Automated Guitar Robot

A Technical Report submitted to the Department of Electrical and Computer Engineering

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Xuanjia Bi

Fall, 2022

Technical Project Team Members

Haotian Ren

Tianyue Guo

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Harry Powell, Department of Electrical and Computer Engineering

RGB – Automated Guitar Robot

Haotian Ren, Tianyue Guo, Xuanjia Bi

12/13/2022

Capstone Design ECE 4440 / ECE4991

Signatures

Statement of work:

Haotian Ren

I am primarily responsible for designing PCB board and doing embedded coding for MSP430 board, so that the boards can control corresponding servo motors to move in the correct directions, in order to perform left-hand fretting and right-hand plucking functions. My secondary responsibility includes building wooden frames and setting up Bluetooth communications.

Tianyue Guo

I am primarily responsible for building the Software GUI interface and setting up the Bluetooth communication between the app and the MSP board, so that the chord information from users can be transmitted successfully to the MSP430 board for further processing. My secondary responsibility consists of writing embedded codes in MSP430 to send out PWM signals to corresponding pins, and of building wooden frames that hold everything together.

Xuanjia Bi

I am mainly responsible for building CAD design for the wooden frames so that the wooden structure can be built in appropriate size, where both the guitar and the servo motors can well fit in. I am also responsible for building the wooden frames using wood-cutting equipment. In addition, I am responsible for assisting embedded coding design in MSP430 and setting up Bluetooth communication.

Table of Contents

Table of Figures

Table of Tables

Abstract

This project aims to build a physical robotic, automated guitar robot based on given guitar tabs with corresponding tempo. Users will send out chord information by typing in a text box or uploading a text file from a laptop to a microcontroller; consequently, the microcontroller processes chord information and controls servo motors to pluck, strum, and press string. This project is designed to perform two main movements: left-hand fretting and right-hand plucking. The automated guitar player implemented software Application Programming Interface (API) to help users input guitar tabs and translate tabs into diatonic chords. This project also takes consideration of commercial uses; thus, the design objectives include flexibility, accessibility, low cost, and performance.

Background

In the past decade, with the fast development of artificial intelligence and robots, applications featuring these technologies can be found almost everywhere in people's lives [1]. For example, autonomous driving assistance systems are popularizing and tend to replace human drivers; automated floor cleaning robots are helping people offload some household responsibilities. In the field of music, such progress is also happening. From the tools that assist musicians with creating songs to the machine learning models that can compose a piece of music, artificial intelligence has already had its foot inside the door of music. However, robots could also have their place in music, assisting people with educational and recreational purposes or even replacing a human player in a band.

An open-source hardware automated guitar robot was made by students from the University of Lincoln [2]. It comprises of left-hand fretting module and right-hand string picking module. Automation is performed through electric linear solenoids. It has high flexibility for the range of instruments, as various types of instruments can be used. The same for level of automation, as the guitar robot can automate either the left hand, right hand, or both, meets the users' needs. There are some other types of automated guitar robots with different implementations.[3][4]

For our project, a similar design to the one illustrated above is used in terms of basic functions such as fretting and string picking. However, there are several important distinctions that suit more for our goals. The automation would not be performed via solenoids, as it would not fit our financial constraints: each solenoid costs at least $30 and 30 solenoids at minimum are needed, while our budget is limited to $500 [5]. Therefore, servo motors will be used instead of solenoids to conduct left-hand fretting. Except the goal of making the guitar robot play all diatonic chords in all 12 keys, another goal is to let the robot read a list of chords from a text file and play them in sequence.

The guitar robot consists of mainly a software part as a controller and an embedded part as an actuator. The software related classes all group members have taken are computer networks, advanced software development, and software development methods. The knowledge gained from these courses will help us in designing the software application and communications between the computer and the microcontroller. For the hardware side, the techniques that will be used are mostly from the course introduction to embedded systems. Besides, experience designing PCBs from the FUN series might also be useful in designing header board connected to the microcontroller.

Physical Constraints

Design Constraints

During our implementations, several constraints were placed since September, and some of them made us pivot our design to find alternative solutions. Design constraints include motor quality, timer system and debugging limitation on embedded device, GUI license, socket and cables availability, 3D-printed firmware toughness, adjustable current-limited power switches, woodshop craft time-consuming.

Servo Motor Quality

Due to budge constrains, the servo motor used is FS90, positional servo motor, made by Pololu inc [7]. Such entry servo motor was lack of quality control; consequently, many cautions aroused such as overcurrent/ stalking issue at certain rotation degree, mismatch between pulse-width modulation and desired rotation degree, and internal wire connection faulty, non-rectangular actuator bases. FS90 should be taken into consideration of the following constraints.

- Maximum rotation degree 180°
- Peak torque limitation up to a 1% duty cycle
- Maximum current without stall 200 mA
- Maximum motor RPM 108
- Stalling torque 18 oz·in

Embedded Device Limitation

MSP430FR2476 [8] microcontroller was part of our design due to its ultra-low-power low-cost with 43 GPIO pins. MSP430 family of low-power microcontrollers consists of devices with different sets of peripherals targeted for various applications, and such devices can help us to send out pulse-width modulation and decoder inputs. However, there are some limitations rase our concerns – complicated timer system and lack of tutorial resources. Following timing constraints are vital.

- The digitally controlled oscillator should operate equal or less than 16 MHz
- MSP430 includes 64KB FRAM and 8KB SRAM
- UART, SPI, and I2C were supported

Software Availability

Our Graphical user interface was developed in PySimpleGUI, and such GUI library can help us build interface without over-spending time on tremendous tutorials. One consideration for our GUI would be license. With PySimpleGUI, there isn't a way to get any formal license, and potentially will have a hard time implementing massive productions. To achieve the proper license, all GNU Lesser General Public License [9] v3.0 should be followed.

Part Availability

Molex micro locks [10] were selected to serve as connectors. Given that thirty-seven servo motors are planned in use, finding proper cables to connect between the motors and PCB becomes our constraints. In domestic market, there are not any in-stock cables with correct size can convert socket mount to female pins.

- 3 pin connectors
- Surface Mount
- Pitch- Mating 0.049" (1.25mm)

Frame Materials

Ultimate Maker Pro S5 [11] with wood-filament was planned to build the guitar frame initially. However, it is questionable whether the firmness of the wood filament can hold the motors as well as the cost to build a large size firm to support the guitar and the motors. Our alternative solution to walkaround highly cost 3D printing materials is woodshop craft, which might cost more time to build but guarantee the firmness of our wood frame.

Manufacturing Limitations

Building PCB boards are the major constraints in terms of Manufacturing limitation. According to Advanced Circuits, a two-layer PCB, Copper top and Copper bottom, pricing is $33 per one with physical constrains [12]:

- Minimum 5 mil line/space
- Minimum 10 mil hole size
- Maximum .125" board thickness
- Maximum 1 oz. Inner weight

Cost Constraints

The budget for this project is limited to $500, which covers all components, printed circuit board manufacturing, and shipping. Original string pressers were planned to use solenoid [13] due to its high stability and simplicity. However, each solenoid costs above $30 and 30 solenoid at minimum are needed, which goes beyond the designed budget. Although replacing soleoid with entry servo motor, FS90 [14], the largest expected cost is still them. More than 30 servo motors with $7 each are needed for left-hand and right-hand designs. The PCB board, MSP430 board, and Bluetooth module are more expensive by unit price compared to servo motor's unit price; however, since ideally, they are one-time purchases, they should not be of major concern. In this project the woodshop craft is done with no cost at a university library.

Tools Employed

To complete automated robot project, all the tools used were divided into four categories: Design, Mechanical, Firmware, and Software. All tools for each category are listed in detail below.

Design

Guitar frame requires many detailed motor holders and guitar supporters. To design such a complicated frame that holds more than 30 motors above fretboard with limited space, a computer-aided design tool, Fusion [15], is used. Ultimate Cura [16] is also used to build the motor holder prototype. In the other hand, designing the circuit and layout the printed circuit board, KiCad [17] was used. KiCad is a free software suite for electronic design automation. It facilitates the design and simulation of electronic hardware. All the automation drawing, design rule check, pin assignments were completed with KiCad. Matlab [18] provides a numeric computing platform to determine the maximum current for this project as well. Lastly, to guarantee all inputs and outputs response properly, Waveform [19] with Analog Discovery 2 [20] were used for mentoring signals

Mechanical

Building the guitar frame and soldering the PCB was conducted in Architecture School and Electrical and Computer Engineering School in the University of Virginia. Several woodworking equipment was used to complete the frame including Table Saw [21], Drill Press [22], Band Saw [23], Disk and Belt Sander [24], Router [25], and Shop Vacuum [26]. In addition, a 3D printer [27], Ultimate Maker Pro S5, is used to build our project. Electronics and soldering equipment such as Soldering Iron [28], Solder Braid [29], Wire Stripper [30] were also used.

Firmware

To build on the firmware design, Code Composer Studio [31], an integrated development environment to develop applications for Texas Instruments embedded processors, was used. Code Composer Studio provides not only a platform to develop the embedded logics but also allows users to debug issue step by step. Additionally, PySerial [32] and Bluetooth Serial Terminal [33] were employed to establish the wireless communication protocol between laptop and robot.

Software

The interface was written in Python 3.10 [34] with many libraries such as PySimpleGUI [35] for GUI, Serial [36] and Time [37] for Bluetooth communications, pathlib [38] for importing guitar chords, and pyinstaller [39] for packing all software into an executable file.

Societal Impact Constraints

Environmental Impact

This project relies on wood as original material to make the entire guitar frame as well as all the parts that support the servo motors. The reason why wood has been chosen is that it allows decent amount of fault tolerance when it comes to height and width measurement. The precise location of each servo motor was difficult to measure. If the wood structure holds a servo motor too high, it could be modified using sandpaper. However, choosing wood comes with concerns on environmental impact and sustainability due to limited resources of wood that the earth can reproduce. Over-farming could raise carbon dioxide emissions, deforestation, material uses and waste. On the other hand, recycling wood can be challenging as the wood frame attaches with different motors tightly, and wood-waste dealers or disposal for recycling or repurposing have a hard time detaching all the motors for small pieces of wood.

If this project goes to production, the alternative way is solid plastic frame or recycled wood fill filament from 3D printing.

Sustainability

Our design is generally sustainable as the wood structure holds the guitar and servo motors at appropriate locations steadily after a large amount of test trials. One concern is that in the very long term, there might still be some abrasion of the wood structure. Particularly, cracks might happen at the part where servo motors are nailed in. Hence, the team has designed several backup parts in order to replace the original when problems happen. In addition, the servo motors might malfunction after a long time of working; hence, the team has made sure that each servo motor is removable and replaceable – they can be unscrewed and nailed in easily.

Health and Safety

This design generally does not contain noticeable health and safety issues, except for flammable parts like the guitar frame and the guitar itself, as they are mainly made of wood. Our design has abided by Code of Federal Regulations 1500.3(c)(6)(vi) [40].

Ethical, Social, and Economic Concerns

This project raises two main ethical and economic concerns that must be addressed if automated robots go to production. First, if some people try to over-promote the automated robots, it could potentially be detrimental for the guitar performers' job security (M. Anderson) [41]. As designers, it is our responsibility to make sure that our automated guitar robot only plays a role as a recreational tool rather than live performers. The second concern comes from device security during data exchange (Abowd) [10]. No personal information is transmitted; hence privacy issues do not raise major concerns. However, security vulnerabilities in Bluetooth technology remain, as there might be various forms of attacks on Bluetooth devices [42].

External Considerations

External Standards

In our project, the GUI application communicates with MSP430 board with Bluetooth. Bluetooth Low Energy (BLE) is used as the technology for communication because BLE is intended to save energy with a low cost. The Bluetooth technology abides by IEEE 802.15.1 standard [43].

IPC standards were required for our PCB design. To ensure design reliability, this project is implemented with IPC-2221, a generic standard that covers almost every aspect of PCB design such as PDN bus

layouts, conductor clearance, and impedance control. The standard was check with the online FreeDFM checker.

UART, Universal Asynchronous Receiver-Transmitter, is used as a standard serial communication between Bluetooth module and the MSP430 board. UART helped receive chords information from GUI application and send back ACK messages, indicating that MSP boards have received information. The baud rate is set to be 9600 bits/second, and the COM ports on laptop is set to be COM8.

This project also abides by Code of Federal Regulations 1500.3(c)(6)(vi), which illustrates that the solid substance should ignite and burn at a rate greater than one-tenth of an inch per second along its major axis.

Intellectual Property Issues

This project is not quite possible to be patented, as there are several similar products already. One guitar playing robot already comprises “a base assembly, a pick assembly, a strumming mechanism and a fretting mechanism”, and the musical instrument is easily interchangeable with another stringed instrument. The strumming and fretting mechanism include similar functions, such as changing chords and controlling speed, but with more flexibility and more ranges of chord selections. Another guitar self-playing robot not only plays instrumental music by itself, but also contains biodegradable musical plectrum. Finally, a guitar playing robot patented in 1998 contains several separate devices that pluck and fret strings, as well as maintaining tensions in the strings. As these robots incredibly resemble covered in our projects and even contain more flexible features, our project is likely not patentable.

Detailed Technical Description of Project

General Overview

As artificial intelligence is gaining popularity in various fields, the goal is to create something for guitars as well. For guitar beginners, it is generally difficult to play guitar chords correctly. The coordination between fretting and plucking requires a large amount of memorization and practice. Hence, the purpose of this project is to create an automatic guitar robot for guitar beginners who struggle to learn guitar chords and for listeners who want to enjoy music. When users import a music sheet into a GUI application, the app will preprocess the input and transmit the guitar chords via Bluetooth to the microcontroller, which will control specific servo motors to finish the action of fretting and plucking. This robot is a real time design; therefore, the guitar robot can receive data and play chords semitonally. Furthermore, this robot can play a sequence of chords or notes with some customized strum or plucking patterns. This automatic guitar robot has great potential to allow users to implement other hand actions such as mute, slide, touch fourth, and lefthand pluck.

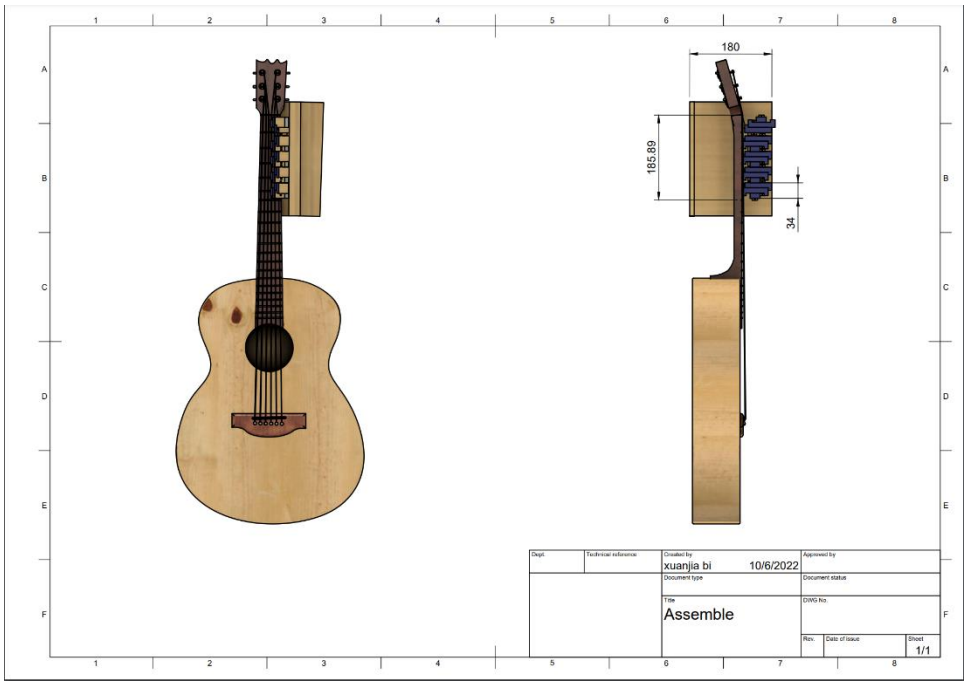


Figure 1. CAD Overview

This project contains both software and hardware sections. The software section allows users to input desired chords, notes, or a sequence of diatonic chords through GUI in laptop. Consequently, the interface processes the inputs into a sequence of binary data and sends them out through Bluetooth. Once Bluetooth module, HC-05, receives the data, it communicates the microcontroller, MSP430, through UART. If MSP430 accepts the input data, it sends back an “ACK” to interface, a signal used in digital communications to notify the GUI and converts the binary data into 3-bit decode information and pulse-width modulation for PCB. Lastly, based on the binary decode information and pulse-width modulation signals, PCB selects corresponding servo motors and sends out the pulse-width modulation signals to control the servo motors.

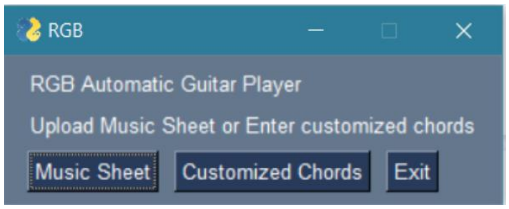


Figure 2 Graphic User Interface

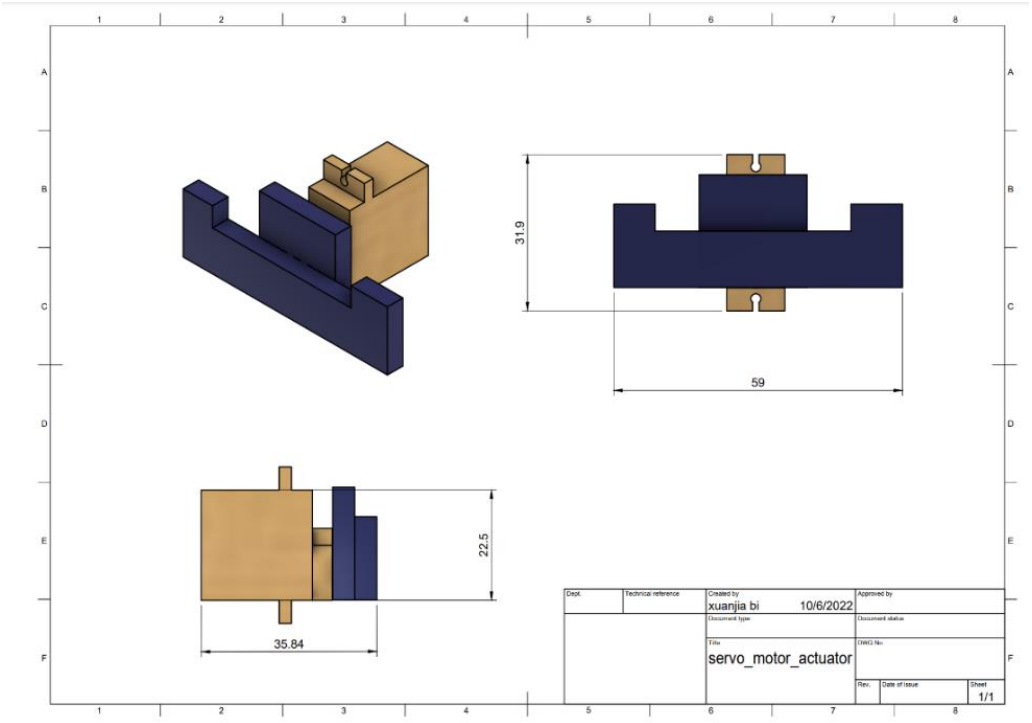


Figure 3 Servo Motor Overview

Components

This automatic guitar robot is divided into four sections hardware, firmware, software, and mechanical.

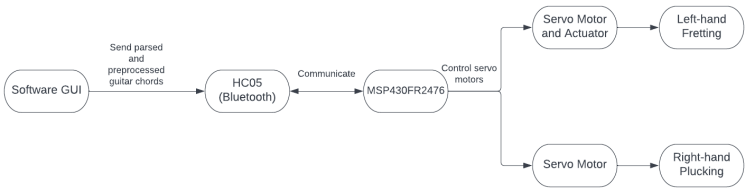


Figure 4 High-level of Block Diagram

Hardware

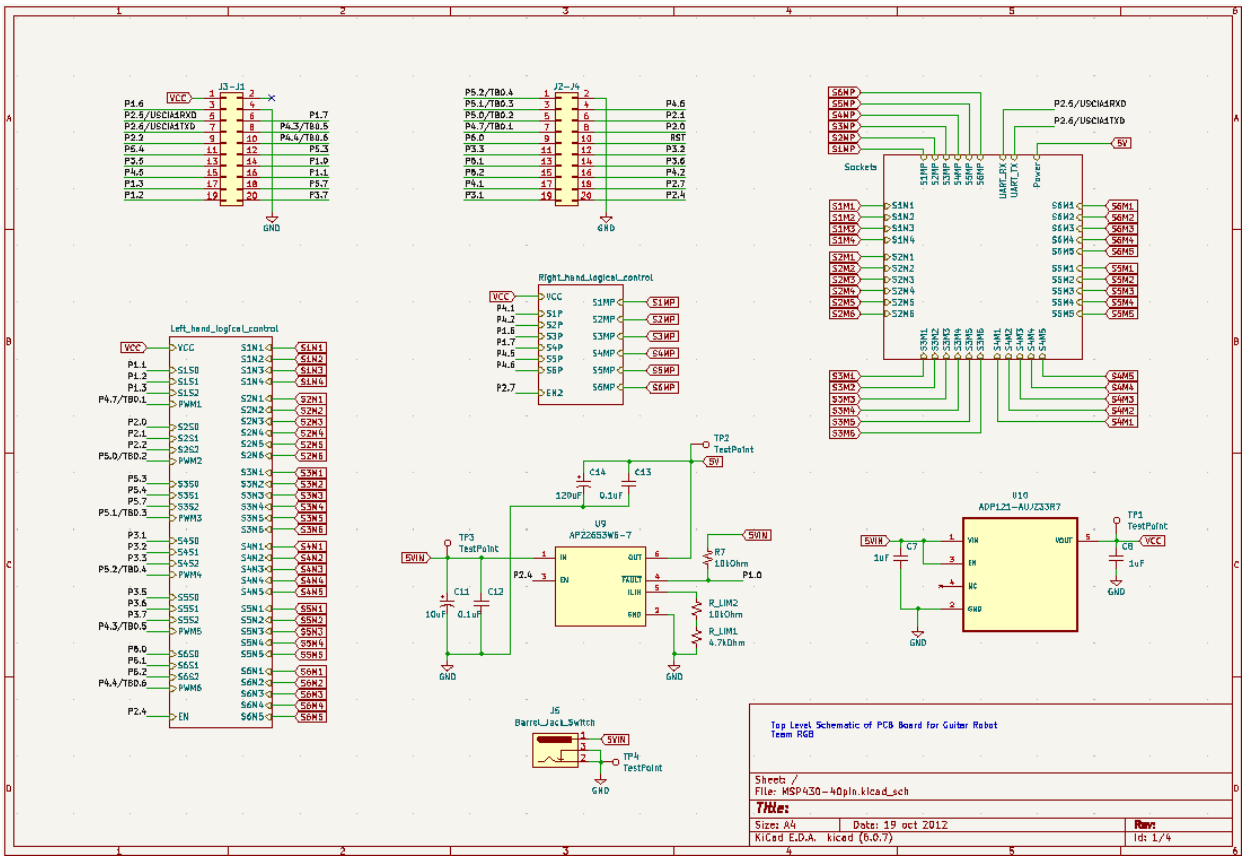


Figure 5 Top Level Schematic

In the top-level schematic, there are 7 different groups of designs, including the sockets for MSP430FR2476, hardware components controlling motors for the left-hand part, hardware components controlling the motors for the right-hand part, a voltage regulator, a power management module, a power jack, and sockets for motors. The detailed designs of these sub-parts are listed below with hierarchical schematics and explanations.

Voltage Regulator

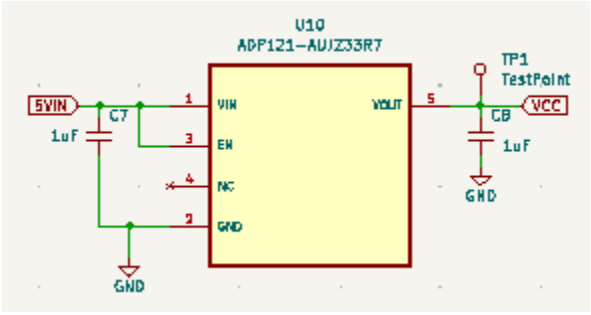


Figure 6 Voltage Regulator

A voltage regulator is used to convert the 5V power input, denoted as *5VIN*, from the power jack to 3.3V power output, denoted as *VCC* in the schematic. The voltage regulating chip used here is ADP131-AUJZ33R7. It has 5 pins with their functions shown in the table below.

Pin No.	Pin Name	Functionality	Net
1	VIN	5V Power Input	5VIN
2	GND	Ground Power Input	GND
3	EN	Chip Enable	5VIN
4	NC	-	-
5	VOUT	3.3V Output	VCC

Table 1. Voltage Regulator Pins Assignment

According to the data sheet of this voltage regulating chip, there are two bypass capacitors of size $1\mu F$ on both the power input and power output sides. To make the chip auto-start when a power is connected to the PCB board, the EN pin is connected to the 5V power line. The NC pin is unused in this chip, so it's left floating.

Power Jack

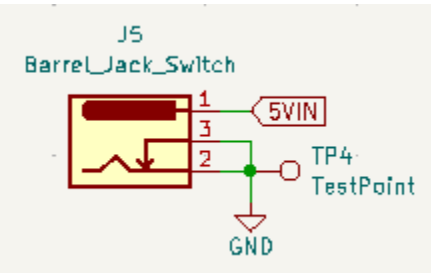


Figure 7 Power Jack

The power jack used in our design is a standard type A barrel jack with 2.1mm inner diameter and 5.5mm outer diameter. Since there is not a power source using battery, both pin 2 and pin 3 are connected to the ground power line, with a test point connected to the ground power line for debugging purpose. Pin 1 is connected to the 5V power input line on the board. This power line is neither connected directly to any chips nor connected to any motors. Instead, it goes into the voltage regulator chip or the power management chip. This design avoid possible damage to the board. If anything in the power supplying part, including the voltage regulator, the power management module, or this barrel jack is shorted, there won't be any surging current affecting the chips, servo motors or the CPU.

Power Management Module

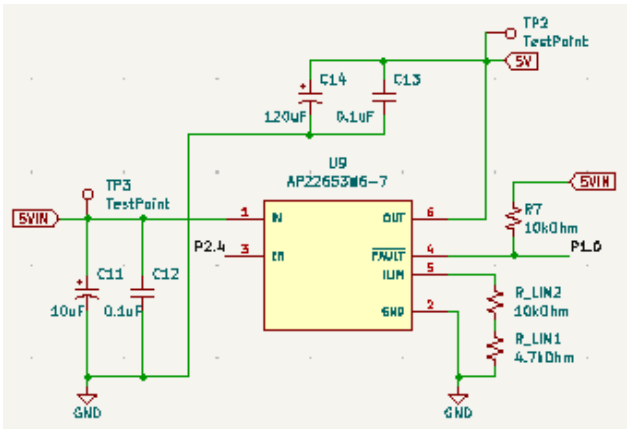


Figure 8 Power Management Module

In the design of actuators, as servo motors are used, a practical problem that must be considered is overheating. When a motor gets stuck, it will hit the stall current, accumulating heat, and finally break down. To effectively limit the current that can be drawn by the servo motors in the case where some motors got stuck, a power management chip is used. The chip, AP22653W6-7, can limit the current supplied to servo motors by setting the resistance between pin 5, *ILIM*, and the ground. Detailed pin information is shown in the table below.

Pin No.	Pin Name	Functionality	Net
1	IN	5V Power Input	5VIN
2	GND	Ground Power Input	GND
3	EN	Chip Enable	P2.4
4	FAULT	Over-current Flag	P1.0
5	ILIM	Limiting Current Configuration	-
6	OUT	5V Power Output with Max Current Limit	5V

Table 2. Power Management Chip Pins Assignment

The *FAULT* pin is active low and connected to P1.0 of the MSP launchpad. P1.0 is also connected to a green LED on the launchpad, therefore, if there is an over-current flag, it's also visible to the user as the green LED will turn off. The enable pin is connect to P2.4, which also controls the enabling of left-hand motor signals. Once P2.4 is set to high, neither of the chips controlling the left-hand motor signal nor this power management chip controlling the power supply to the motors will be activated.

According to the current analysis of motors, each motor in the left-hand part will draw up to 230mA of current and each motor in the right-hand part will draw up to 150mA. In the worst case, there will be 6 motors in the left-hand part and 6 motors in the right-hand part functioning, drawing about 2280mA of current. The equations for current-limit threshold provided in the data sheet is:

$$I_{LIMIT} = 30321/R(k\Omega)^{1.055}$$

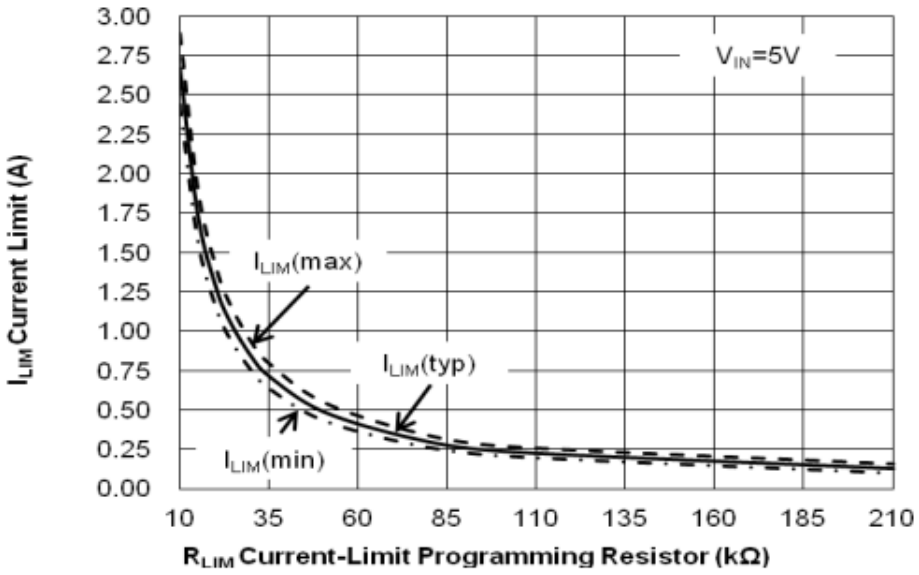


Figure 9 Best Fit Current-Limit Threshold

By choosing $R_{LIM1} = 10k\Omega$, $R_{LIM2} = 1.5k\Omega$

$$I_{LIM} = \frac{30321}{11.5^{1.055}} = 2305mA$$

Even in the worst case, the limit supplied to the motors will be

$$I_{LIM_{max}} = \frac{31033}{11.5^{1.031}} = 2502mA$$

As the voltage adapter have a max current of 3000mA under 5V and the chips, Bluetooth module, and MSP430FR2476 will draw no more than 300mA of current, it's still within the safe zone.

According to the data sheet, two bypass capacitors of size $10\mu F$ and $0.1\mu F$ are connected to the power input side; two bypass capacitors of size $120\mu F$ and $0.1\mu F$ are connected to the power output side; one $10k\Omega$ pull up resistor is connected to the over-current flag signal output.

MSP430FR2476 Socket

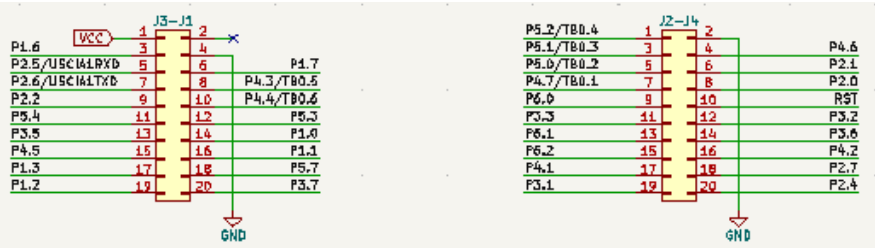
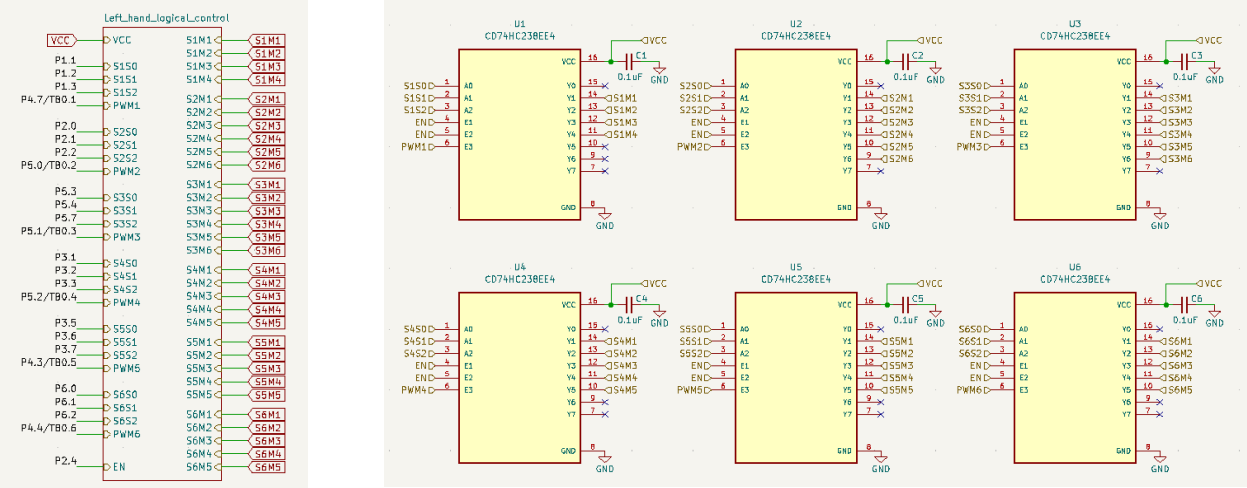


Figure 10 MSP430FR2476 Socket

This part of the schematic is associated with a PCB layout of two 2x10 pin sockets designed specifically for the MSP430 launchpad, which can be found inside Kicad. Though MSP430FR2476 CPU features 43 GPIOs, there are only 35 GPIO pins available on the launchpad. For our design, all of them are assigned to some specific tasks. For the power of the launchpad, pin 1 on J3-J1, the 3.3V power pin, is connected to the 3.3V power line. For pin 2 on J3-J1, the 5V power pin, since there is no need of this pin, it is left floating intentionally. Pin 4 on J3-J1 and pin 2 on J2-J4, the ground pins of the launchpad, are both connected to the ground power line. Usage of other pins are covered in the introduction of other parts below.

Left-hand Control Signal Module



(a) (b)

Figure 11 (a) Top Level View of Left-hand Control Signal Module (b) Detailed view of Left-hand Control Signal Module

In the left -hand control signal module, there is a demultiplexer for signal controlling of motors on each string, responsible for fretting the strings. As shown in the left figure above, the input to this module are signals from the MSP430FR2476 launchpad GPIOs and the power supplies. The detailed assignments of GPIO pins to the demultiplexers are shown in the table below.

Guitar String (Note)	GPIO Name	Signal Description
String1 (E)	P1.1	String 1 Demultiplexer Select Bit 0
	P1.2	String 1 Demultiplexer Select Bit 1
	P1.3	String 1 Demultiplexer Select Bit 2
	P4.7 / TB0.1	PWM Signal from Timer B CCR 1
String 2 (A)	P2.0	String 1 Demultiplexer Select Bit 0
	P2.1	String 1 Demultiplexer Select Bit 1
	P2.2	String 1 Demultiplexer Select Bit 2

	P5.0 / TB0.2	PWM Signal from Timer B CCR 2
String 3 (D)	P5.3	String 1 Demultiplexer Select Bit 0
	P5.4	String 1 Demultiplexer Select Bit 1
	P5.7	String 1 Demultiplexer Select Bit 2
	P5.1 / TB0.3	PWM Signal from Timer B CCR3
String 4 (G)	P3.1	String 1 Demultiplexer Select Bit 0
	P3.2	String 1 Demultiplexer Select Bit 1
	P3.3	String 1 Demultiplexer Select Bit 2
	P5.2 / TB0.4	PWM Signal from Timer B CCR4
String 5 (B)	P3.5	String 1 Demultiplexer Select Bit 0
	P3.6	String 1 Demultiplexer Select Bit 1
	P3.7	String 1 Demultiplexer Select Bit 2
	P4.3 / TB0.5	PWM Signal from Timer B CCR5
String 6 (E)	P6.0	String 1 Demultiplexer Select Bit 0
	P6.1	String 1 Demultiplexer Select Bit 1
	P6.2	String 1 Demultiplexer Select Bit 2
	P4.4 / TB0.6	PWM Signal from Timer B CCR6

Table 3. GPIO Assignment for Left-hand Control Signal Module

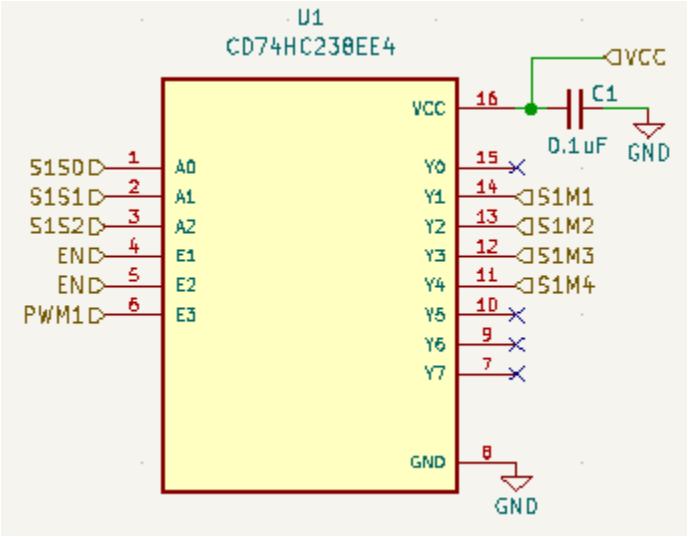
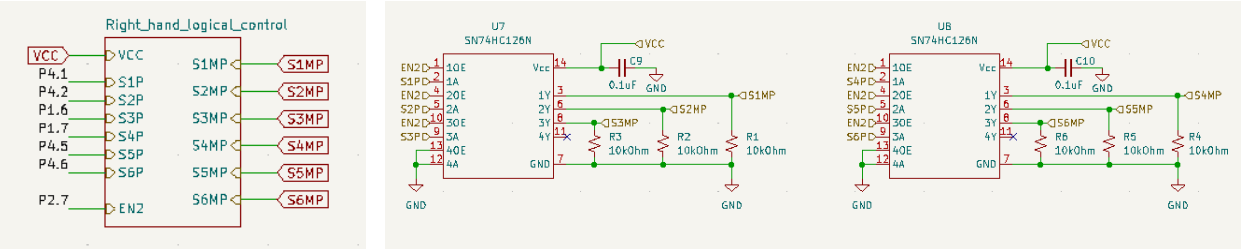


Figure 12 Demultiplexer Design

For each demultiplexer, as shown in the figure above, pins A0 through A2 are select bits, E1 through E3 are all enables, but E3 is used to transmit PWM signals to the signal input of a specific motor selected, a port between Y0 through Y7. Y0 is left floating for no fretting but plucking; Y6 and Y7 are left floating for no fretting and no plucking, in short, doing nothing with that specific string; Y1 through Y5 corresponds to motors above fret 1 though 5. In this case, string 1 fret 5 is not used in any diatonic chord, so it's left floating intentionally. The power input pin of this integrated circuit is connected to 3V power line and a bypass capacitor of size $0.1\mu F$ based on the information from the data sheet.

Right-hand Control Signal Module



(a) **(b)**

Figure 13 (a) Top Level Schematic of Right-hand Control Signal Module (b) Detailed View of Right-hand Control Signal Module

In the right-hand control signal module, there are two buffers designed to separate the signal output of the MSP430FR2476 launchpad and the signal input ports of the servo motors for safety concerns like current leakage. The figure on the left shows the input signals to this module from the CPU. P2.7 is used for enabling the buffers. The other 6 GPIO signals transmit PWM signals and correspond to 6 strings on a guitar.

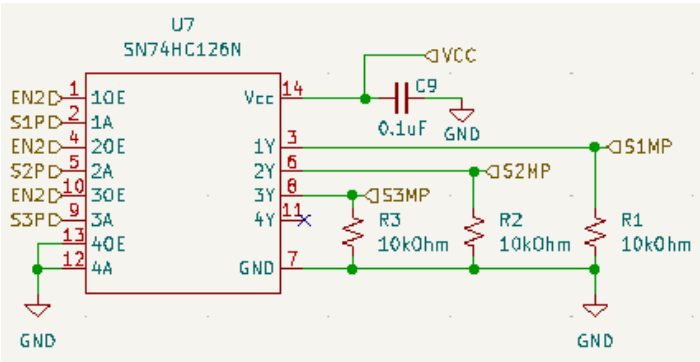


Figure 14 Buffer IC

As shown in the figure above, there are 4 sets of buffers in one buffer IC. As there are 6 strings, only 3 sets of buffers on each chip are used. The other one is left floating, but the input pins to this idle set is connected to the ground to avoid unintentional results on the output side. For each buffer, there is a signal input, an enable, and a signal output. According to the data sheet of this buffer IC, the power supply pin is connected to a capacitor of size $0.1\mu F$. Each output of this chip is connected to a $10k\Omega$ pull-down resistor to make sure that there is no random signal to the servo motors when either the chip or a buffer is disabled.

Sockets for Servo Motors

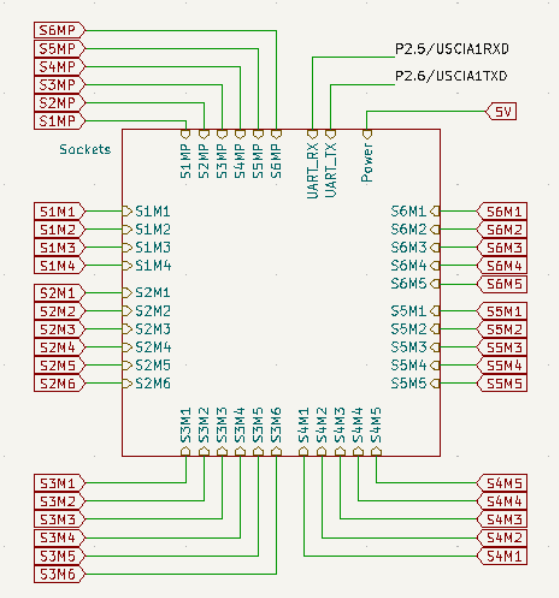


Figure 15 Top Level of Sockets for Servo Motors

In the figure above, it shows the top-level schematic of the sockets for servo motors. There are six groups of sockets corresponding to motors on six strings and a socket for the communication between the MSP430FR2476 launchpad and the Bluetooth module.

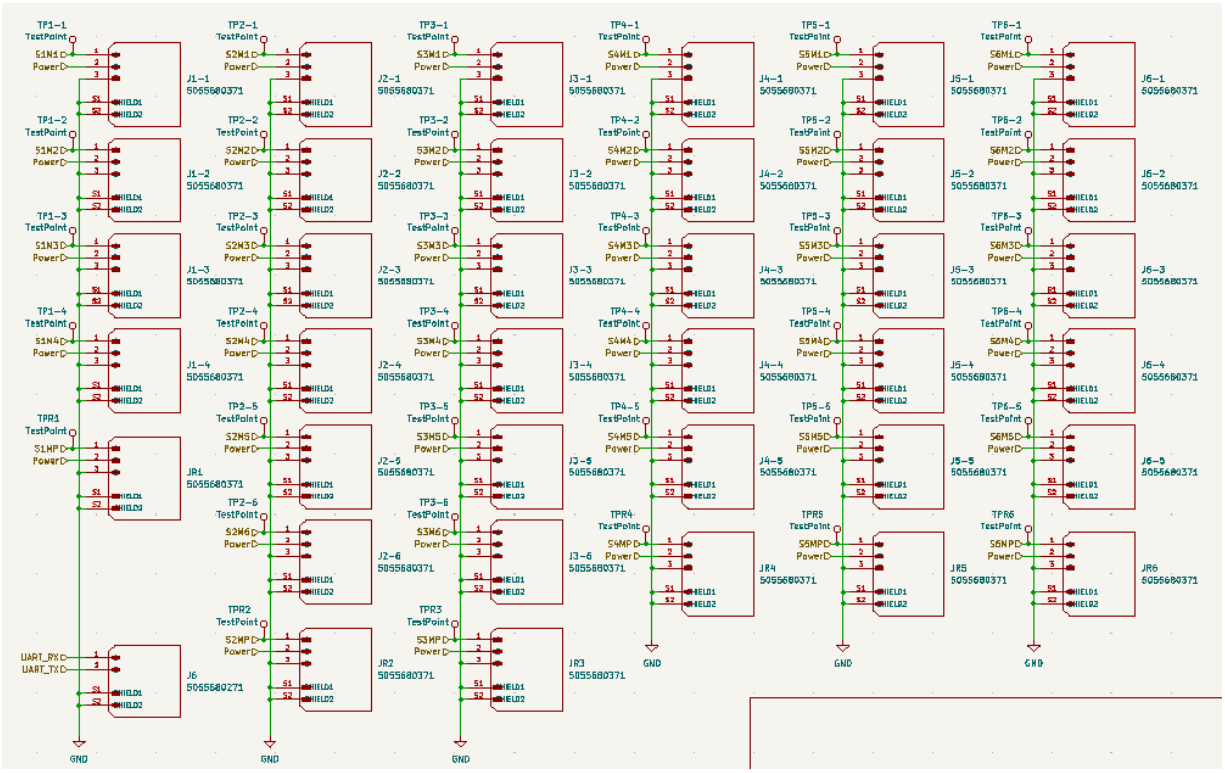
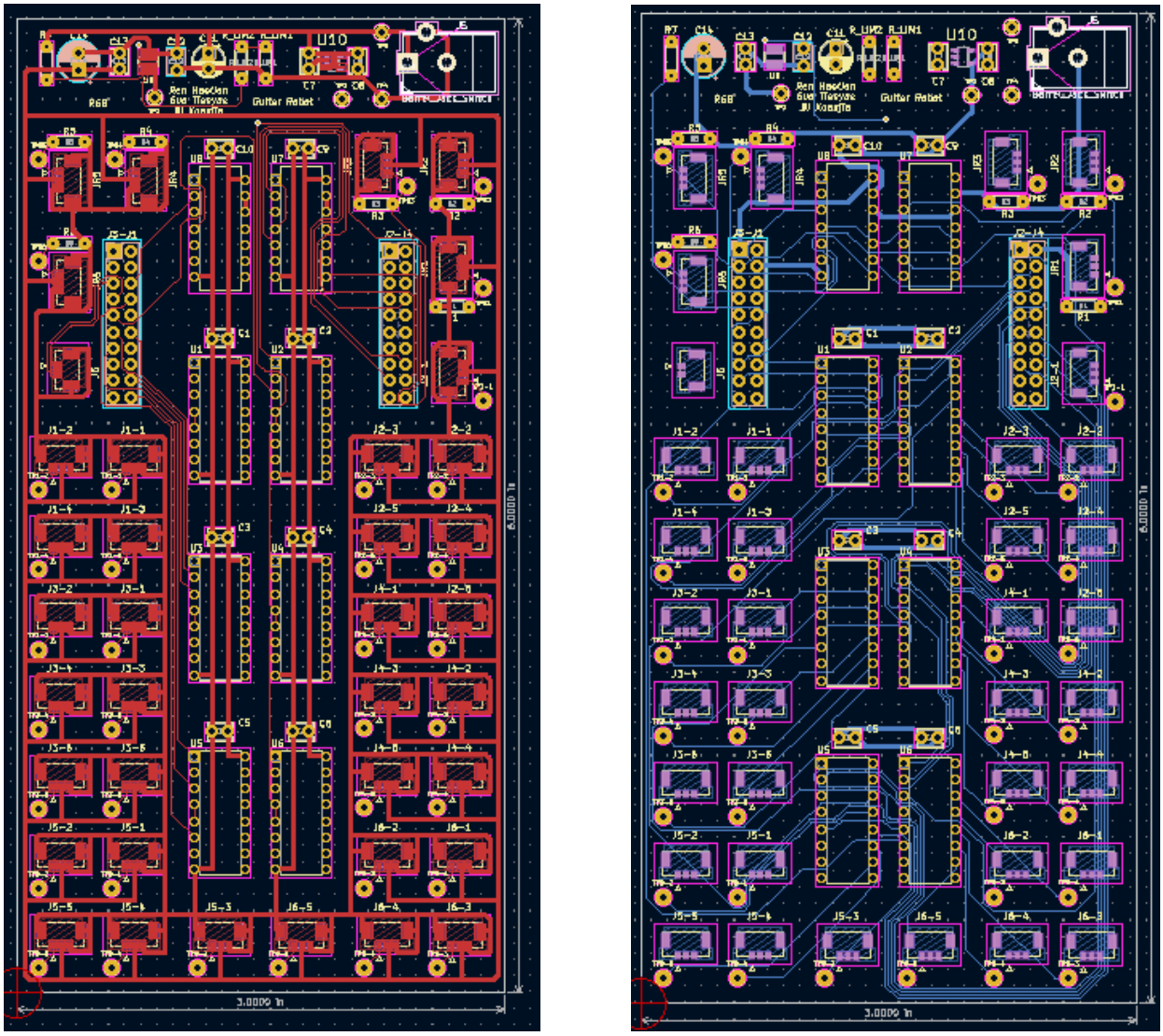


Figure 16 Detailed View of Schematic for Sockets

In the detailed view of the sockets and the other parts in the schematic, it shows how the signals and powers are connected. For servo motors, there are three pins: 5V power input, ground power input, and PWM signal input. For the Bluetooth module, there are only two pins, connecting to UART module TX and RX pins on the MSP430FR2476 launchpad. For every socket, there are two shield pins connected to the ground as intended by the designer of the socket, and there is a test pin for the signal line of each socket.

PCB Layout



(a) (b)

Figure 17 (a) PCB Layout Front Layer (b) PCB Layout Back Layer

In the left figure above, it shows the front copper layer of the PCB board, it contains the layout of most power lines and several signal lines connecting chips to GPIOs on the launchpad. The figure on the right shows the left power lines and most of the signal lines. The size of the whole board is 4.0 inches by 6.0 inches, the size of a modern smart phone. All the IC chips are placed in the middle of the board for ease of connection to the MSP launchpad sockets, with their bypass capacitor placed next to their power input pins. All the sockets are placed around the chips and the MSP launchpad socket, grouped according to each string. The top part of the board is responsible for all power related works. A detailed view of this part is shown below.

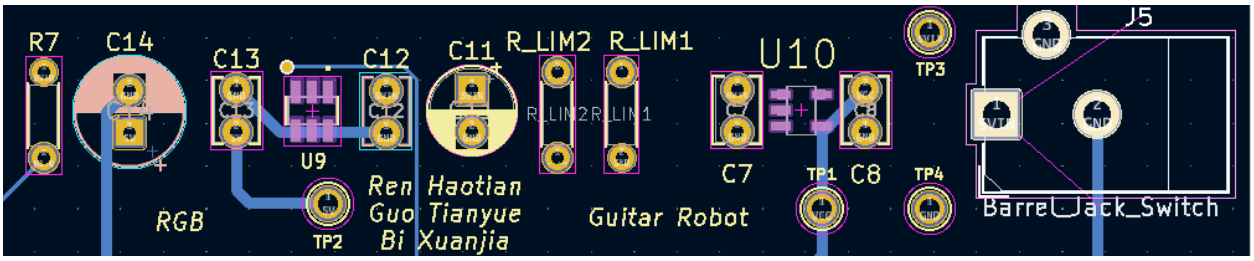


Figure 18 Detailed View of The Power Supply Part

On the right side, J5 is barrel jack for external power supply. Next to it is U10, the 5V to 3.3V voltage regulator with its bypass capacitors. Everything left on the left part of the figure is the power management module. U9 is the chip, with its bypass capacitors, including C11 to C14 right next to it.

Firmware

The firmware in this project are the MSP430FR2476 launchpad[50] and the FS90 servo motors. Detailed information about the microcontroller will be mentioned below. MSP430FR2476 is used for controlling the IC chips, including demultiplexers and buffers, communicating with the PC side through the HC-05 Bluetooth module, and generating PWM signals to control the servo motors.

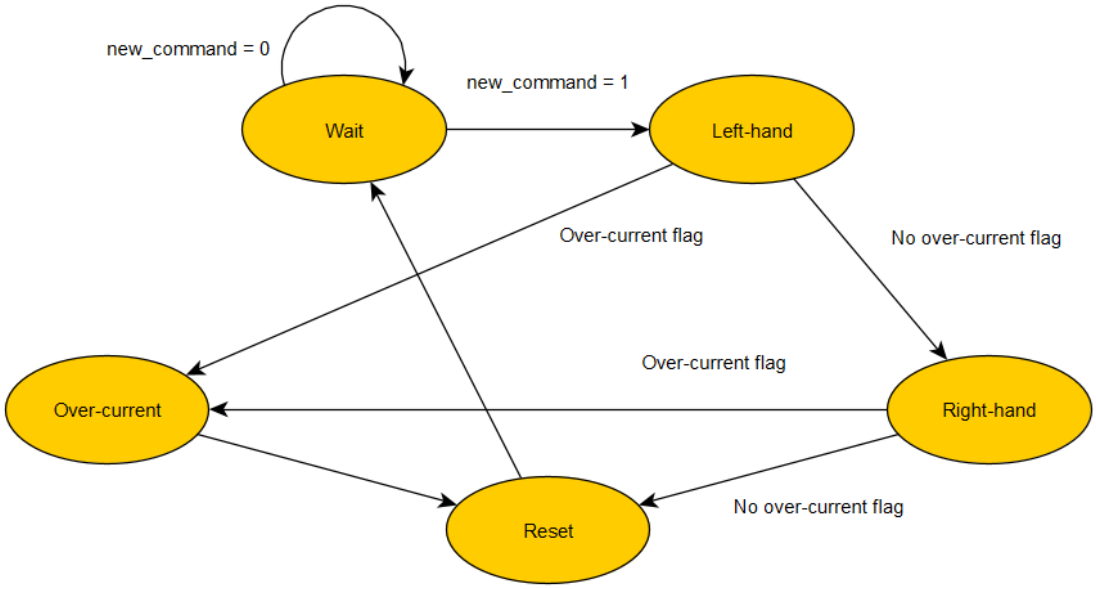


Figure 19 Finite State Machine

In the figure above, it show how the finite state machine looks like inside the process of playing the guitar. There are a total of 5 states. The initial state of the FSM is the Wait state, in which the microcontroller will establish Bluetooth connection to the PC side right after the first call to the output

function of the FSM. In the Wait state, the microcontroller will use the UART module to listen to commands sent from the PC side. Once there is a new command, the FSM will go to the Left-hand state, Right-hand state, and the Reset state sequentially. The Left-hand state will set the PWM signals to press some servo motors down on the strings according to the command received. The Right-hand state will send PWM signals to the motors on the right-hand side to pluck the strings. Finally, another set of PWM signals will be sent to the servo motors on the left-hand side to reset them. During the Left-hand state and the Right-hand state, if there is an over-current event, the FSM will go to the Over-current state to reset all the motors to avoid damages. After a round of operations, the FSM will send an ACK message in the Wait state to the PC side to let it know that a chord has been played and then go back to the Wait state and listen to new commands.

```
4  void configureTimerA3(void)
5  {
6      // Stop and clear timer
7      Timer_A_stop(TIMER_A3_BASE);
8      Timer_A_clear(TIMER_A3_BASE);
9
10     // Set timer register
11     TA3CCR0 = 19999;
12
13     // Start timer
14     TA3CTL |= (TASSEL_2 | ID_0 | MC_1);
15 }
```

Figure 20 Timer Setup

In the figure above, it shows the setup of a timer. First, the timer will be stopped and cleared, then the TAxCCRx or TBxCCRx registers and the TAxCCTL or TBxCCTL registers will be set to generate the PWM signals. Finally, the last line will select the clock used for the timer and start the timer.

According to the data sheet of the servo motors, PWM signals with a duty cycle of 1ms to 2ms and total period of 20ms should be used to turn the servo motors. As the SMCLK of the microcontroller is set to 1MHz, according to the equation below, values of the CCR registers are set.

$$Duty\ time\ of\ CCRx\ Output = \frac{CCRx - 1}{1MHz}$$

The communication between the PC side and the MSP430FR2476 launchpad through Bluetooth module is done through the UART API. It uses timer A1 as a timing reference, reading inputs from P2.5 and sending outputs to P2.6.

```
void Receive_UART_Data(char* input) {
    unsigned int i; // index for loop

    // Get the start byte
    start = EUSCI_A_UART_receiveData(EUSCI_A1_BASE);
    while (start != 'x') {
        start = EUSCI_A_UART_receiveData(EUSCI_A1_BASE);
    }

    // Get the command
    for(i = 0; i < RECEIVE_DATA_SIZE; i++) {
        *(input+i) = EUSCI_A_UART_receiveData(EUSCI_A1_BASE);
    }

    // Get the end byte
    end = EUSCI_A_UART_receiveData(EUSCI_A1_BASE);
    while(end != 'y'){
        end = EUSCI_A_UART_receiveData(EUSCI_A1_BASE);
    }
}
```

Figure 21 Receive Function of Bluetooth Communication

As shown in the figure above, the microcontroller first expects a start byte, which indicates the beginning of a new command. Then, after receiving the command body part, there should be an end byte to finish the communication.

```
void Send_UART_Data(char *str)
{
    unsigned int i = 0; // index for loop

    // Send the load
    for(i = 0; i < MAX_STRBUF_SIZE; i++)
    {
        if (str[i] != 0)
        {
            while (EUSCI_A_UART_queryStatusFlags(EUSCI_A1_BASE, EUSCI_A_UART_BUSY));
            EUSCI_A_UART_transmitData(EUSCI_A1_BASE, str[i]);
        }
        else
        {
            while (EUSCI_A_UART_queryStatusFlags(EUSCI_A1_BASE, EUSCI_A_UART_BUSY));
            EUSCI_A_UART_transmitData(EUSCI_A1_BASE, '\n');
            break;
        }
    }
}
```

Figure 22 Send Function of Bluetooth Communication

For the send function shown above, to send each byte in a string, the microcontroller first check whether the UART port is busy, and then transmit a byte, or a character through the EUSCI function defined in the driver library provided by Texas Instrument for this CPU.

Software

The GUI application was built with PySimpleGUI library on Python. There are mainly three pages: main page, customized chords page, and music sheet page. Users can select either sheet music or customized chords for their input. If they select customized chords window, another page would pop up. On this page, users can type in chords in the text box (I.e. G, Cm, etc). If they select music sheet window, a page will pop up and users can upload a text file, in which multiple chords are written.

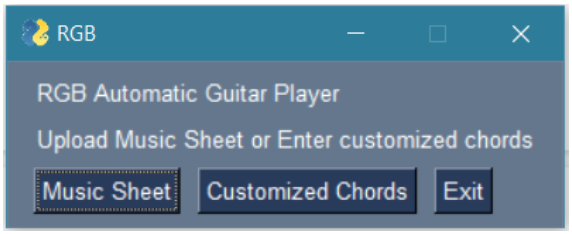


Figure 23. Main Page on GUI

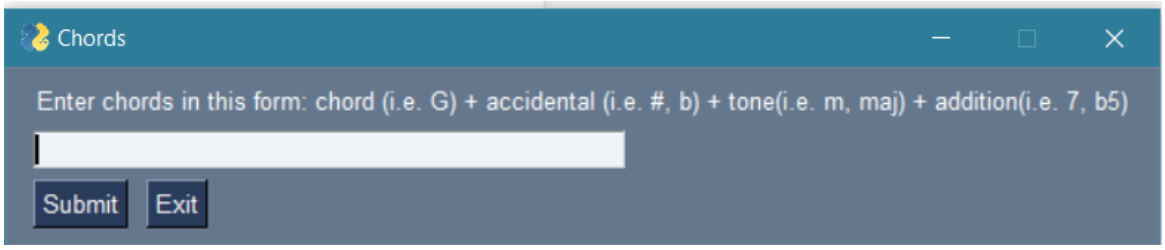


Figure 24. Chord Page on GUI

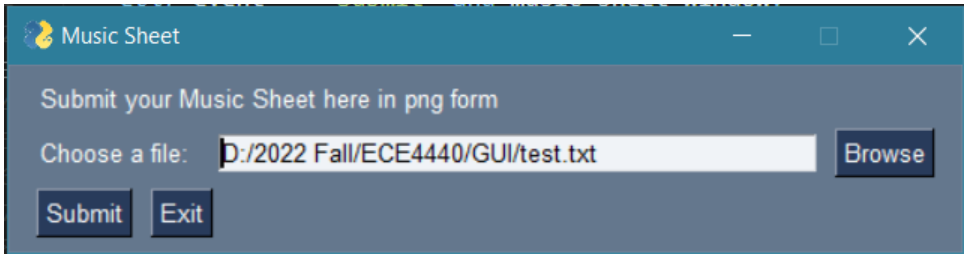


Figure 25. Music Sheet Page on GUI

The Bluetooth connection between GUI application, the Bluetooth module, and the MSP430 board was established upon launch of the GUI application. The Bluetooth function was implemented using PySerial Library, which encapsulates the access for serial port. Generally, the GUI application will send a signal “x777777y” to MSP430 and receive “INI” signal, indicating that the connection is successful. For each sent chord information (a string of length 6), there will be a “x” before it and “y” after it to indicate start point and end point (I.e. x320003y). GUI application will receive “ACK” signal, implying that the robot

has performed and finished left-hand fretting and right-hand plucking, and is ready to receive another chord information.

```python
import serial

class BLE:

    def __init__(self, COM, BAUD):
        self.serialPort = serial.Serial(port=COM, baudrate=BAUD, timeout=1)
        self.COM = COM
        self.BAUD = BAUD

    def btclient_send(self, data):
        # verify connection status
        if not self.serialPort.isOpen():
            self.serialPort = serial.Serial(port=self.COM, baudrate=self.BAUD, timeout=1)
        print("sent data: ", data)
        # send load
        self.serialPort.write(bytes(data, 'UTF-8'))

    def btclient_recv(self):
        # verify connection status
        if not self.serialPort.isOpen():
            self.serialPort = serial.Serial(port=self.COM, baudrate=self.BAUD, timeout=1)
        # receive load
        msg = self.serialPort.readline().decode('utf-8').strip("\n")
        print(msg)
        return msg
```

Figure 26. PySerial Communication

As illustrated above, the chord information is represented as a string of length 6. Each index represents a string, and the value of each character indicates which fret to press on. If the value equals 0, then it means left-hand module shouldn't perform fretting and right-hand module should perform plucking. On the other hand, if the value equals 6 and above, then it means left-hand module should perform fretting and right-hand module shouldn't perform plucking. Each chord information is accompanied with an "x" in the front, signaling the start point, and a "y" in the end, signaling the end point.

Mechanical

Guitar Frame Design

The guitar frame is designed in Fusion. Th hierarchy of the CAD design as listed list, and all original files please see provided folder.

└── Assembly

    ├── Right Hand Frame Design

    ├── Left Hand Frame Design

    ├── Guitar Drawing

└── Servo Motor Drawing

├── Normal Servo Motor Drawing

└──Actuator Header Design

Assembly is a CAD file to put everything together as following Figure 27 shown. There are several modifications to provide additional firmness in actual woodcraft, and further information please see *Mechanical* Section. Everything in CAD drawings follows the actual size of the product including the guitar fretboard. Right hand and left-hand frame manage to all servo motors without any conflictions. Because diatonic chords can be covered by the first five frets, all servo motors with actuators are used to cover the first five fret board.

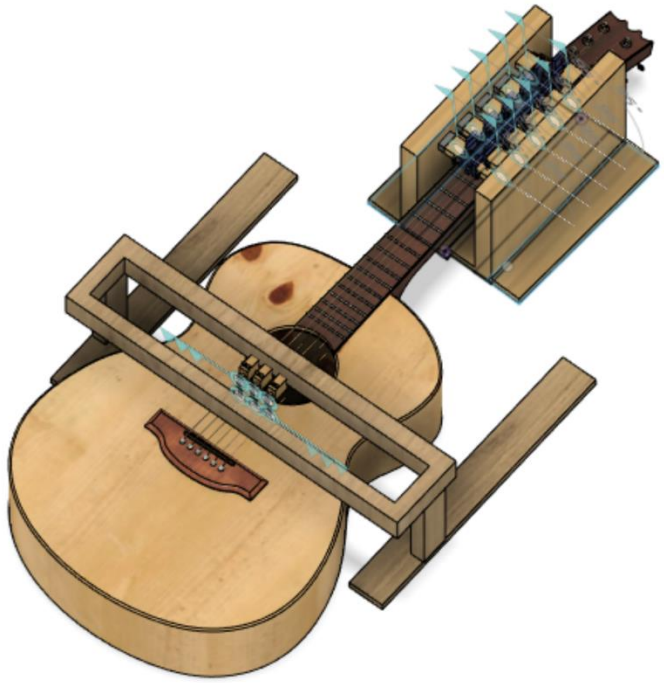


Figure 27 CAD of Entire Robot

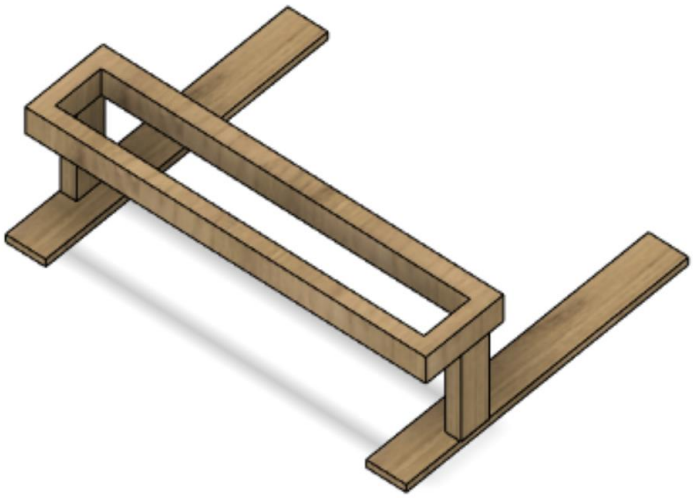


Figure 28 Right-hand plucking wood Frame

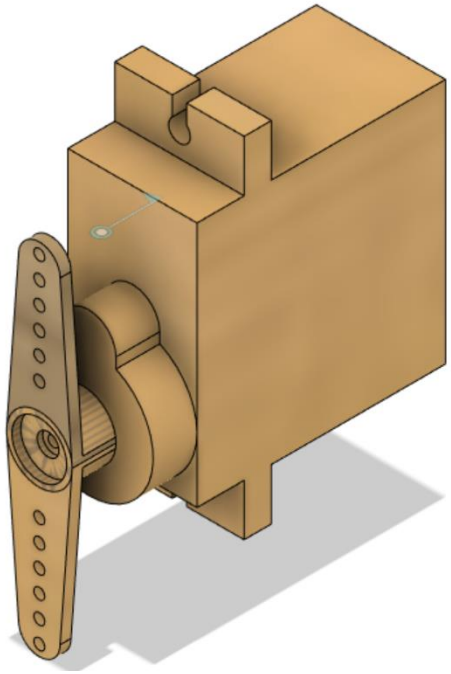


Figure 29 CAD of servo motorFfureu

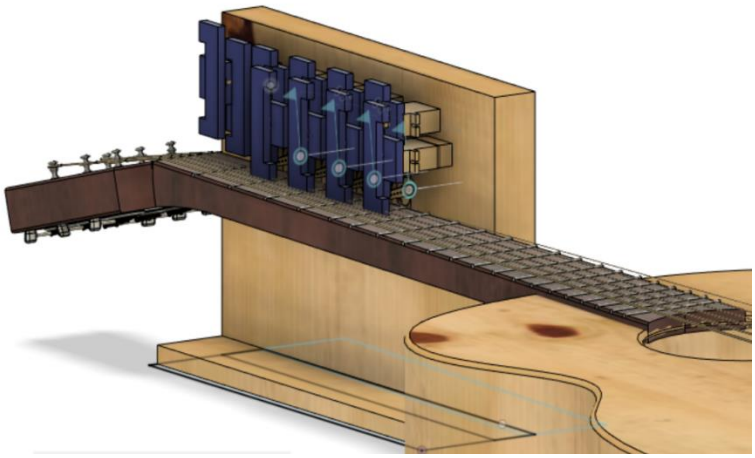


Figure 30 CAD of left-hand module

All the mechanical works were conducted in FabLab in University of Virginia School of Architecture including servo motor holder, Guitar supporter, wire management, and final assembly.

Servo Motor Holder

In both right hand and left-hand frames have servo motor holder to ensure the servo motors are in place in shown Figure 29. Each servo motor has a wood extender to improve string sound quality when the servo motor presses the string.

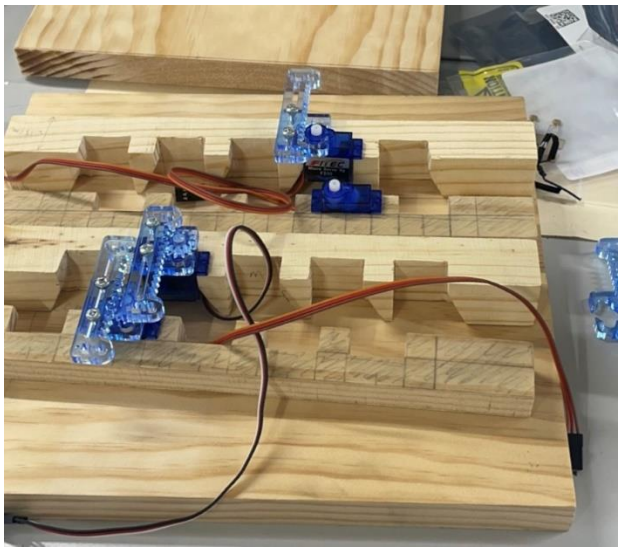


Figure 31 Left-hand Module

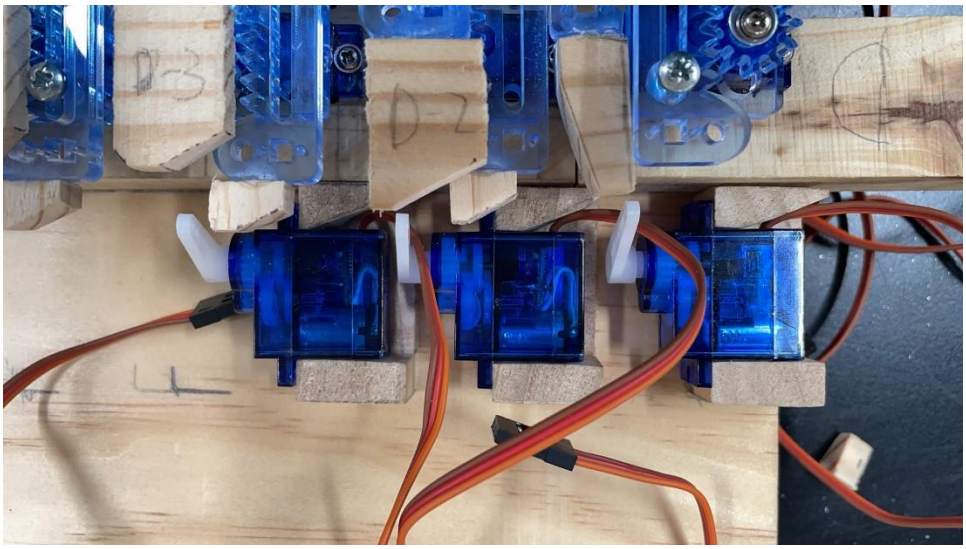


Figure 32 Extenders

Guitar Supporter

To minimize guitar vibrations, guitar supporter is used when the robot is playing music. Figure 31 prevents the guitar from sliding leftward when the servo motors press the strings. Figure 32 is a mechanism to prevent guitar and right frame over-shaking when the servo motors strum the guitar.

Figure 33 Guitar Supporter 1

Figure 34 Guitar Support 2

Connection Management

Given 136 connectors are required to connect with PCB, proper connection management is vital. The holes on the left-hand frame are drilled in Figure 33 to allow servo motors' cables go through.

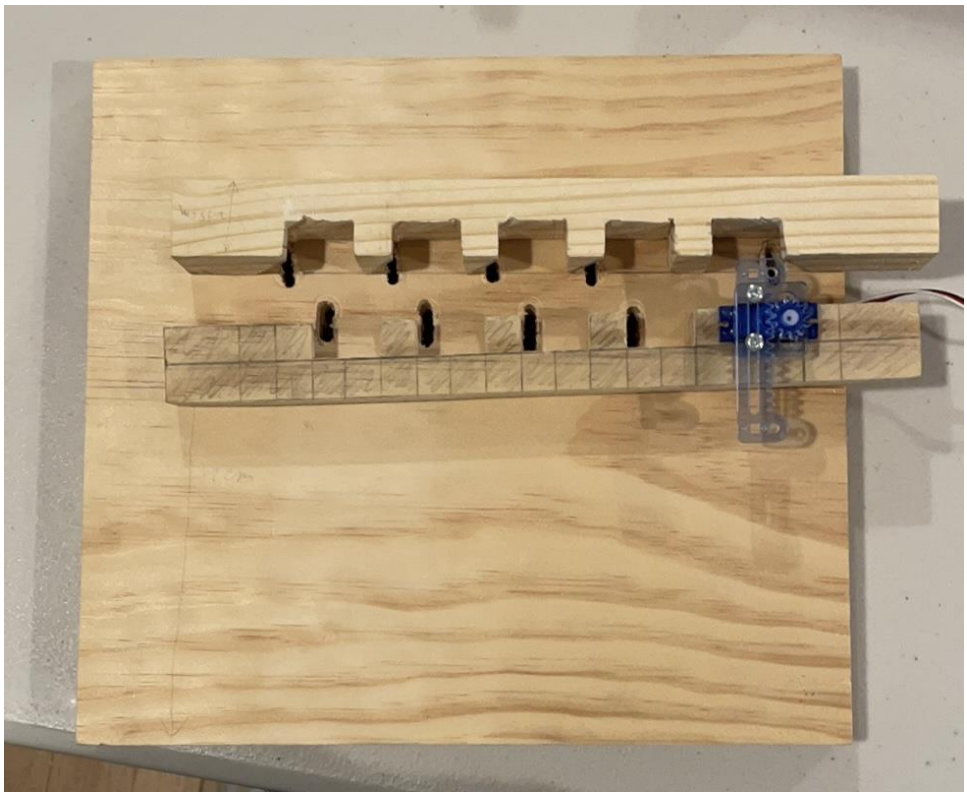


Figure 35 Single Motor Connection Management

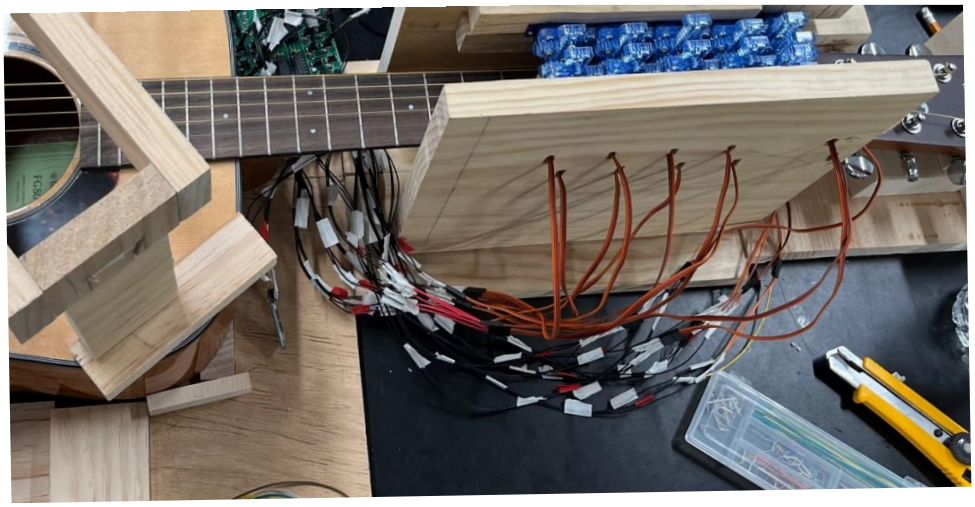


Figure 36 Multiple Motor Connection Management

Project Timeline

GanttProject Figure 35 was created to keep project in the timeline for the project. The original timeline, the timeline after the Midterm Design Review, and the final timeline are shown in Figures 35, 36, and 37. The main difference is PCB took longer to complete because PCB manufacturing took longer than a week to arrive than expected. Since wireless connections were lowered priority and left more time for building guitar frames, Bluetooth connection was postponed toward the end of the semester. Beneficial to deliver dynamic music genres, tuning the strum, plucking timing, and customizing more strumming patterns were added toward the final two weeks. Concerning overcurrent safety issues, all the servo motors were calibrated in the final week.

Based on our research background, course experience, and interests, team members' primary and secondary responsibilities are divided as follows:

Haotian Ren majors in Computer Engineering with strong programming and mechanical design backgrounds:

- Primary responsibility: Hand movements in embedded designs and PCB designs
- Secondary responsibility: Building guitar frames and Bluetooth protocol setup

Tianyue Guo majors in Computer Engineering with strong high-level programming and solid network knowledge:

- Primary responsibility: Software GUI interface and Bluetooth protocol setup
- Secondary responsibility: Hand movements in embedded designs and building guitar frame

Xuanjia Bi majors in Computer Engineering and Electrical Engineering with CAD and PCB experience and research background in embedded design:

- Primary responsibility: CAD design and building guitar frame
- Secondary responsibility: Hand movements in embedded designs and Bluetooth protocol setup
-

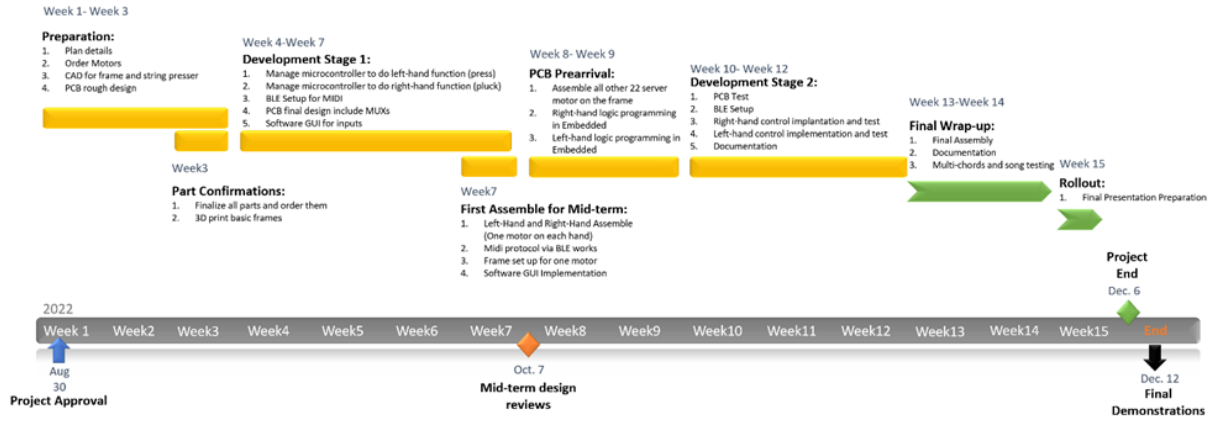


Figure 37 Gantt Chart - Original

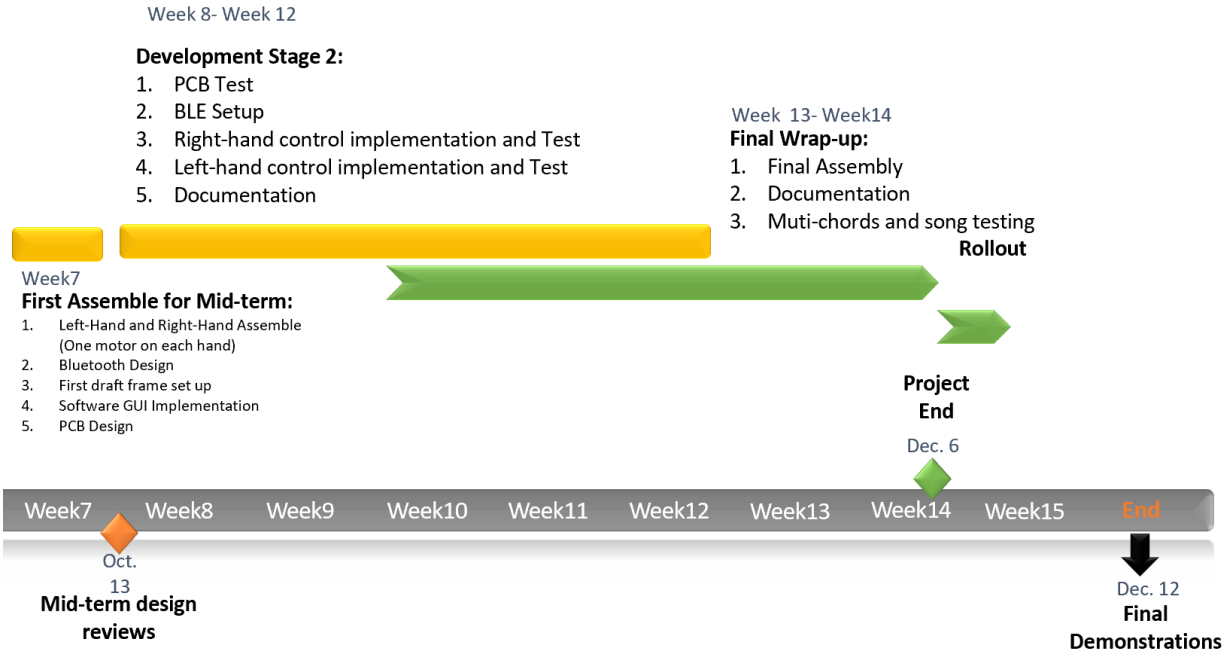


Figure 38 Gantt Chart - Midterm

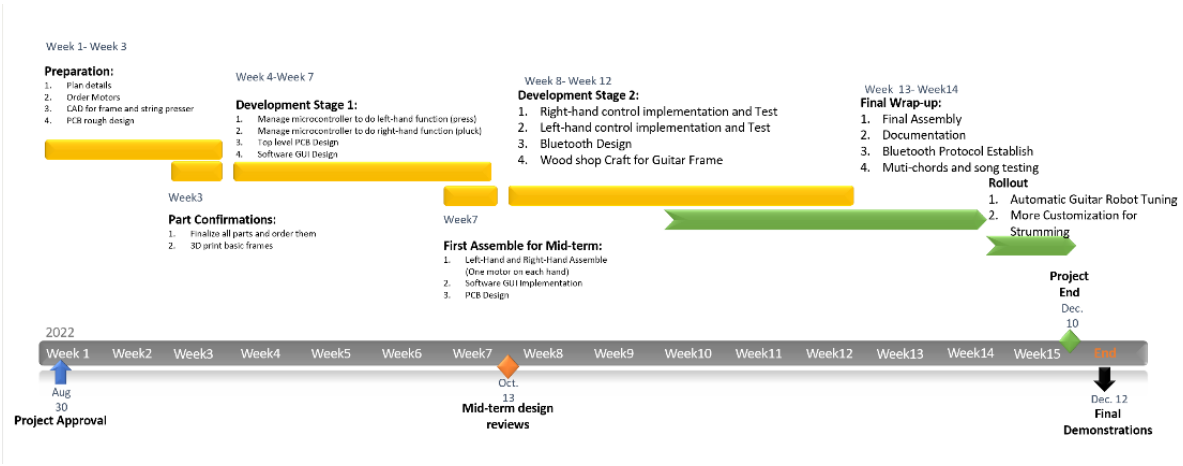


Figure 39 Gantt Chart - Final

Test Plan

The first step of testing is building wood frames for left-hand and right-hand modules. A basic GUI application was built and tested to see if the chord inputs from users (I.e., G, Cm, Bm7b5, etc.) can be received and properly translated to a string of size 6, which indicates fretting positions. The PCB board was designed on KiCad and was tested with DRC test. On MSP430, preliminary code was written and the corresponding pins according to the PCB design were tested to see if PWM signals and decoder signals came through.

Then, for the Bluetooth connection, both the connection between GUI and Bluetooth Module and the connection between Bluetooth Module and MSP430 were tested to make sure the connection works as expected. After FSM states were set up, the pins on MSP430 were tested to see if MSP430 could receive multiple inputs from GUI and output signals at correct FSM states using Code Composer Studio as IDE.

After the wood structure was built, each servo motor was tested to make sure that they can properly function as expected in the left-hand and right-hand modules. Each servo motor was connected to a PWM signal, a 5V power signal, and ground to see if they could properly fret or pluck strings. Then, the duty cycle for PWM signal of each servo motor was adjusted carefully to make sure the fretting and the plucking wouldn't cause buzzing sound.

After the PCB board was received, all the components were soldered and tested using multimeter. Then, the PCB board was tested with MSP430 to see if everything still properly functioned.

Overall, the test plan was followed with a few exceptions. When the duty cycles were being adjusted, it turned out that the wood frame needed to be modified a bit more. Some wooden parts were added to each servo motor and linear actuator in the left-hand module to make sure there was enough friction between the motors and the strings to perform fretting. This adjustment didn't affect the overall testing flow, as all the other tests carried on without problems (PCB tests).

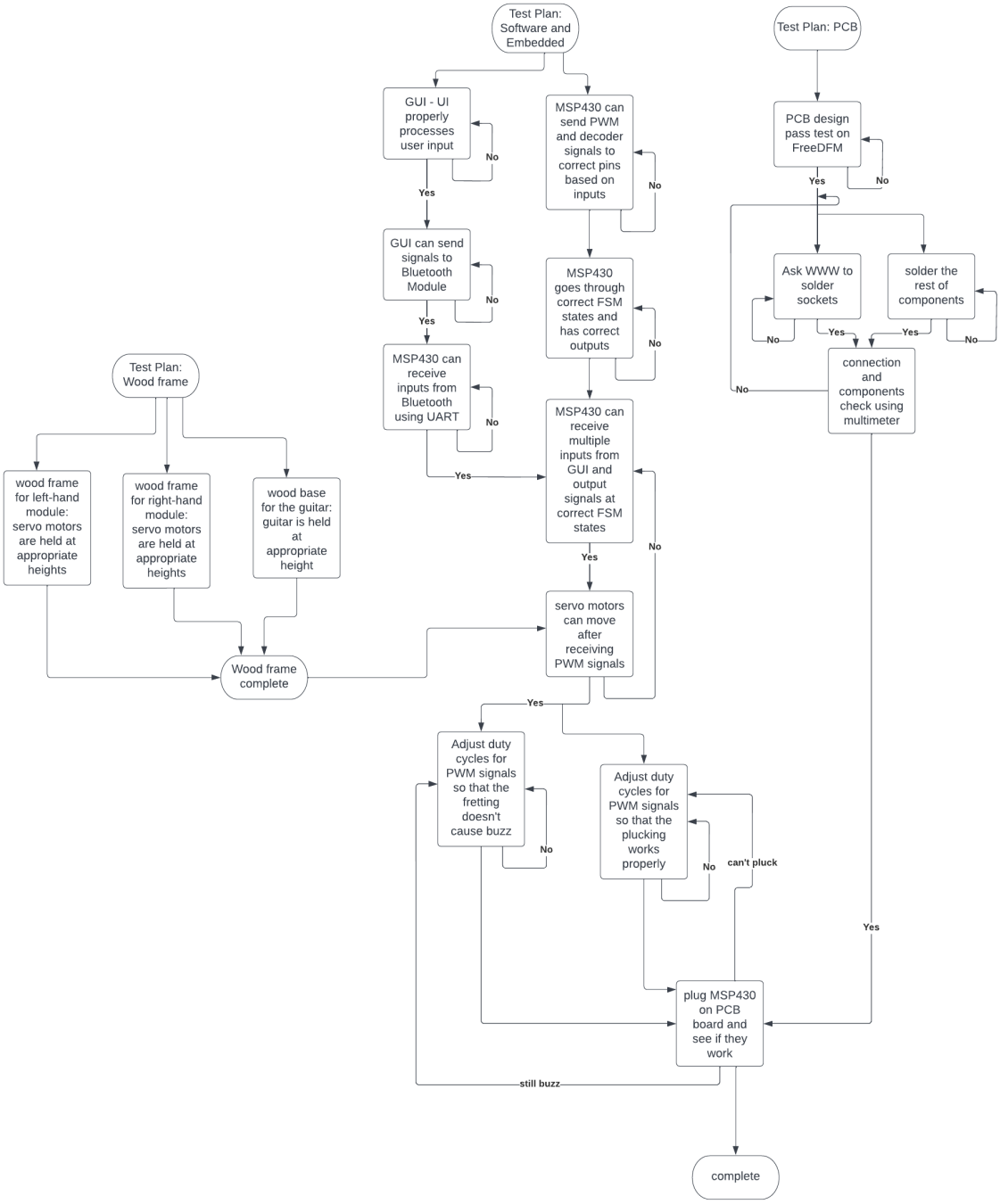


Figure 40 Test Plan Overview

Final Results

The entire system works very well. The guitar robot can play all diatonic chords in response to users' inputs on GUI application, and users can select single strumming pattern (strum once for each chord) or double strumming pattern (strum twice for each chord). The system can also play a series of chords

written in a text file. All the success criteria are passed: all the motors in the left-hand and right-hand modules function properly, the wood structure holds guitar and motors at appropriate locations, and the GUI application is user-friendly. However, there is a tradeoff in the left-hand module. There are 4 less motors than the one planned in the proposal because there are some frets that are never used in diatonic chords. Also, using less motors also fits our cost constraints since the overall cost would be lower. There are also more things built than what has been planned in the proposal, where strumming patterns can be customized, and PCB board is protected by current control.

Costs

Overall, the cost for this automatic guitar robot is $475.15 after taking account of massive production. The most expensive parts will be servo motors. Also, instead of using MSP 430, ASIC could be used in production. According to Carvalho, if 10,000 units went in production, ASIC with fixed cost around $86,000 can lower each computer chip into $10. The initial total cost for building one robot is listed below Table 4, and the total unit cost after massive production is listed in Table 5. For details about cost per part please see appendix.

Items	Total Cost
Electronic Parts	$368.59
Mechanical Parts	$53.31
PCB Manufacturing	$35.26
Embedded Cost	$17.99
Total Cost	$475.15

Table 4 Total Cost

Items	Total Cost in Approximation for Massive Production
Electronic Parts	$250
Mechanical Parts	$50
PCB Manufacturing	$30
Embedded Cost	$10
Power Supply	$20
Total Cost	$360

Table 5. Total Cost in Approximations for Massive Production

Future Work

There is some future work that can be done for improvements. Firstly, there could be more features included for both left-hand and right-hand modules. For example, the right-hand module could include a muting feature that can mute strings when receiving mute signals from users. More strumming patterns and tempo selections could also be included for our users. The left-hand module could use 6 slides

instead of 27 servo motors, in order to cover more frets and to increase flexibility and scope of the project. An auto calibration system can be included with sensors above the guitar, in order to press down on the correct frets. In addition, the servo motors currently in use are not of the best quality and might malfunction after long hours of working. Hence, motors that have better endurability and resilience can be used. Moreover, the guitar robot mainly plays guitar chords as acoustic guitar. In order to play guitars as lead guitar, the robot can be designed in a way that it can play single notes one at a time and in a smooth fashion. Finally, the frame could be designed in a more flexible way, where it can support the robot to play more stringed instruments, such as bass, electric guitars, etc. The frame can be designed as movable parts with customizable sizes.

References

[1] R. G. Smith and J. Eckroth, “Building AI applications: Yesterday, Today, and Tomorrow,” AI Magazine, vol. 38, no. 1, pp. 6–22, 2017. [Accessed: 27-Sep-2022].

[2] A. Henry and C. Fox, “Open-source hardware automated guitar player,” School of Computer Science, University of Lincoln. [Online]. Available: https://eprints.lincoln.ac.uk/id/eprint/45327/1/AutomatedGuitarICMC.pdf . [Accessed: 27-Sep-2022].

[3] S. Leigh, Guitar Machine. https://www.media.mit.edu/posts/guitar-machine/ . [Accessed: 27-Sep-2022].

[4] E. Singer, K. Larke, and D. Bianciardi, “LEMUR GuitarBot: MIDI Robotic String Instrument,” LEMUR. [Online]. Available: https://nagasm.org/NIME/NIME03/NIME03_SingerLarke.pdf . [Accessed: 27-Sep-2022].

[5] “McMaster Carr,” McMaster. [Online]. Available: https://www.mcmaster.com/solenoids/direction-of-operation~linear/. [Accessed: 27-Sep-2022].

[6] S. Yuncheng, "Research on Modeling and Design of Real-Time Embedded Systems," 2014 7th International Conference on Intelligent Computation Technology and Automation, 2014, pp. 547-550, doi: 10.1109/ICICTA.2014.138.

[7] “Pololu - FEETECH FS90R micro continuous rotation servo,” *Pololu Robotics and Electronics*. [Online]. Available: https://www.pololu.com/product-info-merged/2820. [Accessed: 13-Dec-2022].

[8] “MSP430FR2476,” *MSP430FR2476 data sheet, product information and support | TI.com*. [Online]. Available: https://www.ti.com/product/MSP430FR2476. [Accessed: 13-Dec-2022].

[9] “Micro-lock plus connector system,” *Molex*. [Online]. Available: https://www.molex.com/molex/products/family/microlock_plus_wiretoboard_connector_system. [Accessed: 13-Dec-2022].

[10] “GNU lesser general public license v3.0 - GNU Project - Free Software Foundation,” *[A GNU head]* . [Online]. Available: https://www.gnu.org/licenses/lgpl-3.0.en.html. [Accessed: 13-Dec-2022].

[11] “Monoprice.” [Online]. Available: https://downloads.monoprice.com/files/manuals/15710_Manual_170509.pdf. [Accessed: 13-Dec-2022].

[12] “Advanced circuits,” *Printed Circuit Board Manufacturer - PCB Manufacturing and Assembly*. [Online]. Available: https://www.4pcb.com/. [Accessed: 13-Dec-2022].

[13] M. L. Lopes et al., "Tolerance Studies of the Mu2e Solenoid System," in IEEE Transactions on Applied Superconductivity, vol. 24, no. 3, pp. 1-5, June 2014, Art no. 3800105, doi: 10.1109/TASC.2013.2278844. [Accessed: 13-Dec-2022]

[14] F. Sun and J. Luo, "Design and Research on Miniaturization and Lightness Servo Mot or Controller," 2021 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA), 2021, pp. 790-793, doi: 10.1109/AEECA52519.2021.9574233. [Accessed: 13-Dec-2022]

[15] “Fusion 360: 3D CAD, CAM, CAE, & PCB cloud-based software,” *Autodesk*, 05-Dec-2022. [Online]. Available: https://www.autodesk.com/products/fusion-360/overview?term=1-YEAR&tab=subscription. [Accessed: 13-Dec-2022].

[16] “Ultimaker Cura 4.3: Available now,” *https://ultimaker.com*. [Online]. Available: https://ultimaker.com/learn/ultimaker-cura-4-3-available-now. [Accessed: 13-Dec-2022].

[17] “KiCad Eda,” *Schematic Capture & PCB Design Software*. [Online]. Available: https://www.kicad.org/. [Accessed: 13-Dec-2022].

[18] “Matlab,” *MathWorks*. [Online]. Available: https://www.mathworks.com/products/matlab.html. [Accessed: 13-Dec-2022].

[19] “Waveform free,” *Tracktion*. [Online]. Available: https://www.tracktion.com/products/waveform-free. [Accessed: 13-Dec-2022].

[20] S. K, “Analog discovery 2,” *Analog Discovery 2 - Digilent Reference*. [Online]. Available: https://digilent.com/reference/test-and-measurement/analog-discovery-2/start. [Accessed: 13-Dec-2022].

[21] K. C. Chung and M. J. Shauver, “Table saw injuries: Epidemiology and a proposal for preventive measures,” *Plastic and reconstructive surgery*, Nov-2013. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4154236/. [Accessed: 13-Dec-2022].

[22] L. S. U. (LSU), “Drill Presser,” *LSU Mechanical & Industrial Engineering*. [Online]. Available: https://www.lsu.edu/eng/mie/cuf/ammf/safetyrules/DrillPress.php. [Accessed: 13-Dec-2022].

[23] R. H. Todd, “Manufacturing Processes Reference Guide,” *Google Books*. [Online]. Available: https://books.google.com/books?id=6x1smAf_PAcC. [Accessed: 13-Dec-2022].

[24] Chris Baylor. "Belt Sanders - Woodworking Tools - How to Use a Belt Sander". thesprucecrafts. [Accessed: 13-Dec-2022]

[25] Broun, Jeremy (1989). The Incredible Router. Lewes, East Sussex: Guild of Master Craftsman Publications. ISBN 0-946819-17-3.

[26] Harrabin, Roger (1 September 2017). "Sales of inefficient vacuum cleaners banned". BBC News. [Accessed: 13-Dec-2022]

[27] P. S. Kamble, S. A Khoje and J. A Lele, "Recent Developments in 3D Printing Technologies: Review," 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), 2018, pp. 468-473, doi: 10.1109/ICCONS.2018.8662981. [Accessed: 13-Dec-2022]

[28] Bralla, James G. Handbook of Manufacturing Processes - How Products, Components and Materials are Made Industrial Press, 2007 page 297 [Accessed: 13-Dec-2022]

[29] McComb, Gordon; Shamieh, Cathleen (2009), Electronics for Dummies (2nd ed.), For Dummies, p. 251, ISBN 978-0-470-28697-5. [Accessed: 13-Dec-2022]

[30] S. WOOD., “INSULATED WIRE STRIPPING DEVICE.” [Accessed: 13-Dec-2022]

[31] “CCSTUDIO,” CCSTUDIO IDE, configuration, compiler or debugger | TI.com. [Online]. Available: https://www.ti.com/tool/CCSTUDIO. [Accessed: 13-Dec-2022].

[32] “Welcome to pySerial's documentation,” Welcome to pySerial's documentation - pySerial 3.4 documentation. [Online]. Available: https://pyserial.readthedocs.io/en/latest/. [Accessed: 13-Dec-2022].

[33] NMinion, “Bluetooth serial terminal for Windows 10,” Download.com. [Online]. Available: https://download.cnet.com/Bluetooth-Serial-Terminal-for-Windows-10/3000-2121_4-77571575.html. [Accessed: 13-Dec-2022].

[34] “Python release python 3.10.0,” Python.org. [Online]. Available: https://www.python.org/downloads/release/python-3100/. [Accessed: 13-Dec-2022].

[35] “Python guis for humans,” *PySimpleGUI*. [Online]. Available: https://www.pysimplegui.org/en/latest/. [Accessed: 13-Dec-2022].

[36] “Python - python serial communication (pyserial),” DevTut. [Online]. Available: https://devtut.github.io/python/python-serial-communication-pyserial.html[Accessed: 13-Dec-2022].

[37] “Time - Time Access and conversions,” Python documentation. [Online]. Available: https://docs.python.org/3/library/time.html. [Accessed: 13-Dec-2022].

[38] “Pathlib - object-oriented filesystem paths,” Python documentation. [Online]. Available: https://docs.python.org/3/library/pathlib.html. [Accessed: 13-Dec-2022].

[39] “Pyinstaller manual,” PyInstaller Manual - PyInstaller 5.7.0 documentation. [Online]. Available: https://pyinstaller.org/en/stable/. [Accessed: 13-Dec-2022].

[40] “Regulations, mandatory standards and bans,” U.S. Consumer Product Safety Commission. [Online]. Available: https://www.cpsc.gov/Regulations-Laws--Standards/Regulations-Mandatory-Standards-Bans. [Accessed: 27-Sep-2022].

[41] M. Anderson and S. L. Anderson, “Machine Ethics: Creating an Ethical Intelligent Agent”, AIMag, vol. 28, no. 4, p. 15, Dec. 2007. [Accessed: 13-Dec-2022].

[42] Abowd, John M, 2017, “How Will Statistical Agencies Operate When All Data Are Private?”, Journal of Privacy and Confidentiality, 7(3): 1–15. doi:10.29012/jpc.v7i3.404 [Accessed: 13-Dec-2022].

[43] A. Lonzetta, P. Cope, J. Campbell, B. Mohd, and T. Hayajneh, “Security vulnerabilities in bluetooth technology as used in IOT,” Journal of Sensor and Actuator Networks, vol. 7, no. 3, p. 28, 2018. [Accessed: 13-Dec-2022].

[44] A. Pipino, A. Liscidini, K. Wan and A. Baschirotto, "Bluetooth low energy receiver system design," 2015 IEEE International Symposium on Circuits and Systems (ISCAS), 2015, pp. 465-468, doi: 10.1109/ISCAS.2015.7168671. [Accessed: 13-Dec-2022].

[45] Legal Information Institute. (n.d.). 16 CFR § 1500.3—Definitions. Retrieved December 14, 2022, from https://www.law.cornell.edu/cfr/text/16/1500.3 [Accessed: 13-Dec-2022].

[46] White, C., Qureshi, A., Singh, V., Krager, J., Porlier, J., Wood, K., & Crawford, R. (n.d.). Modular automated assistive guitar. https://patents.google.com/patent/US7285709 [Accessed: 13-Dec-2022].

[47] Vignolo, C. (n.d.). SELF-PLAYING ROBOT GUITAR COMPRISING A BIODEGRADABLE SKIN-LEATHERN FORMED CARCASS AND A BIODEGRADABLE SKIN-LEATHERN FORMED MUSICAL PLECTRUM, AND PROTEIN / AMINO ACIDS. https://patents.justia.com/patent/20140165814 [Accessed: 13-Dec-2022].

[48] Caulkins, K., & Caulkins, J. (n.d.). Apparatus for automating a stringed instrument. https://patents.justia.com/patent/6166307 [Accessed: 13-Dec-2022].

[49] "MSP430FR4XX and MSP430FR2xx Family User's guide (rev. I) - ti.com." [Online]. Available: https://www.ti.com/lit/ug/slau445/slau445.pdf. [Accessed: 13-Dec-2022].

Appendix

Item	Qty	Descriptions	Vendor	Manufacuring Number	Unit Price	Total Price
1	11	Capacitors, 0.1uF	Digikey	K104K15X7RF5UL2	$ 0.23	$ 2.53
2	2	Capacitors, 1uF	Digikey	K105K20X7RF55H5	$ 0.24	$ 0.48
3	1	Capacitors, 10uF	Digikey	4191-PW2H101MNN1836A4ES-ND	$ 3.79	$ 3.79
5	38	Micro-lock Sockets, 3 pins 1.20mm	Digikey	5055680371	$ 0.78	$ 29.64
6	1	Micro-lock Sockets, 2 pins 1.20mm	Digikey	5055680271	$ 0.79	$ 0.79
7	1	Barrel Jack Switch	Digikey	1568-PRT-10811-ND	$ 1.05	$ 1.05
8	41	TestPoint	Digikey	534-5000	$ 0.36	$ 14.76
9	8	Ressitors, 10kOhm	Digikey	2197-283-10K-RC-ND	$ 0.25	$ 2.00
10	1	Ressitors, 4.7kOhm	Digikey	2197-283-4_7K-RC-ND	$ 0.30	$ 0.30
11	1	Power Supply Unit	BestBuy	NS-AC3000	$ 24.99	$ 24.99
12	6	Decoder	Moser	CD74HC238EE4	$ 0.70	$ 4.20
13	2	Buffer	Moser	SN74HC126N	$ 1.16	$ 2.32
14	1	Power Switch ICs	Mouser	AP22652W6-7	$ 0.49	$ 0.49
15	1	Voltage Regulator	Mouser	ADP121-AUJZ33R7	$ 1.36	$ 1.36
16	35	Servo Motor	Digikey	1927-2595-ND	$ 6.31	$ 220.85
17	20	Micro-lock Cables, 3 pins 1.20mm	Digikey	WM17170-ND	$ 2.57	$ 51.40
18	1	Micro-lock Cables, 2 pins 1.20mm	Digikey	WM17163-ND	$ 2.64	$ 2.64
19	1	MSP430FR2476 Lauchpad	Digikey	296-53618-ND	$ 17.99	$ 17.99
						$ 381.58

Appendix-A Total Project Costs for Electronics and Embedded Design