ECE 445: Auto-Played Guitar

Design Document

Team 12: Peilin Rao (peilinr2), Jiyu Hu (jiyuhu2), Qianlu Chen (qianluc2)

1. Introduction

Problem and Solution Overview

We will build an auto-played guitar for those people or places that need it for the purpose of demonstration or entertainment. Our customers face a lot of situations that an auto-played guitar can help: Some people love the original sound of guitar music but lack the time and energy to practice it; Just like the existence and common use of auto-played piano in the lobby of grand hotels[1], some places may need a playing guitar for entertainment or creating atmosphere; Guitar stores want to show the good sound quality of the guitars they are selling but it is usually very expensive to hire someone to play for a long period of time; Music creators want to hear the sound of their customized guitar for testing or for remix; Some people may want to hear the authentic sound of guitar music as their wake up alarm; Some new guitar learners wish to hear a demonstration of the musical piece that they are practicing on while watching the notes they should play. Although the idea of listening to guitar music can be easily realized by using music players, the beautiful original sounds from different real guitars are irreplaceable. Our product solves our customer's problem by presenting the auto-played guitar. Although the idea of auto-played guitar, just like the auto-played piano that many hotels have for lobby decoration, is not an innovation, all the existing auto-played guitars are expensive to make, too ponderous to be portable, and the auto-played unit has no compatibility to be reinstalled on a variety of guitars. Therefore, the pivot of our design is to build a piece of automatic guitar playing unit that is affordable, portable and compatible with any type of guitar.

Visual Aid

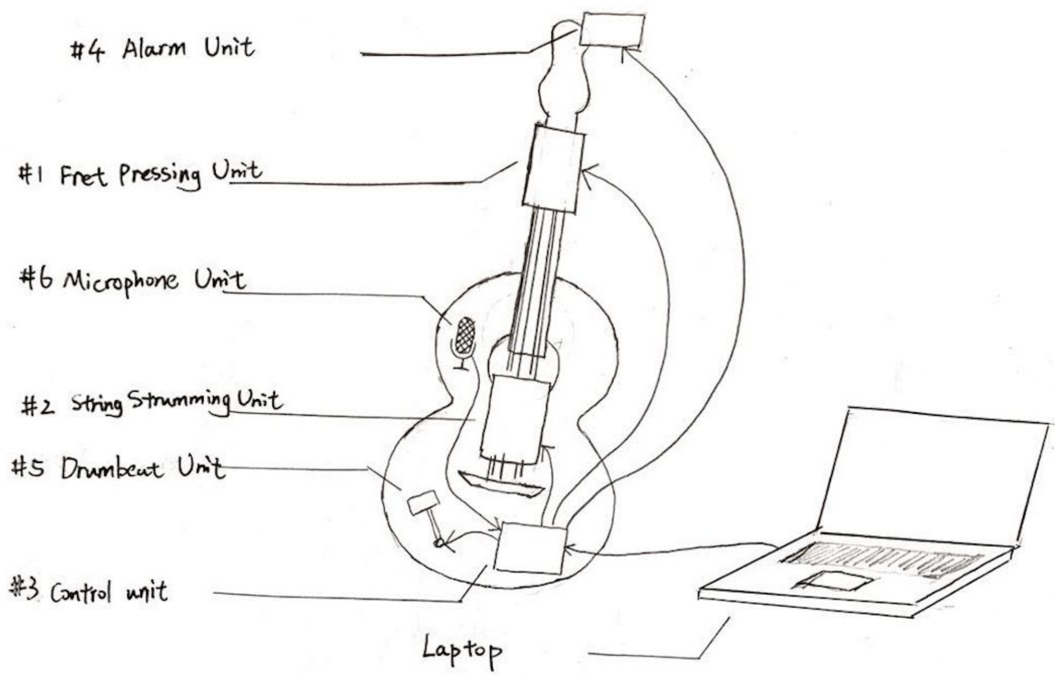


Fig. 1: Visual description

The six units of our design should be able to attach to any type of guitar as the above graph shows. The Control Unit will be programmed through the laptop, and is also in charge of sending records from the Microphone Unit to the laptop. The Control Unit and Microphone Unit all connect to the laptop through USB cables.

High-level requirements

- Be able to play the correct notes on the guitar in tune (an electric tuner should show the notes from auto-played guitar has within 5% discrepancy with the theoretical value) and loudly.
- Be able to deliver the drumbeat and guitar note at the correct rhythm, with adjustable tempo.
- Alarm functionality with synchronized time in the guitar.
- *Additional: Be able to record a piece of music with the microphone and replay with auto-played guitar

2. Design

Block Diagram:

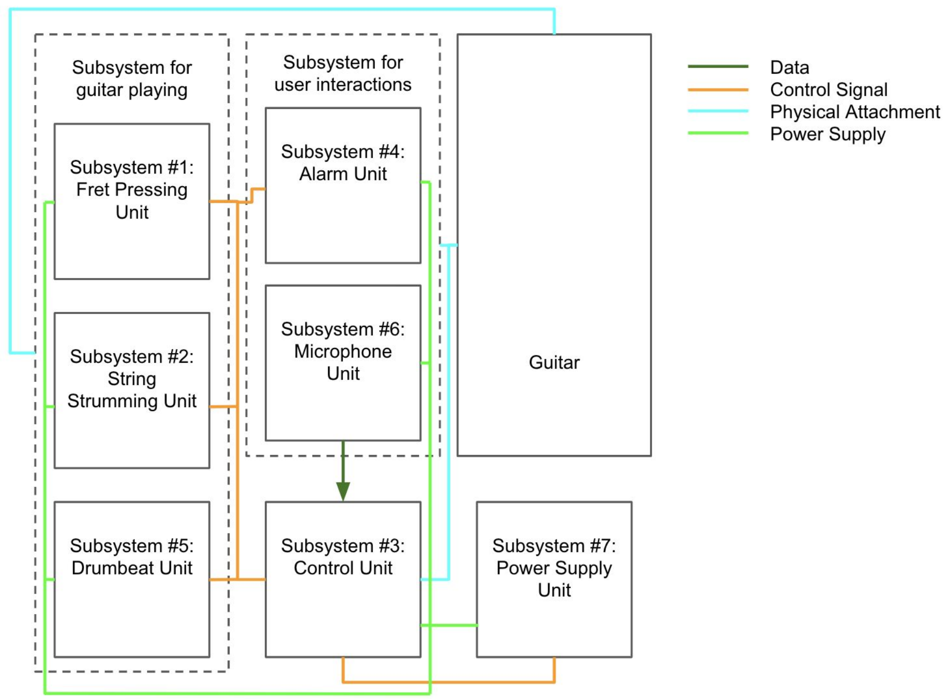


Fig. 2: Block diagram

As illustrated in the figure, a high level design of our system consists of mainly four parts: Subsystem for guitar playing, Subsystem for user interactions, Control unit and Power supply unit. All other Subsystems are controlled collectively by Subsystem #3 Control Unit, which is attached to the body of the guitar. Subsystem for guitar playing, as its name indicates, is responsible for playing music at the correct pitch and rhythm. Included modules are: Subsystem #1 Fret Pressing Unit: installed on the fretboard (neck) of the guitar for pressing and releasing the strings; Subsystem #2 String Strumming Unit: installed on the body of the guitar for strumming and making sound of the strings. These two units work together to play clear and correct notes according to the program; Subsystem #5 Drumbeat Unit: on the bottom of the guitar, it can periodically strike the guitar body to make drum beats. This unit outputs the tempo and adds some taste to the music being played. In Subsystems for User Interactions, Subsystem #4 Alarm Unit is on the head of the guitar with a LED screen for the user to present the time for alarm (play the guitar). The Control Unit keeps synchronized time and compares that with the set alarm time for the alarm functionality. Subsystem #6 Microphone Unit is **an optional part** installed on the guitar body for recording and sound input, it should also interact with the control unit, which will analyze the signal and replay the music according to our plan. Subsystem #7 Power Unit should provide reliable working voltage for all the other parts.

Physical Design

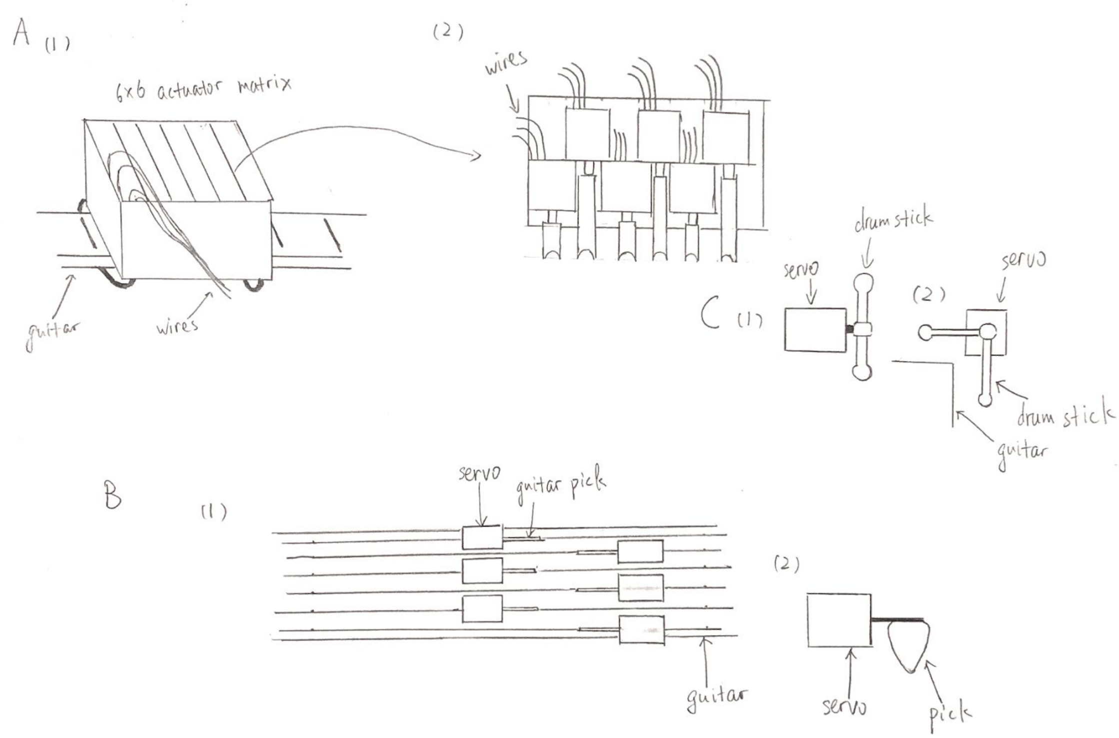


Fig. 3: Physical design

In this section, we will introduce the physical design of the auto-played guitar. We will mainly discuss three parts that need to be specifically explained with physical illustrations.

A. Fret Pressing Unit

   This unit is used for pressing the guitar strings. It contains a 6x6 actuator matrix. The overall unit figure is depicted in figure A(1). This unit looks like a box attached to the neck of the guitar. The box contains 6 rows, each row consists of 6 actuators attached to a 3-D printed plastic board. Currently, we plan to use glues for actuator attachment, but after a simple actuator testing, the actuator generates heat when functioning. Our concern is that the heat of the actuator will melt the glue and detach itself from the board. If this happens, we will switch to use screws for fixing the actuators on the board. Another thing worth noting is that the width of the actuator body is larger than the distance between two adjacent strings on the guitar. Therefore, we adopt a dislocated placement for the 6-actuator alignment, as illustrated in figure A(2) above. Each actuator tip is connected to a small stick with half circle opening on the tip for better pressing the guitar string.

B. String Strumming Unit

   This unit consists of 6 servos each connected with a guitar pick on its end. Figure B(1) shows the placement of the servos on the guitar: the servos are placed in two rows. Row1 takes care of string 1, 3, 5, and row 2 takes care of the other 3 strings. This placement is made out of the concern of the limited space between two adjacent strings. Figure B(2) is a single servo unit looked at from its side. The guitar pick is connected to the end of the servo, and when the servo spins, the pick strikes the string.

C. Drumbeat Unit

   This unit is made of a single servo with a specially designed head. The “L” shape servo head is used for hitting both the surface and the side of the guitar. This would produce two different kinds of sounds, thus adding more flavor to the music played. This unit is usually placed at the position illustrated in C(2) when not in use. When a signal for hitting the guitar surface is triggered, the servo spins toward the guitar surface so that one of the “L” head tips hits the surface. When the guitar side signal arrives, the servo spins in the opposite direction, making the “L” head hit the guitar side.

Subsystem #1: Fret Pressing Unit

This unit is in charge of controlling the actuators attached to the fretboard of the guitar, and it is a crucial part to decide whether the auto-play guitar can play each notes correctly and pellucidly [4]. The initial goal for this unit is to be able to play relatively complex music pieces on the first six frets of the guitar. Given that there are 6 strings on a regular guitar, this unit need to drive at least 6 * 6 = 36 actuators that are hung on the top of the fretboard, pressing or releasing the strings on each fret. If time allows and everything goes smoothly, the project will support more than just 6 frets. The actuators chosen are Sparkfun Push Actuators ROB 11015 (Fig. 4)[4], it is small, affordable and provides large enough force for pressing the strings.

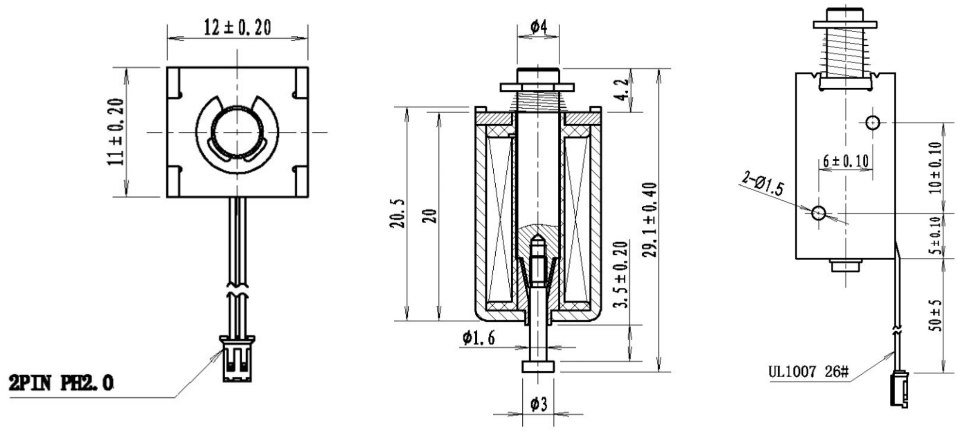


Fig. 4: Actuator ROB 11015

After some testing, when the actuator is installed right above the fret bar, this force is large enough to press the strings. Beside, This part has a working voltage of 5V and working current of 1.1A. Since there is no way to have 36 I/O pins on the ATMega, we are using 3 MCP23017(I2C 16-input/output port expanders) [6] so that we can control 36 actuators with limited I/O pins.

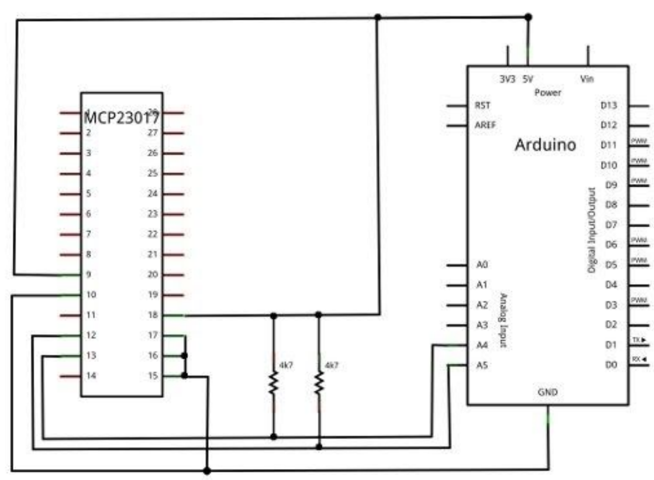


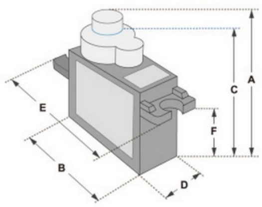
Fig. 5: MCP 23017

The above diagram is an example showing how MCP23017 can be used to extend I2C I/O pins. The red pins on MCP23017 are all extensive useful I2C I/O pins while the arduino code can use analog input A4, A5 to specify the addresses and values of the extra pins.

Requirements	Verification
The actuators should be hanged precisely and stably fixed on the correct place above the guitar neck. The distance between adjacent actuators is **6.64mm ± 0.2 mm** (the distance of adjacent strings on a guitar). The unit is fixed **8.0mm ± 0.2mm** above the guitar neck. The unit should not move after functioning.	A. After all actuators are assembled onto the frame box, use a Vernier caliper to measure distance between each adjacent actuator is **6.64mm ± 0.2 mm.** B. Fix the frame box onto the guitar neck. Measure the distance between the unit and guitar neck (**8.0mm ± 0.2mm**). C. Activate each actuator sequentially for 100 times, perform A. and B. again to verify that the unit has not moved.
Most people can easily tell the discrepancy between theoretical tone and real tone when they are different by a semitone. So we are limiting this discrepancy to be less than half of a semitone to ensure every tone sounds correct. Typically, the range of a semitone is about **16Hz**. Therefore, our tone discrepancy should be within **4Hz**.	A. Activate 1 actuator and pluck at the corresponding string. Measure the produced sound with a tuner. Verify that the sound is within **±4Hz** of the standard note. B. Perform A. for all 36 actuators to verify that the unit plays notes on tune.
The 36 actuators should be able to work individually without breaking the consistency of music. It is reasonable to require the function period on the same actuators should take less than **0.5s±0.05s.**	A. Use a slow motion camera to film several function cycles of the actuators and calculate the mean of time taken for push and release actions of the actuator. B. Perform A. for all 35 actuators to verify that actuators and press the string on time.

Subsystem #2: String Strumming Unit

This unit is the part responsible for strumming strings and getting quality sound out of the guitar. The main components of this unit is the signal amplifier and 6 servos (SG90 9G Micro Servo Motor) [5]. Each servo is connected to a guitar pick that is in charge of strumming one string. Other than the power supply, each servos should receive an analog signal from the control unit indicating how much and how fast it should turn. There should be at least 4 preset behaviors for servos: quickly strumming, slowly strumming, staying on the string (to mute the sound), and stop in the air (do nothing). It will also instantly react to the analog signal from the control unit.



Dimensions & Specifications
A (mm) : 32
B (mm) : 23
C (mm) : 28.5
D (mm) : 12
E (mm) : 32
F (mm) : 19.5
Speed (sec) : 0.1
Torque (kg-cm) : 2.5
Weight (g) : 14.7
Voltage : 4.8 - 6

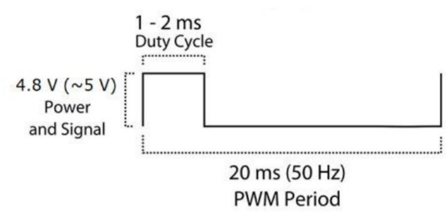


Fig. 6: Servo motor SG90

After doing some experiments, we verified that the torque it provides is large enough to play the string loudly. When working on the correct voltage, it can be controlled by a PWM wave to specify the angle of rotation. However, since we are using 6 servos on this unit and one servos on the drumbeat unit, we are using a PCA9685 [7] servo driver to provide control for all 7 of them. The graph below shows how an PCA9685 can work with Adruino to control up to 12 servos.

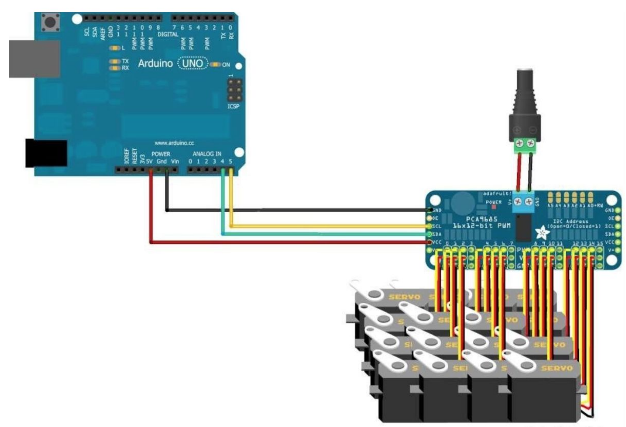


Fig. 7: PCA9685

Requirements	Verification
The servos should be working and attached firmly above the strings on the guitar body. The displacement of each actuator after a strike is less than **0.2mm**. The tips of the guitar picks should be **0.5mm ± 0.2 mm** below the string when strumming.	A. Fix the servos on the guitar. Verify that the guitar pick tips can touch the string. B. Activate each servo to pluck at the string sequentially for 100 times, perform A. again to verify that the unit has not moved.

Each servo should respond fast and correctly (less than **0.2s ± 0.05s**) to the signal of rotation. Up to six servos can be triggered by the same signal at the same time (less than **0.1s ± 0.02s** time discrepancy)	C. Use a slow motion camera to film several function cycles of the servos and verify that the servo can respond within **0.2s** after the signal is given. D. Use a slow motion camera to film 6 servos triggered with the same signal. Verify that the time lapse between the servos is less than **0.1s.**
The sound played should be loud enough (greater than **60 dB**)	Let the servos pluck at the strings, measure the volume of the sound with a sound meter. Verify that the sound is greater than **60 dB**.

Subsystem #3: Control Unit

The Control Unit is the pivotal part of the project. It is an ELEGOO Mega 2560 R3 board (embedded with ATMega 2560) that provides the signals to activate the Fret Pressing Unit, String Strumming Unit and Drumbeat Unit. Also it should be programmed from a laptop to import different music pieces. It takes in data from the Alarm Unit to determine when to play what piece of music as the alarm. It also controls the Microphone Unit in terms of recording and sending the data back to the laptop for analysis.

Requirements	Verification
When the fret pressing unit, string strumming unit, drumbeat unit and alarm unit is correctly figured, the produced music should be correct and consistent. And the ATMega should operate at **16 MHz.** To ensure the consistency of music pieces, there should be less than **0.1s** delay between signals when activating the same piece, such as actuators and servos.	The correct behavior of this unit is verified by all the other units performing correctly. Not separate verification needed for the Control Unit.
Optional: it should save the records from microphone unit	When the recording from the microphone unit ends, the control unit should process the sound signal and save it as an mp3 file to the laptop signal. The mp3 file should be clear enough in terms of replaying the music.

Subsystem #4: Alarm Unit

The Alarm Unit is one of the advanced functionalities of the project. It puts a small LED screen on the head of the guitar to show the alarm time and two push buttons for adjusting that time. A user can use the push buttons to set the alarm. When there is a change of the time, the Alarm Unit will send the updated time as data into the Control Unit. When the system time of the Control Unit matches the set alarm time, the guitar will play a piece of preset music. Pushing any of the two buttons while the music is playing will stop the performance.

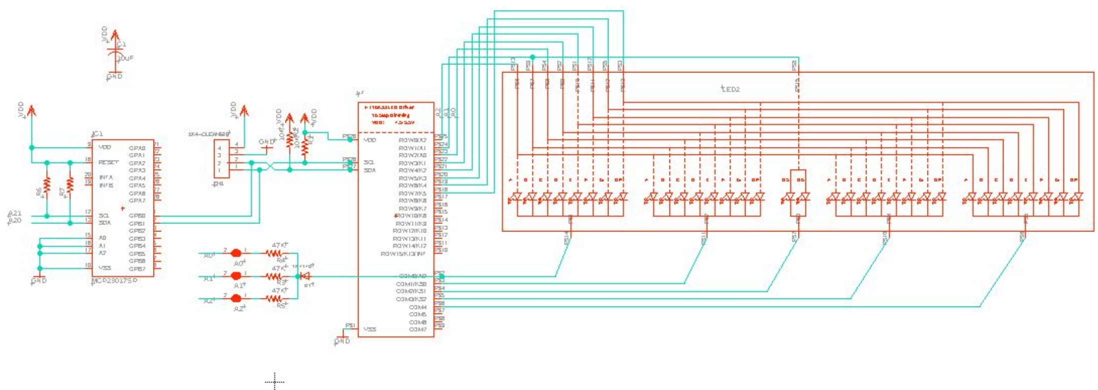


Fig. 8: LED schematic diagram

This is the schematic for an example for Arduino connecting the LED part. From left to right sequentially is the MCP23017 I2C I/O extender, the HT16K33 LED Driver[8], and the LED. We would use Pin 21 and 20 on the Arduino Mega for sending out control signals to the LED. The address is 0x70.

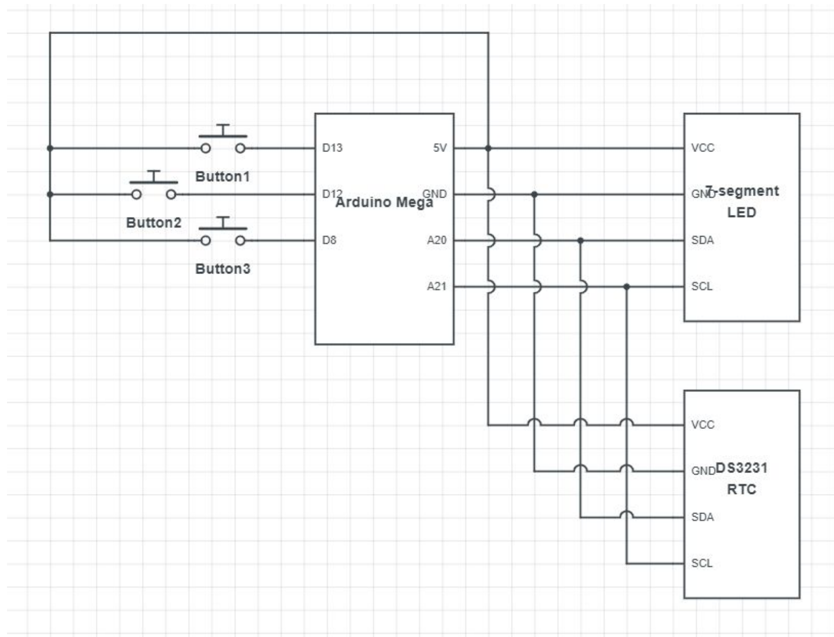


Fig. 9: Alarm Unit schematic

This is the block diagram for RTC and LED and buttons. Both the RTC and the LED will be using the I2C for communication. We will be using a DS3231 RTC chip kit developed by DIY More and directly leave connection pins for inserting. There will be three buttons in all. The mode button(Button 1) will switch between 0: Show time mode, 1: Adjust alarm hour mode, 2: Adjust alarm minute mode. The add button (Button 2) and the minus button (Button 3) will be used to adjust the hour/minute number accordingly.

Requirements	Verification
Colon on LED should blink every 500ms.	Connect to Arduino and print time stamp to serial port every time we send a signal for the colon to blink. Do this for 30 seconds and calculate the time interval through the time stamp.
After pressing Button 1, the LED should respond in 500ms and change mode.	Connect to Arduino and print time stamp to serial port every time it receives and responds to a push signal. Press the button for 10 seconds, calculate the time interval through the time stamp.

Button 2 and 3 should change nothing in Mode 0. In Mode 1 and 2, Button 2 and 3 should add or subtract the number by 1 every 500ms.	Connect to Arduino and print time stamp every to serial port every time it receives and responds to a push signal. Press the button for 10 seconds, calculate the time interval through the time stamp, and the number should add by 20 or minus by 20.
The alarm will go off as set time and date once it reaches the same hour and minute. The difference should be no bigger than 1 second.	A. Set the time for 1 minute later by date of month and wait for it to start the guitar at the exact time. B. Set the time for 1 minute later by day of week and wait for it to start the guitar at the exact time. C. Set the time for 10 hours later by date of month and wait for it to start the guitar at the exact time.

Subsystem #5: Drumbeat Unit

The aim of this unit is to put a servo that connects to a drumstick on the guitar body. It would hit the guitar surface at a certain frequency while playing guitar music. The drum beat serves as an adjunction and enriches the vividity of the performance, while it can be used as a metronome for guitar practicing. It can be activated by the programs in the control unit. Also the angle and speed of servo rotation should be directly controlled by the analog signal from the Control Unit.

Requirements	Verification
The tempo of the drum beats should be correct (tempo discrepancy ± **5%** of time gap).	Record the tempo as an audio file and analyze the time difference between each peak. Verify that the drum beats are at the correct frequency.
Should be able to provide two kinds of drumbeat (hit on the surface of the guitar & hit on the side of the guitar). The two kinds of drumbeat should be clear	A. Hit the guitar with this unit on both guitar body and guitar side. Verify that the produced sound is different B. Measure several drum beats with a sound

enough to tell from each other. Both sounds should reach at least **70 dB**.	meter to verify that the volume of the sounds are greater than **70 dB**.

Subsystem #6: Microphone Unit

This unit can record the music piece from the guitar and send this music data back into the laptop. The purpose for this unit is for the music writer or singer who wants a record of real guitar sound or a mixture of guitar and other instrument while the guitar is automatically playing.

Requirements	Verification
Optional: This unit should be able to record the play music.	**Optional:** The recording is clear and recognizable as the original piece of music.

Subsystem #7: Power Supply Unit

This unit provides power to all other units. After considering the large amount of current that the Fret Pressing Unit might draw (usual estimate around 6.6A), we decide that batteries would not be a good fit for the design due to the cost and insufficient durability. Therefore we now chose to use AC to DC Power Supply Adapter.

Requirements	Verification
The Power Supply Unit must be able to supply a voltage of **5V±0.2V** for a current load up to **10A**.	A. Link Fret Pressing Unit to PCB and use test points to confirm that **5V±0.2V** voltage and sufficient current is supplied (**1.1A±0.1A** more for every actuator activated). B. Link String Pressing Unit to PCB and use test points to confirm that **5V±0.2V** voltage is supplied and can play at least 70dB when playing. C. Link Arduino Mega to PCB and confirm that 5V output pin can output **5V±0.2V** voltage D. Link Alarm Unit to PCB and confirm that

	5V±0.2V voltage is supplied and **1mA±0.1mA** current is supplied E. Link Drum Beat Unit to PCB and confirm that **5V±0.2V** voltage is supplied and **360mA±5mA** is supplied.
The Power Supply Unit must be able to convert 120AV voltage to 5V DC.	Link to PCB and use test point to confirm the output voltage is **5V±0.2V**

Tolerance Analysis

One important concern we have is the attachment of the Fret Pressing Unit and String Strumming Unit. In order to play the music correctly, we need the units to be firmly attached to the guitar. Otherwise during playing a piece of music, the units might move on the guitar. This will negatively influence the functionality since we would no longer have guarantee to play the correct notes or hit the correct string. We have weighed some of the sample actuators, and the average weight is about 30 grams. For the Fret Pressing Unit, we need 36 of these aligned in a 6x6 matrix. Therefore, the weight of the unit is over 1 kilogram. We should attach the unit to the guitar by either tying the unit to the guitar or using some extra supporting frames. The trade-off here is that tying the unit costs less and the structure is more portable as well, while extra frames will provide a more solid support to the unit so that it is less likely to move during the play.

Another problem we need to take into consideration is the power of the actuators. The ROB-11015 actuator we plan to use has a parameter of 5V 1.1A. according to the datasheet. Therefore, according to the formula:

$$P = I \times V$$

The power consumption of one actuator isb 5.5W. Note that not all actuators work at the same time since a guitar only has 6 strings, only one actuator on each string will work at a time, meaning that at most 6 actuators are going to work at a time. Then total power consumption, according to:

$$P_{total} = \sum P$$

should roughly be $6 \times 5.5 = 33W$. This calculation does not include the power of other units, such as the control unit, alarm unit and drumbeat unit. If we want to make our design more power-friendly, we might want to switch to some actuators with less power consumption. However, according to our initial test with ROB-11015, this type of actuator is just able to press the string. We are afraid that actuators with less power consumption will not be able to even press the string.

After experimenting with the actuator, we also observe that the temperature of the servo would rise to 65 ℃ when working, according to the datasheet [5]. We will use abs material, which has a glass transition temperature of 105 ℃, for the actuator support frame.

COVID-19 Contingency Planning:

All team members can order parts online. The hardware assembling work will be conducted together in Peilin’s room, along with weekly meetings and necessary in person meetings. We have a 3D printer in Peilin’s apartment for manufacturing all mechanical parts. And Peilin also has multimeters and vernier calipers to accomplish measurement and verifications. There should be no foreseeable impediments caused by COVID-19 related inconveniences.

3. Cost and Schedule

Cost Analysis:

Item	Cost of item (US dollars)
PCA9685 x 1	6.69
MCP23017 x 3	3x3 = 9
3D printing parts	20
Arduino Mega with ATMega 2560	40
Servors x 7	3 x 7 = 21

Actuators x 36	4 x 36 = 144
Wires and resistance	5
Microphone	10
Time record chip	4
Hex LED	4

Therefore, the total cost to build our design is about 263.69 USD. The labor fee is 3 (people) x 50 (hours per capital) x 2.5 x 15 (USD per hour) = 5625 USD. So the total cost is 5888.69 USD.

Schedules:

Date	**Peilin Rao**	**Jiyu Hu**	**Qianlu Chen**
10/7/2020	Finish the 3D printing part of the fret pressing unit. Start to design the 3D printing part of the string strimming unit.	Finish the prototype circuit of the Drumbeat unit. Finishing purchase of all the requirements parts.	Finish Testing the LED and RTC module on breadboard. Achieve function of showing time, and setting alarm with button. Draw the PCB routing for LED and RTC and button.
10/14/2020	Draw the PCB part for the fret pressing unit.	Draw the PCB part for the string strumming unit.	Integrate PCB design for all units.
10/21/2020	Test the prototype of the fret pressing unit circuit.	Test the prototype of the fret pressing unit circuit.	Integrate all circuits on breadboard for testing and await for PCB arrival.
10/28/2020	Start to write code in center control part, including possible finite state machines and control logic	Take tests on all individual parts except center control, fix bugs if any.	Soldering and verification for first round PCB. Make necessary changes and improvements, submit the final version of PCB.
11/4/2020	Program music into the control control music.	Test the code and testing.	Integration of code and testing.
11/11/2020	Edge tests on basic functional units (#1 #2 #3)	Edge tests on advanced units (#4 #5 #6)	Integration of code and testing
11/18/2020	Demonstration	Demonstration	Demonstration
After that	Write final paper	Write final paper	Write final paper

4. Ethics and Safety

Our Auto-played guitar module itself is a compact and lightweight device that will attach directly to a guitar. Therefore, it is very hard for the device to cause harm to the user even under situations of misuse. The only potential safety issue is the working temperature of the actuators in the Fret Pressing Unit (65 ℃). If touched by the user by mistake, this unit might burn the fingers or other body parts that are in direct contact with the unit. We solve this issue by placing a physical cover to this unit so that the users have no chance touching the hot parts even when misuse. We do not need to worry about the cover worsening the sound of the guitar since the guitar sound is not from the parts covered.

The main concern is the misuse of the device might cause damage to the device itself as well as the guitar.

- The Fret pressing unit utilizes some small motors that should be aligned to the guitar strings. If this unit is not properly placed, the motor might hit some fragile parts of the guitar. During our design, we will make the alignment of the motors on the frame robust enough so that the motors will not shift their place on the frame, also, we will tune the strength of the motion of pressing the string in an attempt to prevent potential damage during misuse.
- The units are connected through physical wires, which might stumble the users if placed wrongly. We will create a physical frame to stabilize the wires in the structure so that the users has much less chance to touch or break the wires.

We anticipate concerns about our work going against #2.6 in ACM Code of Ethics[2] since some people think that machines should never replace human beings in terms of creating art and music. However, in our opinion, the project's purpose is never to replace the human efforts in creating and performing fine music. In contrast, it is rather a means for many more people to enjoy music and get more access to the beautiful and original sound of guitars.

5. Citations

[1] Yamaha Disklavier ENSPIRE Product Demo | Piano Gallery Utah: (Apr.9.) Retrieved on October 01, 2020 from https://www.youtube.com/watch?v=sxI0PIPZjBQ

[2] The Code affirms an obligation of computing professionals to use their skills for the benefit of society. (n.d.). Retrieved October 01, 2020, from https://www.acm.org/code-of-ethics

[3] O2 Details. (2011.08). Retrieved October 01, 2020, from http://nwavguy.blogspot.com/2011/08/o2-details.html

[4] ZHO-0420S-05A4.5 SPECIFICATION.pdf (2015, 05) Retrieved October 01, 2020: https://cdn.sparkfun.com/datasheets/Robotics/ZHO-0420S-05A4.5%20SPECIFICATION.pdf

[5] Servo motor SG90 Data Sheet(2016, 06) Retrieved October 01, 2020: http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf

[6]MCP23017/MCP23S17 Data Sheet (2016, 07) Retrieved October 01, 2020: https://ww1.microchip.com/downloads/en/DeviceDoc/20001952C.pdf

[7] PCA9685 16-channel, 12-bit PWM Fm+ I2C-bus LED controller(2015, 04) Retrieved October 01, 2020 from https://cdn-shop.adafruit.com/datasheets/PCA9685.pdf

[8]ht16K33v110 Data Sheet (2011, May) Retrieved October 01 2020, from https://cdn-shop.adafruit.com/datasheets/ht16K33v110.pdf