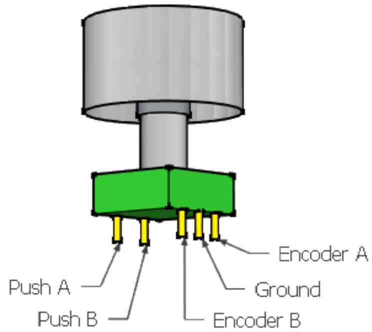


Figure 26 - Pushbutton Rotary Encoder 3D Model

4.3.1.1.4 Battery and Battery Plug

While the eGuitar aims to include a rechargeable battery, the initial design concept relies on a standard 9V battery, connected with a standard 9V battery plug. This aims to ensure ease of early testing of the power management systems while allowing independent testing of the logic boards and other hardware. This battery is included as part of the user interface in that this implementation would require the user to swap batteries as each runs out of power. The standard 9v battery and plug are shown from a top-side perspective within the battery bay of the enclosure on the left side of Figure 27 - Battery Bay 3D Model. For reference, the right side of Figure 27 shows the battery door on the outside of the enclosure, as viewed from the underside.

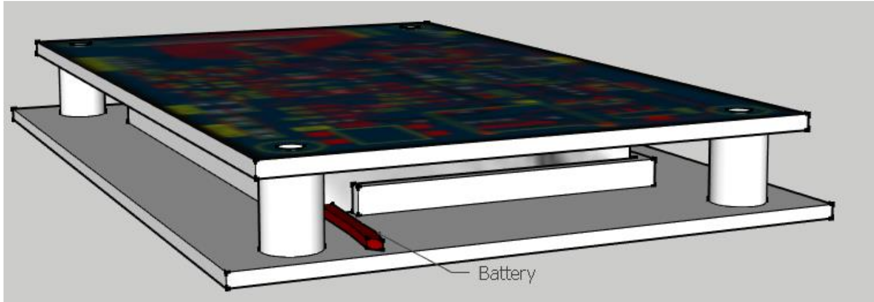


Figure 27 - Battery Bay 3D Model

4.3.1.2Custom Components

The onboard control box enclosure relies heavily on custom 3D printed parts. Figure 28 - 3D Printed Components shows each component from a perspective indicating depth.

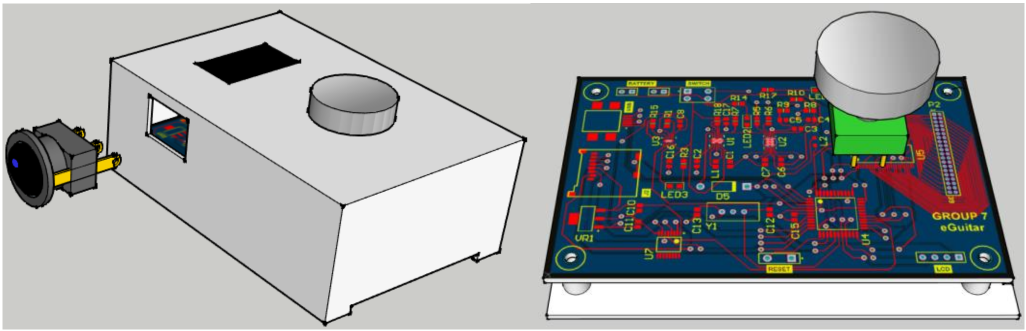


Figure 28 - 3D Printed Components

The custom 3D modeled parts include the following:

- A bottom shell with protruding columns for screw holes
- A top shell with per-component recesses and standoffs
- A battery bay divider piece fully enclosed when assembled

The 3D models were developed by a combination of measurements from part data sheets and iterative trail-and-error 3D prints. Each component has fitted recesses and/or standoffs built into at least one of the two main shells so that all parts fit snugly and securely when assembled. The final assembled unit has a seam separating top and bottom shells. The two shells are attached with small machine screws.

4.3.2Embedded UI Software

The embedded UI software design involves generalized layouts for various operation modes, settings and options. The display libraries available accommodate text, rectangles and even individual pixel control. While individual pixel control allows for virtually limitless possibilities (2^{128*64} for a single-brightness monochrome display to be exact), the UI design is limited to simple, legible and intuitive layouts so that even a small display can be information-dense.

4.3.2.1Menu Navigation

The highest level of user interaction with the onboard control box entails menu navigation. As soon as the control box is powered, the display shows a main menu whose direct submenus can be selected between. Selection of a submenu results in further submenus until the user reaches an atomic option or mode selection. Every menu besides the main menu has the main menu as its top option for easy return. Menu options are navigates such that a clockwise turn on the rotary encoder shaft results in a downward traversal, counter-clockwise results in an upward traversal, and a shaft click results in a confirmation of the highlighted option. Figure 29 - Menu Navigation shows an example submenu titled “Utilities” with three submenus and the options to return to the main menu.

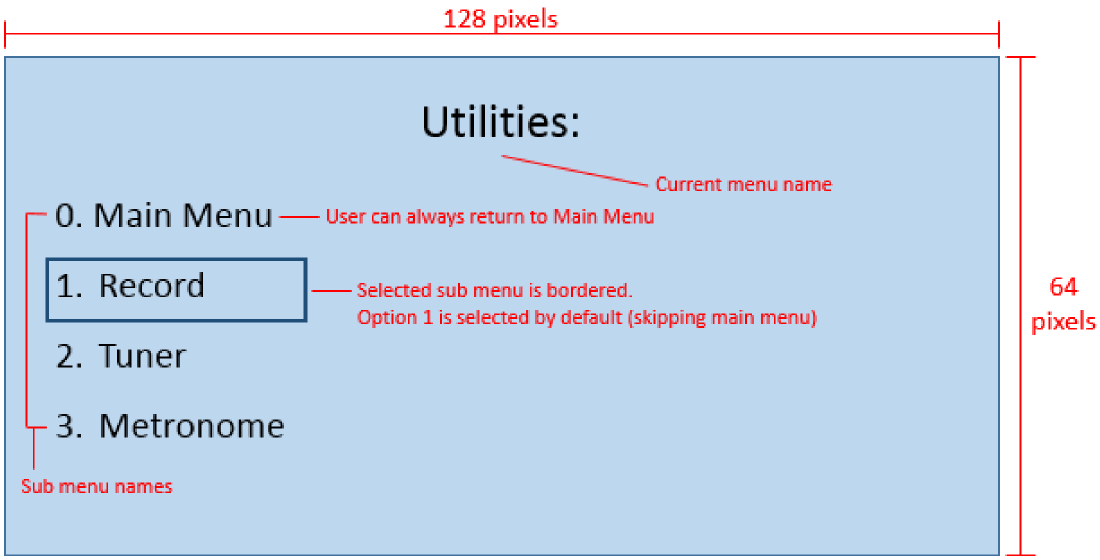


Figure 29 - Menu Navigation

4.3.2.2Absolute and Relative Values

When the user has navigated the menu tree to arrive at an atomic setting or mode, the mode was to be either be based on either an absolute or relative value. The display and modification of both values entails a screen with a title, current value, and value line. In the case of absolute values, the reference values are of a fixed range e.g. 0-100% for volume. In the case of relative values, the reference values are values that neighbor the current value e.g. adjacent notes when tuning a guitar string. Absolute and relative values are shown below in Figure 30 - Absolute Value: Volume and Figure 31 - Relative Value: Tuner respectively.

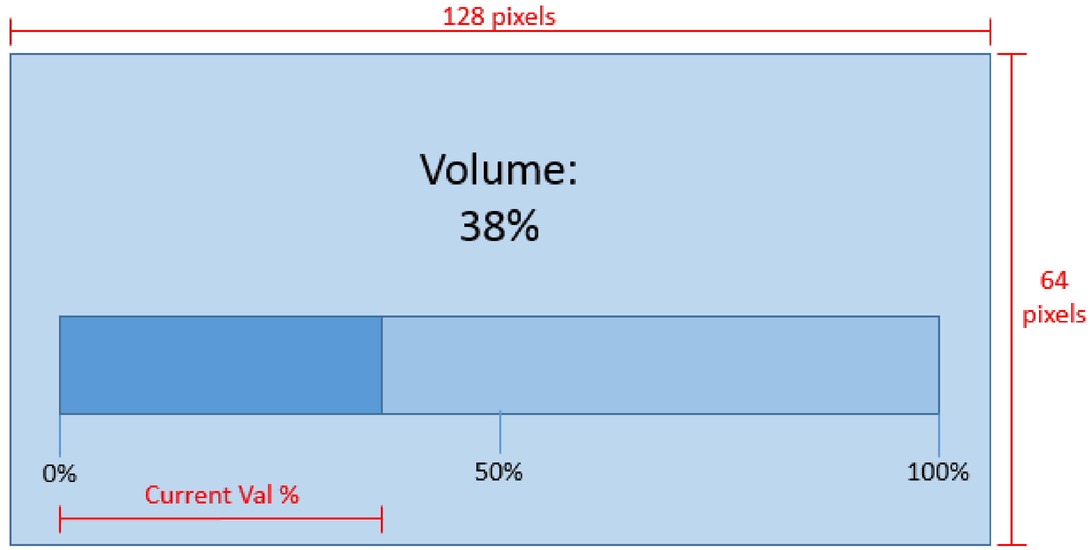


Figure 30 - Absolute Value: Volume

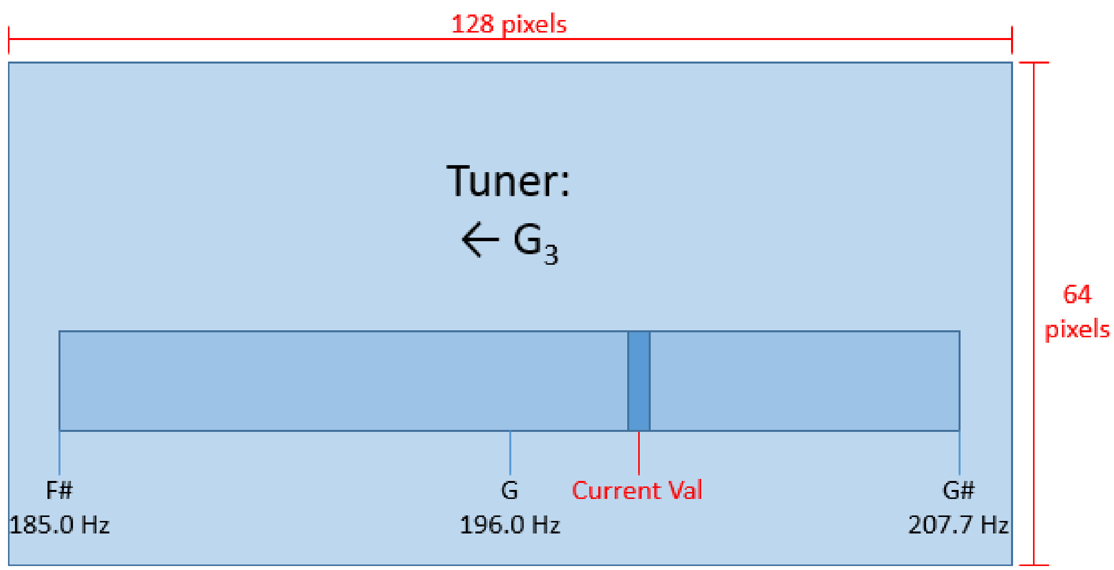


Figure 31 - Relative Value: Tuner

4.3.3PC UI Software

~~As the final appearance of the PC user interface will largely be dictated by its final implementation, the UI design will only be discussed at high level here.~~ The PC software serves as a mechanism for tablature recording, editing, transfer, format import/export and possibly playback. As such, these modes ~~will~~ are~~likely be~~ navigable within a single window via ~~a top ribbon, menu bar and side panels~~a tabbed pane. The goal with the eGuitar PC UI is to provide the user with an intuitive and legible interaction. The UI also supports drag-and-drop tab importing.

4.4 Digital Signal Processor (DSP)

The DSP software design largely falls to a decision on a first-attempt pitch detection algorithm. If functional, the pitch detection algorithms provided by TI is used for the eGuitar system. If these functions are insufficient for the project's purposes, a custom pitch detection implementation must be considered. Given the various existing time-domain and frequency-domain algorithms researched, a custom implementation would likely involve a hybrid temporal/spectral approach involving zero-crossings from the time domain compared against fast Fourier transform based spectral plot of the frequency domain. Ultimately, the algorithms provided by TI were insufficient as well as an FFT solution. Therefore, the eGuitar digital signal processing algorithm utilizes two stage auto-correlation. Auto-Correlation is usually slow to detect pitch since it has to iterate overall all samples, however the two stage implementation significantly decreases processing time. The first stage of the eGuitars pitch detection pulls every fifth sampled voltage and determines a rough upper and lower bounds frequency for each string. This upper and lower bounds is then fed into the second stage where it is refined down to 0.2% of the actual string frequency. The frequency is then compared against a frequency table to determine the correct MIDI note.

4.5 Guitar Pickup

The eGuitar system endeavors to be a cheap and easy to use system. In order to achieve this the guitar pickup that is chosen needed to be compatible with all guitar styles while maintaining a low cost point, simple design and accurate frequency output. To achieve this the eGuitar team elected to salvage a hexaphonic pickup from a video game plastic guitar. The pickup can be seen in Figure 32 below.

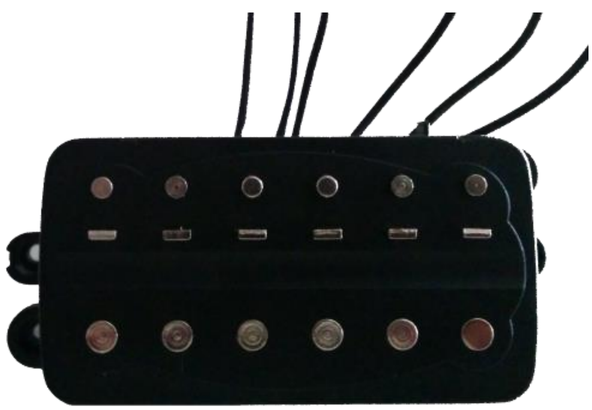


Figure 32 – Hexaphonic Pickup

4.6 Embedded Memory

Our custom PCB includes a micro SD card slot for the case that the user wants to store tablature (untethered mode) rather than just stream it over serial (tethered mode). A very high quantity of tablatures can be supported on the SD card, given that we store tablature in a condensed intermediate format that only contains the necessary information for playback on the LED matrix. In the case that the user requires an extreme quantity of stored tablature, the user need only purchase an SD card of higher capacity than the included 2GB micro SD. The microprocessor includes more than enough onboard memory to store user preferences.

4.7 Power Supply

A desired feature of the eGuitar system is portability. The goal is to not keep the user tied down with any cables while they are using the system. This means that the power supply must also be portable, instead of being plugged into a wall socket. Therefore, a single lithium polymer battery powers the main control unit as well as the LED fretboard matrix. The battery is able to recharge without any disconnection from the control board. This is achieved by including a charging circuit on the PCB designed for the eGuitar. The battery has a standard output of 3.7V, but with specific hardware, it can be boosted to have 5V output.

4.7.1 Specifications

The battery currently selected is a 1200mAh lithium polymer battery. Table 17 below shows the specifications of this battery.

Item	Specifications
Nominal Capacity	1200mAh
Nominal Voltage	3.7V
Standard Charge Current	0.2 C_5A
Max Charge Current	1 C_5A
Charge cut-off Voltage	4.2V
Impedance	< 50mΩ

Table 17 - Lithium Polymer Battery Specifications

4.7.2 Circuit Design

Figure 39 shows the schematic of the charging system that is implemented in the eGuitar.

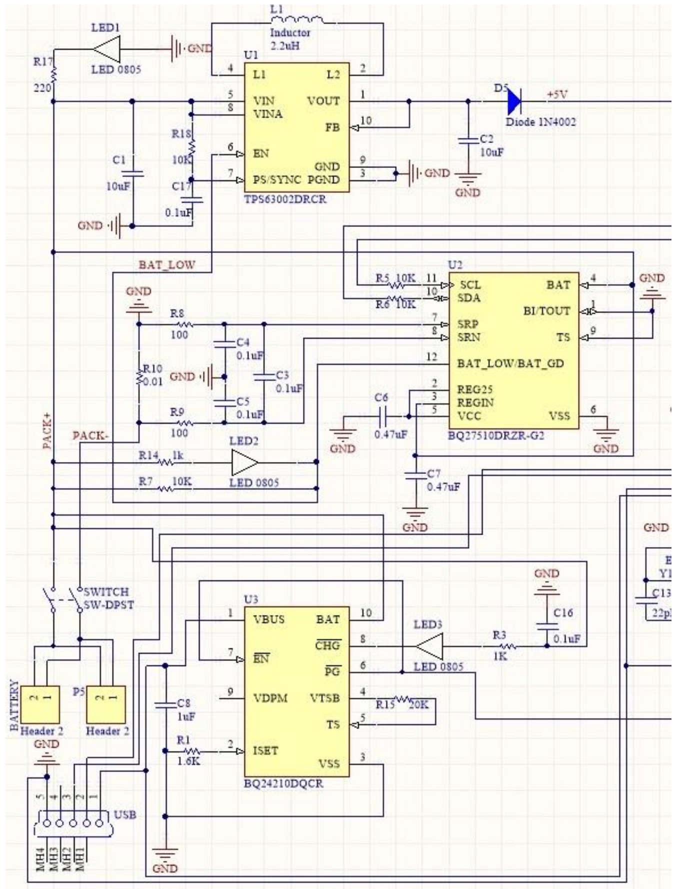


Figure 33 - Charging System Schematic

For the charging circuit, four components are integrated to provide power to the system as well as charge the battery. The specific components are listed below.

- TPS63001/2 – Buck-Boost Power Converter
- BQ27510 – Fuel Gauge Detector for battery
- BQ24210 – Lithium Ion Battery Charger
- 1200mAh Lithium Polymer Battery

The Buck-Boost Power Converter is featured in the top left of Figure 39. This component takes in voltage from the battery and boosts the input to 5V output to power the system. The Fuel Gauge Detector shown on the right side is used in conjunction with the power converter to measure remaining battery capacitance. If the battery is low, a signal is sent to the Lithium Ion Battery Charger to begin the charging sequence. The process of charging is fully detailed in Section 3.9.4.3 Charging System. Once charging completes, the Lithium Ion Battery Charger discontinues charging and waits for the battery low signal to be sent from the Fuel Gauge Detector. Figure 40 below shows the charging system connected to the main control unit.

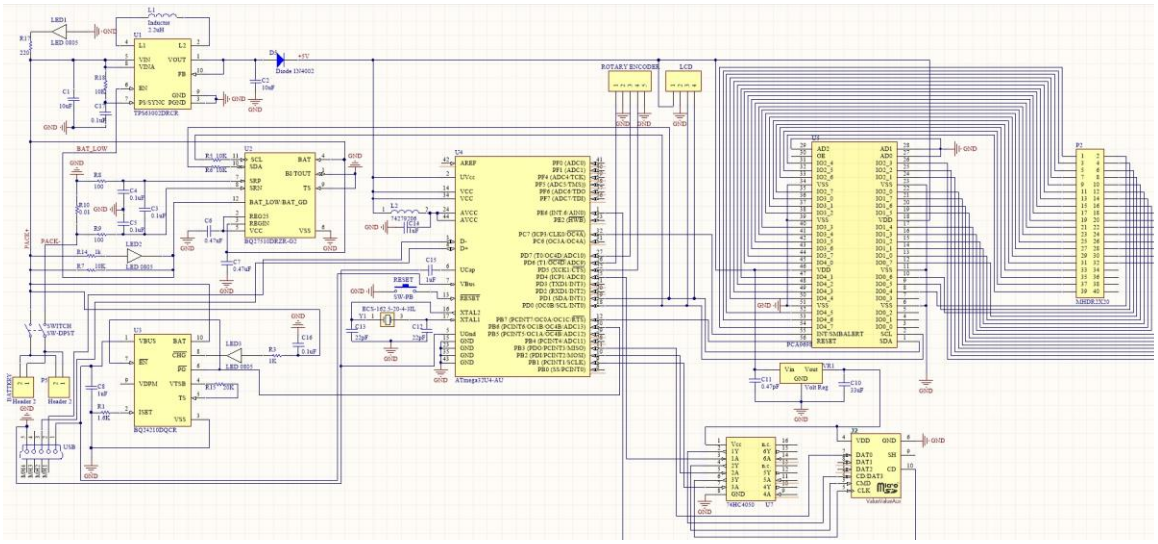


Figure 34 - Charging System Connected to Control Board

4.8 Software Architecture

The eGuitar system features a large variety of software components. In general the eGuitar's software can be broken down into two major pieces, the embedded software and windows software. These two major pieces can then be broken down into smaller pieces. Each of these components is in charge of a specific role in the eGuitar system's operation. Proper separation and communication between these components allows for an innovative and effective learning system.

4.8.1Embedded Software

The embedded component of the eGuitar system is responsible for managing all of the onboard systems. These systems range in functionality from LED control to preforming processing on the audio signal from each string, communicating with a windows computer, preforming audio playback of metronome and interfacing with the user through physical controls. In order to manage each of these subsystems the eGuitar team has created an Application Manager. The Application Manager is responsible for controlling each subsystem and sending data between them. It ensures that each component gets the data it needs, as-well as allocating processing time to each of the individual components. A breakdown of the onboard system can be seen in Figure 35 which shows the breakdown of the high level system and the connectivity between each section.

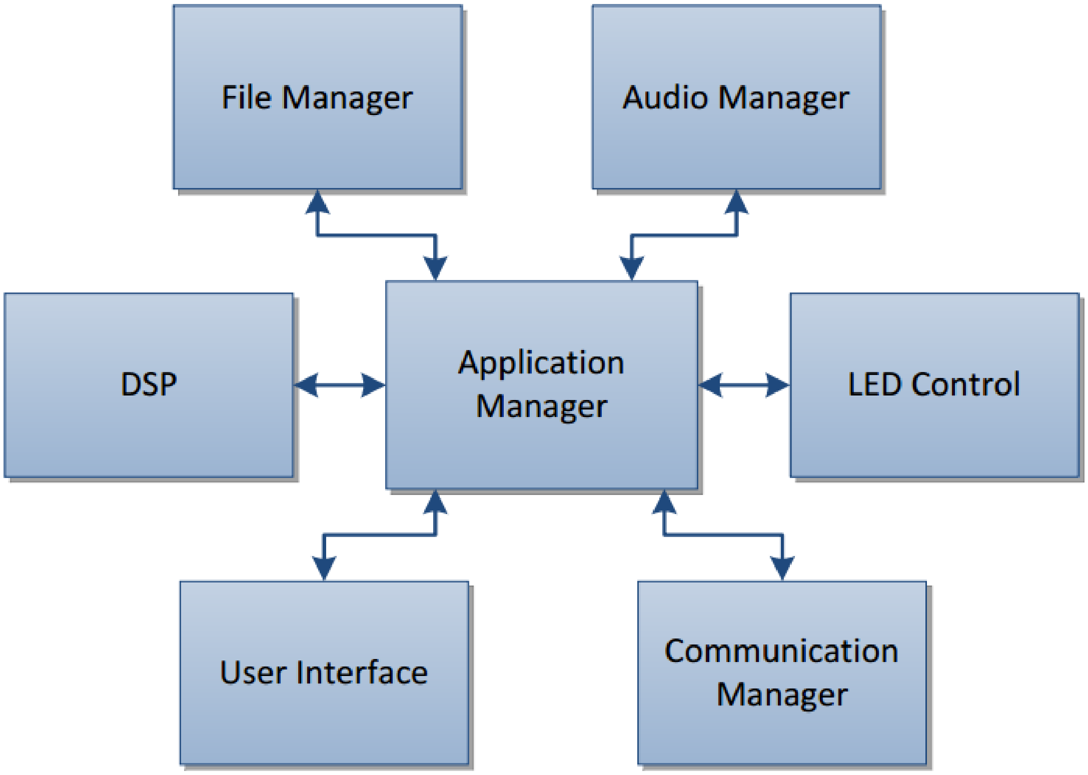


Figure 35 - Embedded Software Architecture

4.8.1.1 File Manager

The eGuitar's onboard system features a micro SD card for file storage. In order to handle the micro SD card the File Manager subsystem was created. The File Manager subsystem allows the remainder of the eGuitar system to access and store files on the micro SD card with minimal overhead. Contained within this subsystem are three classes as seen in Figure 36. These three classes are responsible for interfacing with the micro SD card, parsing through files stored on the card and exporting data structures for storage on the card. The SD Card Interface class is responsible for setting up any registers and communication protocols required to connect to the SD Card. It also handles the physical reading and writing of data to the card. Along with data read and write it handles detection of the SD card and ensure there is enough space to store a new file. The File Parser subclass is responsible for converting data returned by the SD Card Interface to usable data for the remainder of the eGuitar's subsystems. It accomplishes this by parsing through or decoding the data into usable structures and symbols. The final class contained in the File Manager is the Tab Exporter. The Tab Exporter is responsible for translating structures, created by the DSP subsystem, into user readable text files for manual manipulation or manipulation using the windows application.

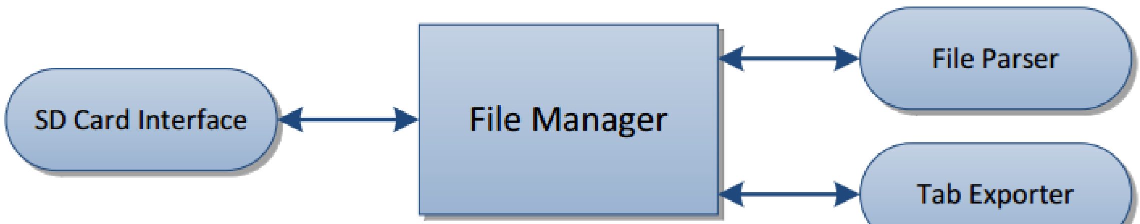


Figure 36 - Embedded File Manager

4.8.1.2DSP

The eGuitar's embedded system contains a DSP subsystem. This subsystem is responsible for handling audio input from each of the six guitar strings as well as analyzing the data and returning that analysis to the Application Manager. This subsystem has two subclasses, as seen in Figure 37, that it uses to accomplish these tasks. The first of these is the Audio Input class. This class is responsible for handling the six audio input channels and converting each set of data to a usable format for analysis. The second subclass of the DSP subsystem is the DSP Algorithm class. This class is responsible for preforming analysis on each of the six audio signals returned to the DSP subsystem by the Audio Input class. This class utilizes the hardware accelerated DSP functions built into the Texas Instruments C5535 DSP processor. The result returned by this class is a set of notes that the remainder of the eGuitar system can easily understand.

Figure 37 - Embedded DSP Manager

4.8.1.3Audio Manager

The eGuitar's embedded system has the ability to generate sound from saved tablature information or to create a metronome for the user to follow along with as they create there tablature. The Audio Manager is responsible for generation of this audio. To handle each of its task the Audio Manager is broken down into several parts as seen in Figure 38. This sound is then sent to the headphone jack or the built in speaker. In order to generate the audio waves, the Audio Manager contains the subclass Tone Generator. The Tone Generator has built in functions to convert tablature read from the micro SD card, passed in by the File Manager, or to create a metronome based upon a user definable beats per minute. The audio generated by the Tone Generator is then sent to the Audio Managers Audio Output subclass. The Audio Output class is responsible for detecting whether headphones are connected to the eGuitar. If headphones are connected it outputs the audio wave sent by the Tone Generator to the users headphones at a user customizable volume. If the Audio Output class was unable to detect headphones connected to the eGuitar system, it then uses the eGuitar's onboard speaker to play the audio.

Figure 38 - Embedded Audio Manager

4.8.1.4 User Interface

In order to facilitate ease of use, the eGuitar features a user interface to enable customization of settings and functions. In order for the eGuitar team to encapsulate this functionality the eGuitar team created the User Interface subsystem. This system has a role of handling all operations of the user interface such that the other eGuitar subsystems do not need to update their own user interface. The User Interface system ensures that each of the menus are displayed in a uniform manor as well as handle physical user inputs. Any change events that are created by the User Interface is interpreted by the Application Manager and sent to the appropriate subclass. The user Interface is comprised of two subclasses, as seen in Figure 39, Physical User Interface and OLED Display Control. The Physical User Interface class handles all input commands from devices such as buttons and potentiometers. If one of these devices changes state that new state is sent to the User Interface's main class which then interprets the device state to determine what action should be made. The OLED Display Control handles setup and management of the eGuitar's onboard OLED display. This class allows the user interface to simply send what should be displayed instead of having to manage how it is displayed on the OLED display. Along with determining how to display data, the OLED Display Control handles the brightness of the onboard OLED display and screen timeouts to save battery life.

Figure 39 - Embedded User Interface

4.8.1.5Communication Manager

Although the eGuitar is designed to work as a standalone system, the eGuitar team has elected to integrate interoperability between itself and the user's personal computer. This allowed the team to expand the eGuitar's functionality significantly. In order to manage communication between the user's personal computer and the eGuitar system, the eGuitar team created the Communication Manager subsystem as seen in Figure 40. The Communication Manager enables each subsystem of the eGuitar to communicate with the eGuitar application on the user's computer effortlessly. It accomplishes this by creating easy to use functions that accept common data types and convert them to serialized data to be sent across the USB 2.0 connection. In order to send data over the USB connection the Communication Manager performs setup of the onboard UART as well as runtime management of the UART. This include receivings data as well as transmitting data. When data is received via the UART the Communication Manager is tasked with decoding this data into structures and types that the eGuitar's other subsystems can understand. In addition the Communication Manager is tasked with converting all data types and structures sent from the Application Manager into a format that can be sent through the UART and understood by the eGuitar's personal computer application.

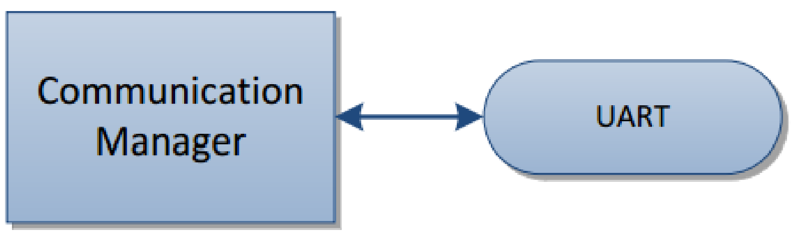


Figure 40 - Embedded Communication Manager

4.8.1.6LED Control

The eGuitar's primary goal is to aid the user in learning the basics of playing guitar. In order to achieve this goal the eGuitar features LED's along the fretboard for each of the six strings in each fret. This results in the eGuitar having to control a large quantity of LED's. In order to offload this control from the Texas Instruments C5535 DSP a Texas Instruments MSP430 has been integrated into the system. Adding in the MSP430 however causes the LED Control subsystem to span multiple devices. In order to accomplish this it was broken down into the subsystems seen in Figure 41. As shown in Figure 41, the C5535 DSP and the MSFP430 processor are operated in a master slave style relationship. The eGuitar team chose to utilize the C5535 DSP to handle most of the eGuitar's functionality due to its increased processor power and wide range of inputs. Due to this the MSP430 acts as a LED controller to manage power each of the LED's as well as determining which ones need to be turned on and off according to the message sent from the C5535 DPS. Since two Texas Instruments embedded processors are being used the eGuitar team chose to integrate an I^2C

communication system into the LED Control systems to facilitate message passing.

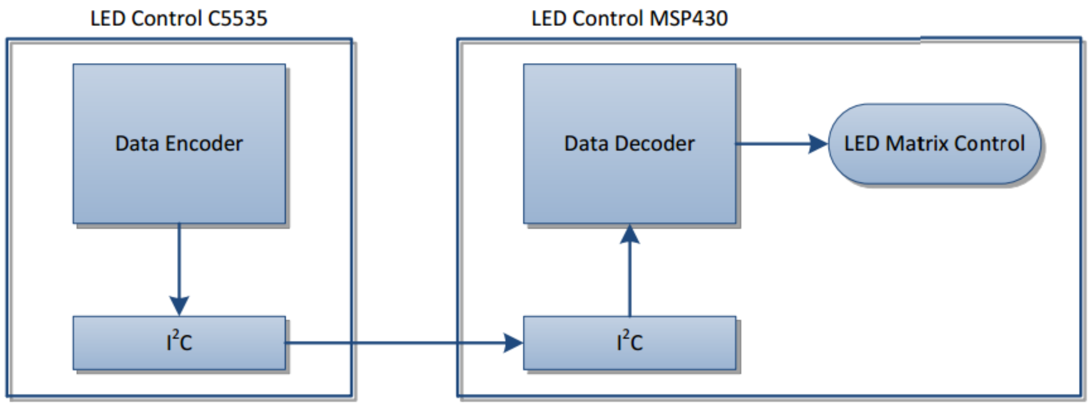


Figure 41 - Embedded LED Control

4.8.1.6.1 LED Control MSP430

The LED Control subsystem that is on the eGuitar's onboard ATmega32u4 processor is responsible for decoding I^2C messages and interpreting them to turn LED's on or off. The LED Control system on the ATmega can be broken down into three major components as seen in Figure 42. The I^2C class is responsible for all setup related to I^2C startup and shutdown. The I^2C class also handles receiving of all messages from the Host PC's LED Control subsystem which are then passed into the Data Decoder. The Data Decoder class is responsible for converting the messages received via I^2C and converting them into an understandable format for LED control. The decoded messages are then sent to the LED Matrix Control class. This class is responsible for ensuring that each LED is in the correct state and turning on or off the correct LED when a message is received.

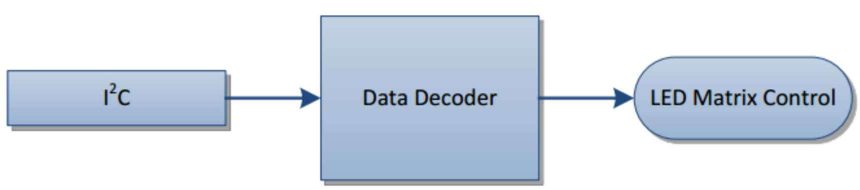


Figure 42 - MSP430 LED Control

4.8.2 Windows Software

The Windows based application of the eGuitar system is designed to enhance the eGuitar system and the users overall experience. This application enables the user to communicate with the eGuitar system during use as well as editing of tablature offline. In addition the windows application supports functionality for syncing data between the windows application and the eGuitar's micro SD card and playing back recorded tablature. The hierarchy that the eGuitar team chose also enabled the system to seamlessly connect to the embedded eGuitar system so the user can connect and disconnect it at will. In order to achieve these functions the eGuitar team broke the windows application down into separate subsystems as seen in Figure 43. This breakdown allowed the eGuitar team to create clear separations in functionality which yielded cleaner code and better overall reuse of complex functions. In order to manage each of the subclass's the eGuitar team created an Application Manager. The Application Manager is responsible for transferring data between each of the subsystem as well as managing startup, shutdown and cleanup of each subsystem.

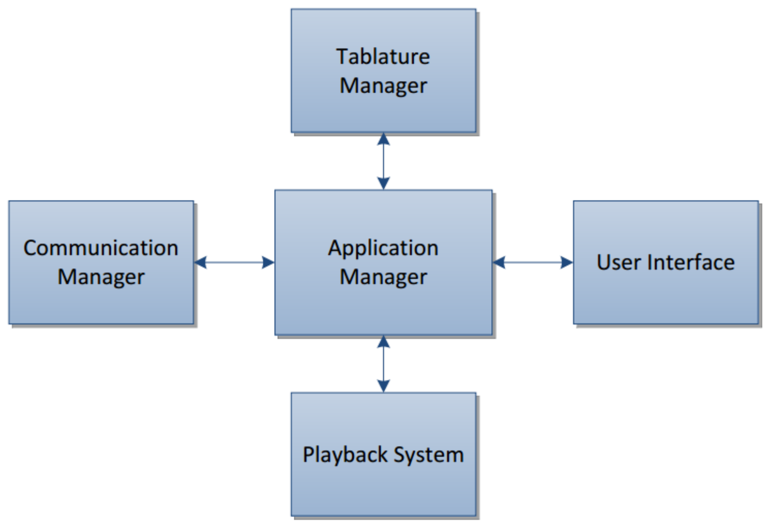


Figure 43 – Windows Software Architecture

4.8.2.1 Tablature Manager

The Tablature Manager subsystem enables the eGuitar application to import, export and edit tablature files. This subsystem is also responsible for maintaining the data structures for the active tablature document. In order to separate the individual components of this subsystem the eGuitar broke it down into the four components shown in Figure 44. The Tablature Parser class is responsible for reading tablature in from the file system and decoding it into data structures that can be understood by the Tablature Manager. This class contains functionality for importing several standard file formats in order to make the eGuitar's application compatible with other tablature applications. The Tablature Exporter class is responsible for converting the eGuitar application's data structures into standardized tablature file formats. In addition the Tablature Exporter class handles all file input and output to the user's hard drive. The final component of the Tablature Manager subclass is the Tablature Editor. The tablature Editor is responsible for editing the active tablature data structures contained within the eGuitar Tablature Manager subclass. This class takes in changes made by the User Interface subclass and ensures that any changes are valid. If the changes made by the user are valid it then makes the corresponding changes to the underlying data structure and send updates to the User Interface subclass through the Application Manager.

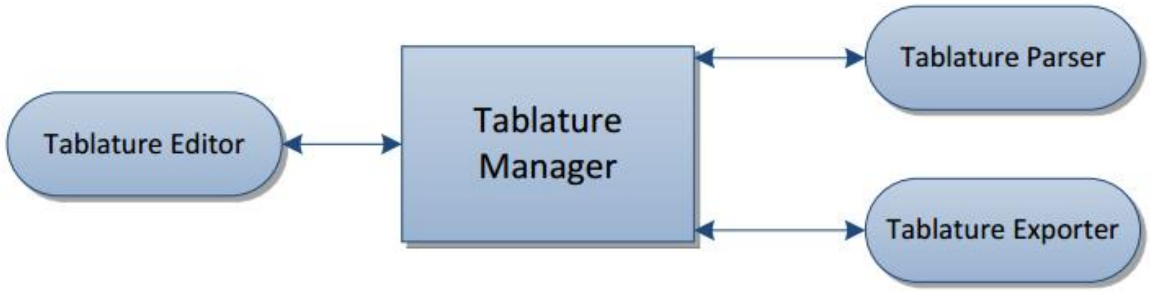


Figure 44 – Windows Tablature Manager

4.8.2.2Communication Manager

The Communication Manager subsystem of the eGuitar application enables the windows based application to seamlessly connect to the eGuitar's embedded system. The design of this subsystem can be visualized in Figure 45 where the USB 2.0 Bus is the user's personal computer hardware USB bus. In order for the Communication Manager to communicate with the eGuitar's embedded system it utilizes a USB Virtual Com Port. This allows the eGuitar application to transfer data between the embedded and windows based application using serial communication. Communication in this manor allows the eGuitar application to take full advantage of the USB 2.0 bus speeds. The Communication Manager subsystem also handles encoding tablature information for transfer to the eGuitar's onboard micro SD card as well as the transfer of messages for changing settings and passing commands to the eGuitar's embedded system.

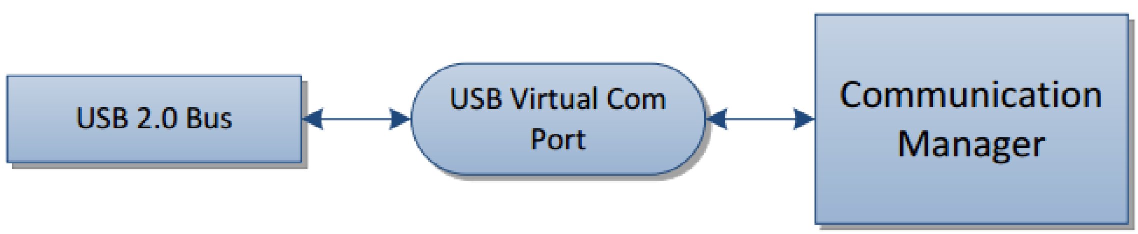


Figure 45 – Windows Communication Manager

4.8.2.3User Interface

The User Interface subsystem of the eGuitar application enables the windows based application to display information to the user and accept input from them. Due to the complexity of the User Interface the eGuitar team has elected to break it up into five separate components as seen in Figure 46. The primary class of the subsystem, the User Interface class, is responsible for displaying the primary interface as well as handling input from the user. This includes all button clicks, mouse clicks, and mouse drags. This class is written in such a way that it is expandable and the eGuitar team can develop plugins to increase the eGuitar's robustness. The eGuitar team included four plugins that cannot be removed or changed by the end user. These plugins are a Recording Interface, Tab Visualizer, Tab Editor and a Generic Interface. The Recording Interface allows the user to control the tablature recording settings of the eGuitar's embedded processor. This interface also enables control for starting and stopping recording of tablature information. The Tab Visualizer interface is responsible for parsing and displaying tablature information to the user. This interface also enables the user to visualize tablature information as it is being recorded on the eGuitar's embedded system. The Tab Editor interface enables the user to modify tablature information in real time after recording or from tablature data loaded in from a file. This interface creates a powerful set of tools that the user can use to modify the underlying complex tablature information, enabling them to fix errors in the recording or modify the song they have created. Finally the Generic Interface provides a wide range of interface features to the user. It is responsible for several menu systems and options such as importing, exporting, saving of the tablature files. This interface also allows the user to browse the files stored on the eGuitar's embedded micro SD card. In addition to browsing the micro SD cards contents the user is able to sync additional files to the micro SD card and delete files as well.

Figure 46 – Windows User Interface

4.8.2.4Playback System

The Playback System subsystem of the eGuitar application enables the user to listen to tablature created by the user or imported tablature created by another tablature application. In order for the eGuitar team to create this system they broke it down into three major components as seen in Figure 47. The Tab Decoder component of the Playback System allows the eGuitar application to decode tablature information from its internal data structures into audio waves. The Tab Decoder system is built in such a way that it can be expanded by end users to represent different instruments. The Audio Output component of the Playback System handles physical output through the windows computer. It takes the audio wave generated by the Tab Decoder and create a stereo output that is sent to the computers speakers or headphone jack. This component allows the user to adjust volume independently from the computers global volume level. The final component is the Playback Systems control class. This component handles message passing between the Tab Decoder component and the Audio Output component. In addition the Playback System control class handles all data communication between the Application Manager subsystems to transfer data to the other subsystems.

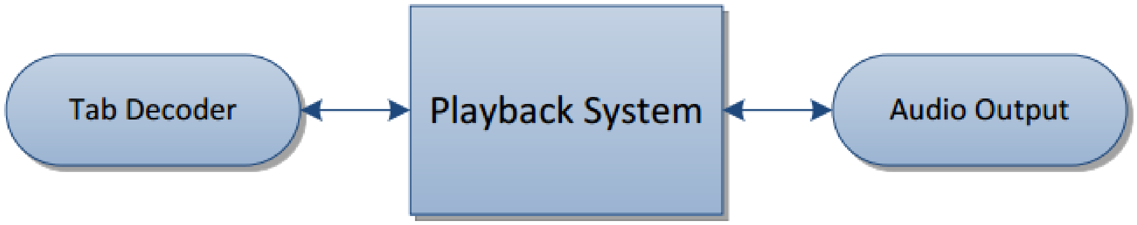


Figure 47 – Windows Playback System

4.9 Computer Interface

4.9.1USB Interconnectivity

The eGuitar system is designed to be used anywhere. However, it supports connectivity between a windows computer and the eGuitar embedded system. Using the eGuitar system in this manner enables the user to charge the eGuitar battery while using the system or to actively visualize the automated tablature system in action. While the eGuitar is connected to a windows computer the user is able to transfer files between the eGuitar's built in micro SD card and the computer.

4.9.1.1Embedded Hardware

To facilitate communication between the windows computer and the eGuitar system, the hardware needed to support communication between the embedded systems components and the windows computer software interface. The basic hardware design can be seen in Figure 48 which details the overall flow of communication between components in the eGuitar system when operating in tethered mode. To facilitate communication with the windows computer UART that is built into the C5535 DSP board was utilized. This UART is configured to operate as a USB Virtual Communication Port. With the UART configured in this manor the windows computer is able to see it as a communication device with which we can transfer data using USB 2.0 style serial communication. This enables rapid transfer of data between the devices. In addition the software enables the eGuitar's windows software to visualize and modify files stored on the micro SD card. Using the C5535 DSP chip the windows computer can access the micro SD card through the built in UART and the C5535 DSP chip's built in SD card interface module.

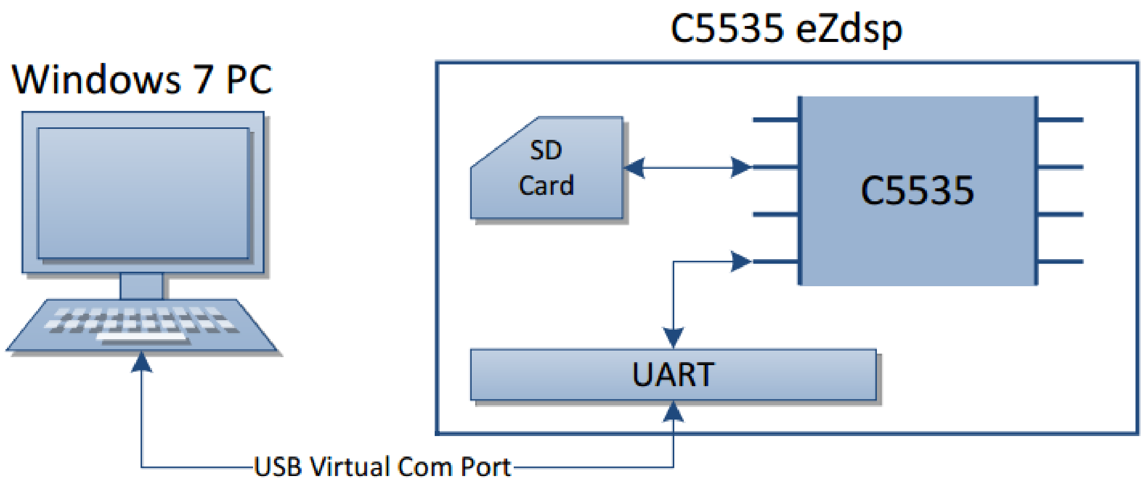


Figure 48 - Computer to Embedded Interface

4.9.1.2 Windows Software

The eGuitar windows application has been designed by the eGuitar team to support connectivity with the eGuitar embedded hardware. This enables advanced features of the eGuitar system as well as syncing of file to the onboard micro SD card. When the eGuitar user connects the eGuitar to a windows based computer the windows application can detect its presence automatically. Once the device has been connected the windows software enables hidden controls that facilitate control over the embedded eGuitar software. These controls allow for manipulation of the eGuitar's settings through an easy to use graphical user interface.

4.9.1.2.1 Learning

When the eGuitar is in learning mode, additional controls are made available to the user such as play, pause, rewind and stop. These controls allow the user to control playback of the LED's on the fretboard remotely such that they can move through the song and rewind at their own pace. In addition, the notes that they are currently playing according to the fretboard LEDs are displayed in the windows application for them to follow along. This interface also enables the user to choose songs or tablature that they wish to learn through the window application and make the correct changes to the eGuitar's embedded software.

4.9.1.2.2 Recording

When the eGuitar is in recording mode additional controls are made available to the user such as record, pause and stop. These controls allow the user to control the current tablature recording being done by the eGuitar's embedded DSP chip. The user is also able to visualize the tablature being created in real time enabling them to check for errors or make changes immediately. The tablature that is created can be stored on the eGuitar's embedded micro SD card as well as in the windows application. The eGuitar's windows application enables the user to edit the tablature being created in real-time during recording. Due to this, the windows application automaticity syncs any tablature changes to the eGuitar's embedded micro SD card such that no data is out of sync.

4.9.2Tablature Parser

The eGuitar team has elected to include a tablature parser into the eGuitar windows application. This parser enables to user to import tablature that is in the Power Tab format. Power Tab is free open source tablature creation software that is popular among home musicians. Due to Power Tab's popularity among musicians the eGuitar team chose to make it there default file format. This parser allows the eGuitar user to import the files into the eGuitar's windows application for viewing, manipulation and transfer to the eGuitar's embedded micro SD card. The eGuitar team recognizes however that additional tablature formats are used by musicians. Due to this the eGuitar team has elected to write the eGuitar's windows application in such a way that additional parsers can be written as plugins to the eGuitar's windows application. This enables the eGuitar Windows application to be expandable by anyone and support new formats as they are created in the future

5 Design Summary

As a convenience, this section covers the high-level design decisions for each subcomponent of the eGuitar system. For a more detailed analysis of the design of each component, please refer to Section 4 – Design.

5.1 Hardware

The following sections summarize the final decisions for the hardware design. The overall goal for the hardware on the eGuitar system is to be unobtrusive and efficient.

5.1.1LED Fretboard Matrix

The eGuitar system is designed to teach the user how to play the guitar. Beginners typically only play on the first three or four frets on a guitar, so the eGuitar system covers up to five frets on a guitar to support the user's needs. The hardware of the eGuitar system includes the LED fretboard matrix and main control unit. The LED fretboard matrix is comprised of 6 LEDs per fret on the guitar. Five frets are supported as well as an open string. Open means the user does not press down on a string but still plays it. Therefore, a total of 36 LEDs are implemented for the user to learn a variety of notes and chords on a guitar. The LED fretboard matrix was designed to be placed underneath the strings of the guitar. The six individual PCBs housing 6 LEDs each are connected to a flat ribbon cable that leads down the neck of the guitar to the main control unit. Solder pads are designed on the bottom of each PCB for the ribbon cable to easily solder onto. The LED matrix PCB material is a custom 0.5mm 1-layer FR4 in order to fit the size requirements. 4PCB or Advanced Circuits was the PCB manufacturer for the eGuitar's main control unit and Saturn PCB was the LED fretboard matrix manufacturer.

5.1.2Control System

The main control unit features a custom designed LED control unit and custom designed charging unit. The LED control unit is comprised of an ATmega32u4 microcontroller and a 40-bit input output expander. The ATmega acts as a serial slave to the host PC. Once a signal is sent from the PC application, the ATmega32u4 sends out control signals to each of the 36 LEDs. A lithium ion battery powers the entire system. The battery has a charging unit implemented in the PCB design. The charging unit implements three integrated circuits to accurately and safely charge the lithium polymer battery. Those three ICs are the TPS63001/2 Buck-Boost Power Converter, the BQ27510 Fuel Gauge Detector, and the BQ24210 Lithium Ion Battery Charger. The Buck-Boost Power Converter boosts the 3.7V from the lithium polymer battery to 5V, while the Fuel Gauge Detector monitors the battery level. Once the battery is low, the Lithium Ion Battery Charger begins the charging process. The schematic drawings in Section 4.7.2 show how the charging system integrates into the LED control system. The total control unit has both LED control and charging system on one PCB. This PCB uses the standard FR4 material. The PCB only needs a maximum of two layers for appropriate board layout.

5.1.3User Interface Hardware

The onboard eGuitar system is housed in a custom-made 3D printed enclosure. As this control box has many user controls, it is considered user interface hardware. The aim of this control box is that all user options and functions be navigable with minimal and intuitive hardware controls.

The user-facing controls include the following input/output devices:

- A 128x64 pixel monochrome OLED display
- A push-button rotary encoder for option navigation and selection
- A power switch to engage or disengage power
- A Mini-B USB female port for charging and PC connectivity
- An SD card slot
- A reset button

5.1.4Guitar Pickup

The guitar pickup utilized by the eGuitar system is of a custom design by the eGuitar team. This pickup, called a hexaphonic pickup, differs from standard magnetic guitar pickups since it features an output line for each of the strings on the guitar. To achieve this, the custom pickup utilizes six individual electromagnetic transducers. Each transducer is created by reusing the electromagnets contained within electromagnetic relay switches. In addition this custom pickup is stretched out compared to a traditional pickup. This feature allows the eGuitar to minimize interference and noise created by the neighboring transducers. This design enables the eGuitar team to create a low cost and high performance guitar pickup.

5.2 Software

The summary of design decision for the software portion of the eGuitar system is detailed here. Overall goals of any software developed for this system include maintainability, legibility, efficiency and performance.

5.2.1User Interface Software

The embedded user interface software design involves generalized layouts for various operation modes, settings and options. The display libraries available accommodate text, rectangles and even individual pixel control. While individual pixel control allows for virtually limitless possibilities, the UI design is limited to simple, legible and intuitive layouts so that even a small display can present relevant information.

5.2.2Tablature Parser

The eGuitar's integrated tablature parser is responsible for importing tablature from other tablature information. The initial parser only supports importing tablature files created by Power Tab, since it is one of the most popular tablature creation software applications. In order to ensure the eGuitar is compatible with other tablature formats its parsing application allows custom plugins to be created that integrate seamlessly with its software.

5.2.3LED Matrix Controller

The eGuitar's onboard hardware is responsible for control of the LED fretboard in addition to its storage and audio processing responsibilities. In order to offload the control of the LED's from the C5535 DSP chip an MSP430 is integrated into the system. The eGuitar team thus utilizes two sets of software to manage these components, one on the C5535 DSP and another on the MSP430.

5.2.3.1C5535 DSP

The eGuitar's integrated C5535 DSP software incorporates the functionality to parse the stored tablature information and transfer of the LED control data to the MSP430. The software achieves this by reading the tablature data from the integrated micro SD card. This data is then be decoded by the software into tablature structures that the software can easily traverse. The C5535 software then decodes the current chord for that moment in time for transmission to the MSP430. To transfer this data to the MSP430 the software utilizes the hardware

I^2C interface. This interface is used to transmit the chord information as serialized strings which are then decoded on the MSP430.

5.2.3.2 ATmega32u4

The eGuitar's integrated ATmega chip contains software for receiving and decoding of chord information. The software then transforms the decoded chord into LED on off commands to be sent to the ATmega's IO expander. In order to achieve this goal the software has to read messages received as serialized data by the I^2C bus. This data is then be decoded and translated into a form that can be sent to the IO expander over I^2C. The IO expander software then translates this data into hardware on off states which controls the LEDs.

6 Design Constraints

The eGuitar system encountered a number of design constraints that included economic and manufacturing difficulties. These limitations are detailed below in their individual sections.

6.1 Economic

For the eGuitar system's LED fretboard PCBs, the original material chosen to use was flexboard. After many talks with different PCB manufacturers, using the flex material would double or even triple the cost of the project. With our budget being set below $1000, the flex material could not be used. An alternative was selected, however, and 1 layer FR4 was used to still meet the design goals for the LED fretboards. Another economic design constraint was the hexaphonic pickup used for the digital signal processing. While this type of pickup exists and is made as a consumer product, the pricing of the third-party pickups also was greater than our budget allowed. Luckily, after considerable research, the eGuitar team found a hexaphonic pickup within a video game guitar controller.

6.2 Manufacturability

The eGuitar also has some manufacturing constraints when it comes to the 3D printed enclosure. During final assembly, the enclosure design was slightly smaller than the contents that needed to fit in it. Therefore, the final enclosure needed to be hot glued together to keep all contents in place. For manufacturability, the final enclosure design would not be ideal to mass produce. Also, the eGuitar is meant to fit on any guitar, but the sizing of the LED fretboard PCBs requires some redesigning to have mass manufacturing of the parts. Also the LED PCBs are held in place by double sided tape in its current iteration, and a clasping system would need to be design for a consumer product.

7 Project Prototype Construction

The eGuitar system's prototyping phases involved that iteratively more functional builds. Ultimately, a coherent system was formed from knowledge gained in the construction and testing of prototyped subsystems.

7.1 Hardware

This section details the construction of prototypes for each hardware subsystem of the eGuitar system. Efforts have been made to foster independently functioning systems that can simplify the final implementation of subsystem interaction.

7.1.1Control System

The main control unit of the eGuitar system contains a standard microcontroller, battery charging circuit, and on-board memory. The final designs of the control unit can be found in Section 4.1 - PCB and in Section 5.1 - Hardware Design Summary. Prior to the final hardware being implemented, a prototype stage was used to test design ideas. These ideas can range from LED control over input output expanders to communication between controllers. The following sections detail the hardware used in the prototyping stage as well as how all of the components communicate for testing purposes.

7.1.1.1Control System Prototype Hardware

During the prototype stage, custom PCBs were not used to test the individual components. Instead, the prototype hardware included some of the components that the final PCB is designed off of. For example, the custom PCB includes an ATmega32u4 microcontroller as well as a charging system, but for prototyping purposes, an Arduino Mega and TI Fuel Tank BoosterPack was tested instead. The Arduino Mega features 54 I/O pins, a mini-USB port, and push buttons for reset. The Fuel Tank BoosterPack includes a lithium polymer battery as well as a charging PCB. The pickup used for prototyping was the same as the final implementation. A salvaged hexaphonic pickup was tested for all digital signal processing.

Once testing using the prototype boards completed, a second stage of prototyping began with breadboard usage. In this second phase, the integrated circuits that are included in the final PCB design were placed on a large breadboard to test the individual components. This step is to ensure that all of the components work together before a final PCB is designed and ordered. In order to use a breadboard, the ICs must have through-hole pins. If certain ICs only come in surface-mount variants, than an adapter can be used. For example, the charging circuit ICs are all leadless surface mount chips. The company Proto-Advantage sells breakout boards that can convert surface mount chips to breadboard compatible devices. The ICs included in the prototype breadboard can be found in Section 4.1.1.2 – Hardware Configuration. In Figure 50 below, it shows the Arduino Mega, an SD card shield, and a large breadboard being used to test all the components in the control board. What is not shown is the LED PCBs connected to the 40-bit I/O expander due to the messiness of them. The three green squares in the middle of the breadboard are the three ICs used in the charging circuit.

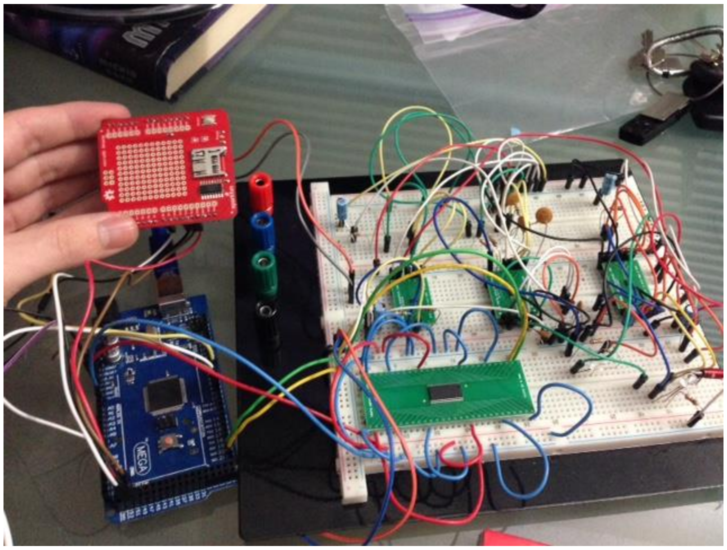


Figure 50 – Prototype Control Board

7.1.1.2Control System Connectivity

For the first prototype stage, the Arduino Mega is connected to the host PC. The Fuel Tank BoosterPack is designed to fit on top of the MSP430 LaunchPad, so the shield was connected via jumper cables. To individually test the LED fretboard matrix, the Arduino Mega receives serial data from the PC and translate the data into I^2C to control the I/O Expander. The OLED screen and rotary encoder were on a separate breadboard for their own individual testing as shown in Figure 51.

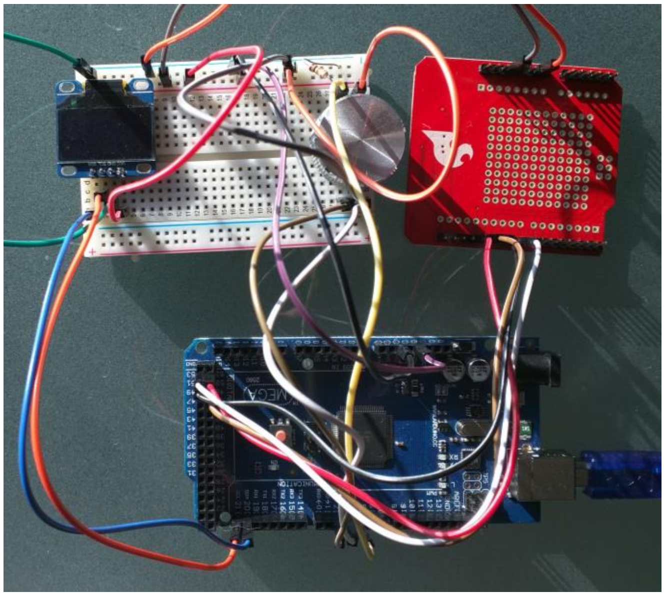


Figure 51 – OLED and Rotary Encoder prototyping

7.1.2LED Fretboard Matrix

In the final design of the LED fretboard matrix, six individual PCBs house six LEDs that correspond to each string per fret on a guitar. The PCB is required to fit underneath the strings and also not interfere with the user's ability to play. For prototyping purposes, the sizing and form factor do not need to meet these requirements. The following sections detail how the LED fretboard matrix was implemented in the prototype stage.

7.1.2.1LED Prototype Hardware

Since the prototype hardware can bypass most requirements of the fretboard matrix, the LEDs used were standard 5mm LEDs that can be typically found in beginner electronics kits. These were placed on a breadboard that connected to the Arduino Mega. Also included on the breadboard were the appropriate resistors to maintain the correct current to the LEDs. The LEDs were powered by the Fuel Tank BoosterPack, which also connected to the breadboard. For the first prototype stage, a small subsection of 12 LEDs were controlled and tested. The second prototype stage included all of the ICs mounted onto a breadboard. When this stage occurred, 36 LEDs were used to test the full range of the eGuitar system. Lastly, the final build LED PCBs arrived early, so prototyping the LED hardware was mostly with the actual LED boards.

7.1.2.2LED Connectivity

During the first prototype stage, the Arduino Mega controlled 12 LEDs using its I/O pins. Once the 40-bit I/O expander was implemented in the second prototype stage, then all 36 LEDs were connected to that hardware. The Mega then switched to controlling the I/O expander, rather than the LEDs themselves.

Power was connected to the breadboard for both prototype stages. The ground pin from the Mega was used as the main ground line that connected to the breadboard and all of the LEDs.

7.1.3 Individual pickup

In order for the eGuitar team to properly test their guitar pickup design a prototype was constructed. The eGuitar team chose to make their prototype pickup simpler in order to speed up debugging and the build process. This pickup was designed to only pickup audio from a single guitar string. This allowed the eGuitar team to ensure they are getting the proper frequency response and their coil design is feasible. To create this prototype a single electromagnet was removed from a magnetic relay. The eGuitar team then soldered ground and output lines onto the pickup for testing. The ground and output line was then connected to a standard quarter inch male stereo jack. The pickup was then placed into a housing constructed of 3D printed ABS plastic to mimic the final pickup housing. In order to place the pickup on the guitar the eGuitar team simply used double sided tape as this prototype did not need to be easily reusable.

Once the final salvaged pickup was received, it was connected to six USB sound cards and the prototype shown in Figure 53 was the first iteration. The second prototype of the hexaphonic pickup included custom 3D printed brackets that clamped onto the sound hole. Figure 52 below shows the design of these clamps, while Figure 54 shows the pickup in the sound hole of an acoustic guitar.

Figure 52 – Clamping Mechanism for pickup

Figure 53 – First Hex Pickup Prototype

Figure 54 – Hex Pickup with custom clamps

7.1.4 OLED Display Control

Prototyping of the OLED display control hardware simply involved connecting the OLED display to the Arduino Mega on a breadboard. As the OLED display supports the I^2C and SPI standards, only ground, power, SDA and SCL lines needed to be connected. Proper performance of the display yielded rendering of various shapes, text, and individual pixel control.

7.2 Software

This section details the construction of prototypes for each software subsystem of the eGuitar system. Efforts have been made to foster independently functioning systems that can simplify the final implementation of subsystem interaction.

7.2.1Tablature Parser

The eGuitar team created a prototype parser to ensure that Power Tab files can be easily read in and interpreted. This prototype was written using C# due to its powerful string manipulation functions as well as its easy to use file IO functions. The initial prototype created by the eGuitar team did not support plugins or expansion by other developers. This feature was added in the final implementation. In addition, this parser is limited in its feature set as it is incapable of translating many complex guitar tabs into usable data structures. These complex guitar tabs are added after the initial prototype is proved to work and the eGuitar team is comfortable with moving onto the next phase.

7.2.2LED Control

Since the eGuitar team created two prototype versions of the LED fretboard control system two software prototypes were created. Phase one consists of software to control only twelve LED’s on the MSP430. Phase two is closer to the complete eGuitar system, which utilizes thirty six LEDs connected with an IO expander.

7.2.2.1Phase One

Phase one of the eGuitar’s LED control software focuses primarily on control of a minimal number of LEDs. The software is written for an MSP430 Launchpad development board. Since the LED’s are connected directly to the MSP430’s IO pins, the software focuses solely on user defined control of each pin independently. The software achieves this by manually setting each pins output voltage to high or low accordingly.

7.2.2.2Phase Two

Phase two of the eGuitar’s LED control software is considerably more advanced than phase one and resemble the final system software. This phase focuses on control of thirty-six LEDs via an IO expander. The hardware utilizes an IO expander controlled through I^2C; this in turn means that the software for phase two needs to be capable of communicating through I^2C. The software for phase two also needs to be capable of indexing the thirty-six LEDs correctly and encode the data for which LEDs are on and off into a format that can be transferred to the IO expander through I^2C.

7.2.3 OLED Display Control

Prototyping the OLED display control software simply involves the usage of available libraries tailored to 128x64 pixel displays. These libraries leverage the I^2C or SPI standards to draw images, text, simple shapes, or even allow individual pixel control. The functionalities relevant to the eGuitar system are text and shape rendering. This involves specific library function calls specifying on-screen position, font or line size, and other visual-tweaking parameters.

7.2.4 Digital Signal Processor

Prototyping of the C5535 ezDSP digital signal processor chip software simply involved the usage of available libraries in detecting pitch from as many channels as possible. While single-channel pitch detection is sufficient for tuning the guitar one string at a time, tablature transcription requires 6-note polyphony. The first-stage prototype focused on single-string note detection while the second-stage prototype, utilizing DSP on Windows, was tested alongside the custom polyphonic pickup in attempts to detect simultaneous notes across strings.

8 Project Prototype Testing

The eGuitar system sought to thoroughly test its iterative prototyping phases that involve increasingly functional systems. Ultimately, a coherent system was formed from knowledge gained in the construction and testing of prototyped subsystems. In this section, the various testing procedures are detailed in easy-to follow formats with an explicit procedure and pass/fail expected result for each test.

8.1 Digital Signal Processor

Table 18 below details the test procedures for the Digital Signal Processor prototype.

Test	Procedure	Pass	Fail
Single Note Detection	Pluck one string in a known location	The correct note is identified	No note or incorrect note is detected
Multiple Note Detection	Strum a chord composed of known notes	Each note within the chord is simultaneously identified	No notes or incorrect set of notes identified

Table 18 - Digital Signal Processor Test Procedures

8.2 LED Matrix

The LED fretboard matrix is comprised of 36 individual LEDs that each have a 680Ω resistor attached. An ATmega32u4 chip that sends commands to a 40-bit I/O expander controls the LED matrix. Table 19 are the test procedures for the hardware aspect of the LED fretboard matrix.

Test	Procedure	Pass	Fail
LED Functionality	Connect one LED in series with a resistor to the MSP430 and ground. Send HIGH signal to pin.	LED turns on	LED stays off
LED Input Current	Measure input current to the LED after resistor	30mA input current	>30mA input current
40-Bit I/O Expander Output Commands	Send I^2C message from MSP430 to I/O Expander. Measure output voltage at desired output pin.	Desired output pin is HIGH	Desired output pin is LOW
40-Bit I/O Expander Output Voltage	Measure the output voltage from each pin that HIGH signal is sent to	~5V Output Voltage Measured	<5V Output Voltage Measured
LED Brightness	Observe LED HIGH status and determine brightness level with personal reaction to it	Brightness level is not disturbing and viewable	Brightness level is disturbing and hurts the eye

Table 19 - LED Matrix Test Procedures

8.3 Power Supply

The power supply features both a lithium polymer battery and a charging circuit. The charging portion of the power supply needed to effectively monitor battery levels and be able to safely charge the battery. In order to achieve these requirements, three components works together to form the charging unit. The schematics and ICs are detailed in Section 4.7 - Power Supply. For prototype testing a digital multimeter was used for some procedures. Table 20 below lists the tests that were performed on the power supply.

Test	Procedure	Pass	Fail

Lithium Polymer Battery Output Voltage	Measure output voltage of Lithium Polymer Battery	3.7V Output measured	<3.7V Output Measured
Buck-Boost Power Converter Input Voltage	Measure input voltage from lithium polymer battery to Buck-Boost Power Converter	3.7V Output measured	<3.7V Output Measured
Buck-Boost Power Converter Output Voltage	Measure output voltage to see if IC is bucking or boosting power. 3.7V Lithium Polymer Battery should be boosted to 5V.	5V Output Measured	<5V Output Measured
Fuel Gauge Detector - LED Indicator	Observe connected LED to Fuel Gauge when battery power decreases. If power drops below threshold, LED turns on.	LED Indicator goes HIGH	LED Indicator stays LOW
Fuel Gauge Detector – Function	Measure lithium polymer battery capacitance, then measure Fuel Gauge Detector output to indicate whether IC shows low battery.	Signals battery low when capacitance is actually low	Does nothing Signals battery low when not low
Lithium Ion Battery Charger	Measure output voltage when battery is low to indicate a charge is occurring.	Charge is taking place	No output voltage measured
Lithium Ion Battery Charger	Measure battery power after charge has occurred	Battery power levels increased after charge occurred	Battery power stayed the same or lowered after charge
Lithium Ion Battery Charger LED Indicator	Observe connected LED when a charge is taking place.	LED goes HIGH when charging	LED goes HIGH when not charging / LED stays LOW

Table 20 - Power Supply Test Procedures

8.4 OLED User Interface

Table 21 below details the test procedures for the OLED user interface prototype.

Test	Procedure	Pass	Fail
Text Drawing	Run software routine for text rendering	Only input text is drawn and observed	Incorrect text or unexpected visual artifacts
Shape Drawing	Run software routine for rectangle rendering	Only input shape is drawn and observed	Incorrect shape or unexpected visual artifacts
Individual Pixel Manipulation	Run software routine for individual pixel manipulation	Only input pixel set is drawn and observed	Incorrect pixels drawn or unexpected visual artifacts

Table 21 - OLED User Interface Test Procedures

8.5 Tablature Parser

In order for the eGuitar team to properly test the tablature parsing software a sample tablature file was required. The eGuitar team used several different tablature files in order to test components one at a time. These files ranged from a single note in the tab to six different notes being played at once and multiple tabs. Table 22 below details these procedures.

Test	Procedure	Pass	Fail
Opening Test	Open Power Tab file	ASCII data is printed to the screen	Application crashes or returns file not found.
Single Note and tab	Open single note Power Tab file	The note is printed to the screen with the correct time code	Incorrect time code or note is printed to the screen
Double note and single tab	Open Power Tab file containing two notes in one tab	The notes are printed to the screen with the correct time code	Incorrect time codes or notes are printed to the screen
Single note and two tabs	Open Power Tab file containing two tabs with single notes	The notes are printed to the screen with the correct time code	Incorrect time code or note is printed to the screen
Double note and two tabs	Open Power Tab file containing two notes and two tabs	The notes are printed to the screen with the correct time code	Incorrect time codes or notes are printed to the screen

Six notes and one tab	Open Power Tab file containing six notes and one tab	The notes are printed to the screen with the correct time code	Incorrect time codes or notes are printed to the screen
Six notes and six tabs	Open Power Tab file containing six notes for each of the six tabs	The notes are printed to the screen with the correct time code	Incorrect time codes or notes are printed to the screen

Table 22 - Tablature Parser Test Procedures

8.6 Pickup

In order for the eGuitar team to properly test the custom guitar pickup the following tests were performed. These tests required the use of a standard guitar amplifier and a multimeter for them to be completed properly. Table 23 below details these procedures.

Test	Procedure	Pass	Fail
Connection	Plug pickup into guitar amplifier	Audible feedback is not heard	Significant feedback is heard
Single Note	Pluck the string in a known location	The correct note is heard	No note or incorrect note is heard
Incorrect string	Pluck a string other than the one for the pickup	No audio is heard	Audio is heard
Voltage test	Pluck the string above the pickup	A voltage of 100mv to 2 volts is detected	Voltage reading is < 100mv or greater than 2v

Table 23 – Guitar Pickup Test Procedures

9 Project Operation

This section details the functionalities of the final eGuitar system. The beginner functionalities are based on the LED Fretboard, while more advanced users will focus more on the hexaphonic pickup elements.

9.1 LED Fretboard

The LED Fretboard consists of 6 individual PCBs that correspond to the first five frets on a guitar and open note playing. Beginner users can download any PowerTab guitar tablature and load it onto the provided SD card, or upload the tab using the windows application. Once the selected song has been loaded, the user can press the rotary encoder on the control box attached to the guitar to begin playback. The LED Fretboards light up with the correct finger placements for the song selected. If the playback is too slow or too fast, the user can rotate the rotary encoder to adjust the speed of playback. The user may also change the beats per minute (BPM) on the windows application.

9.2 Hexaphonic Pickup

For advanced users, they can select the guitar tuner tab on the windows application to tune their instrument, six strings at a time. Once their guitar is tuned, the hexaphonic pickup is able to record each individual string being played, and can record every note and fret used. With this recording, the user can look back and find the sections of song that they enjoyed and remember it. An example of both systems in use is a teacher/student situation. A guitar teacher may use the eGuitar system to record a basic lesson for their student using the hexaphonic pickup. The teacher loads the recorded tablature into the control board’s SD card and then hand the system over to their student to practice. The student then plays back the teacher’s recorded lesson and is shown the finger placements and tempo on the LED fretboard. The final system can be seen in Figure 55 below.

Figure 55 – First Hex Pickup Prototype

10 Administrative Content

10.1 Project Budget and Financing

As this project was not sponsored, the group attempted to collect sample parts where possible, such as the hexaphonic pickup. Table 24 below lists the total cost for each component.

Item Description	**Cost**
F.O. filament, wire, proto board, glue gun	$23.00
LilyPad LED White/Blue	$11.83
Pizza	$15.00
TMDX5535EZDSP (TI ezDSP chip)	$115.53
2xTSSOP-56 to DIP Adapter	$29.98
TSSOP-20 Breakout / Tweezers / Help Hands / Male headers	$28.58
I/O Expander / MSP430G2553 / LEDs	$48.00
Flux Pen	$19.84
Radxa Rock Light	$69.99
8x USB sound card	$66.1
3x 568-1455-5-ND Analog Mux	$6.53
2x usb 2.0 slim hub	$11.98
Edimax EW-7811Un	$8.99
5V 2A microUSB wall charger	$7.99
2x 8GB microSDHC card Class 10	$13.98
ODROID-C1	$36.95
LED Boards - China Build	$428.62
Control Board	$153.11
SD Card Stuf	$26.94
Components for PCBs	$182.45
Ribbon Cable	$36.75
Total Cost	$1342.14

Table 24 - eGuitar Budget

10.2 Project Milestones

The biggest success of the eGuitar project was the time management exhibited by the group. The project milestones shown below in Table 25 show the features that were planned for implementation. By the end of the second semester, the eGuitar functioned well enough to allow for the user experience mentioned in Project Function. Table 25 below details the group's minimum and ideal milestones for both semesters.

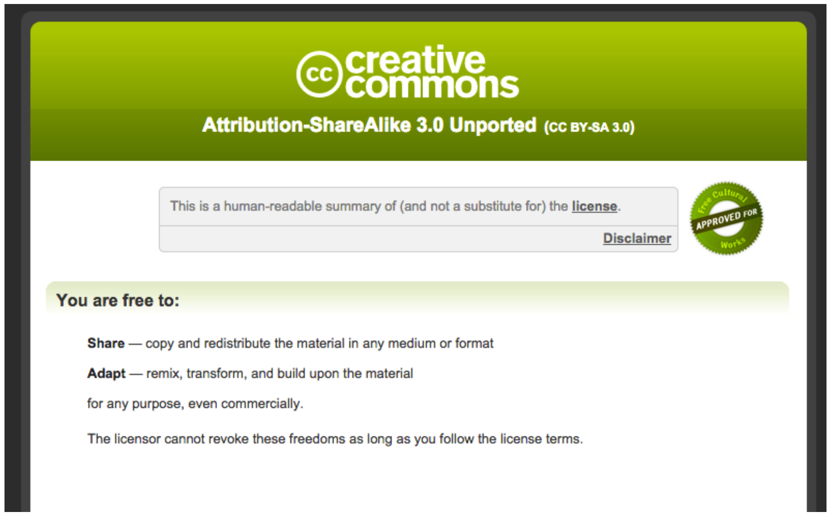
	Minimum milestones	Hopeful milestones
First semester	Course-required documentation	Prototyped LED control
	Procurement of viable parts for building final system	Prototyped software (parser)
Second semester	Single-note DSP	Polyphonic DSP
	On-board tuner	Tablature recording
	On-board metronome	Lessons playback
	Tablature playback	Additional fret LED coverage
	On-board LED control	Motorized tuning (unlikely)

Table 25 - eGuitar Project Milestones

Appendices

Appendix A – Copyright Permissions

- **Wolfram Alpha**: http://www.wolframalpha.com/termsofuse/ "The specific images, such as plots, typeset formulas, and tables, as well as the general page layouts, are all copyrighted by Wolfram|Alpha at the time Wolfram|Alpha generates them. A great deal of scholarship and innovation is included in the results generated and displayed by Wolfram|Alpha, including the presentations, collections, and juxtapositions of data, and the choices involved in formulating and composing mathematical results; these are also protected by copyright. You may use any results, including copyrighted results, from Wolfram|Alpha for personal use and in academic or non-commercial publications, provided you comply with these terms."
- **Texas Instruments**: http://www.ti.com/corp/docs/legal/termsofuse.shtml#info "Subject to any Service Terms that may apply, TI grants you permission to download, reproduce, display, and distribute TI Services on this site solely for non-commercial or personal use, provided that you do not modify such TI Services, and provided further that you retain all copyright and proprietary notices as they appear in such TI Services."
- **NXP Semiconductors**: http://www.nxp.com/terms-of-use.html "Copyright and all other proprietary rights in the Content (including but not limited to software, audio, video, text and photographs) rests with NXP Semiconductors or its licensors. All rights in the Content not expressly granted herein are reserved. Except as otherwise provided, the Content published on this Web Site may be reproduced or distributed in unmodified form for personal non-commercial use only. Any other use of the Content, including without limitation distribution, reproduction, modification, display or transmission without the prior written consent of NXP is strictly prohibited. All copyright and other proprietary notices shall be retained on all reproductions."
- **Sparkfun Creative Commons:** http://creativecommons.org/licenses/by-sa/3.0/



Appendix B – Works Cited

[1] Patricio Cohen. “Concepts and Terminology in Printed Circuit Boards (PCB).” Internet: http://www.pcb.electrosoft-engineering.com/04-articles-custom-system-design-and-pcb/01-printed-circuit-board-concepts/printed-circuit-board-pcb-concepts.html, 2010 [Accessed Nov. 11, 2014].

[2] Angela Wright. “Psychological Properties of Colours.” Internet: http://www.colour-affects.co.uk/psychological-properties-of-colours, 2014 [Accessed Nov. 8, 2014].

[3] W. E. Johns. “Notes on LEDs.” Internet: http://www.gizmology.net/LEDs.htm, 2003 [Nov. 9, 2014].

[4] TheLedLight. “LED Application Notes – LED Basics.” Internet: http://www.theledlight.com/technical1.html, Jan. 1, 2000 [Accessed Nov. 9, 2014].

[5] Jeffrey Bausch. “LED 101: Identifying different types of LEDs.” Internet: http://www.electronicproducts.com/Optoelectronics/LEDs/LED_101_Identifying_different_types_of_LEDs.aspx, Oct. 24, 2011 [Accessed Nov. 7, 2014].

[6] Kingbright, “SMD Chip LED Lamp,” APHBM2012CGSYKC Datasheet, Jun. 10, 2014.

[7] Yetda Industry Ltd, “SMD RGB LED,” 5060BRG4 Datasheet, [Accessed Nov. 9, 2014].

[8] PCBFABEXPRESS. “PCB Materials Glossary.” Internet: https://ecommerce.pcbfabexpress.com/glossary_materials.jsp, 2014 [Accessed Nov. 6, 2014].

[9] Alex R Englsih, “3D Printer Filament Buyer’s Guide.” Internet: http://www.protoparadigm.com/news-updates/3d-printer-filament-buyers-guide/, Jun. 14, 2012 [Accessed Nov. 8, 2014].

[10] Atmel, “8-Bit AVR Microcontroller with 32K Bytes In-System Programmable Flash,” ATmega328 Datasheet, Oct. 9 2013 [Accessed Nov. 9, 2014].

[11] All About Circuits, “Shift Registers Introduction.” Internet: http://www.allaboutcircuits.com/vol_4/chpt_12/1.html, 2014 [Accessed Nov. 7, 2014].
[12] ElectronicsTutorials, “The Shift Register.” Internet: http://www.electronics-tutorials.ws/sequential/seq_5.html, Nov. 22, 2014 [Nov. 7, 2014].
[13] Fairchild Semiconductor, “8-Bit Shift Register with Output Latches,” MM74HC595 Datasheet, Jun. 2009 [Accessed Nov. 9, 2014].
[14] USB.org, “USB Power Delivery.” Internet: www.usb.org/developers/powerdelivery, 2014 [Accessed Nov. 9 2014].
[15] Battery University, “What’s the Best Battery?” Internet: http://batteryuniversity.com/learn/article/whats_the_best_battery, 2014 [Accessed Nov. 9, 2014].
[16] BatterySavers, “Battery Life Questions: Which Batteries Last Longer?” Internet: http://www.batterysavers.com/Compare-Batteries.html, 2014 [Accessed Nov. 9, 2014].
[17] Microchip, “Miniature Single-Cell, Fully Integrated Li-Ion, Li-Polymer Charge Management Controllers,” MCP73831/2 Datasheet, Nov. 29, 2012 [Accessed Nov. 9, 2014].
[18] Wolfram Alpha LLC. 2014. Query: “plot y=sin(2*pi*x) and y = sin(4*pi*x) from x = 0 to 1”. Internet: http://www.wolframalpha.com/input/?i=plot+y%3Dsin%282*pi*x%29+and+y+%3D+sin%284*pi*x+%29+from+x+%3D+0+to+1 [Accessed Nov 25, 2014].
[19] Wolfram Alpha LLC. 2014. Query: “plot y = n*log base 10(n) and y=n^2 from n = 0 to 25”. Internet: http://www.wolframalpha.com/input/?i=plot+y+%3D+n*log+base+10%28n%29+and+y%3Dn%5E2+from+n+%3D+0+to+25 [Accessed Nov 25, 2014].