PROJECT REPORT

(SELF PLAYING GUITAR)

INTRODUCTION :

If you like to make music, but don't consider yourself particularly talented to play an instrument, we have come up with an innovative solution, AN AUTONOMOUS GUITAR. The structure makes use of an Arduino Uno Board, along with servo motors. This means the guitar can literally strum itself.

MOTIVATION :

:> In this technologically evolving era, where everything is becoming automated, why not musical instruments? Therefore we chose to make this as our project to be a small little part of this technological revolution.

:> Secondly, you might be wondering "how will it help people". Why does it have to? Why can't it just be made to give people the pleasure of music, make a life a little more stress-free?

:> All of us like music and we wanted to reflect that in our project too.

PROBLEM STATEMENT :

:> Our problem is centered around the making of a self playing guitar. In order to achieve the desired output, we need to correctly set up the servo motor configurations and give the right rotation to hit the string. The basis of this idea was taken to help the physically disabled ones to enjoy playing an instrument and also as a stress releasing agent for many more.

:> The servo motors were not easily getting attached to the guitar.

:> Firstly we tried making the self-playing guitar using two strings but then we faced the difficulties with the placing of servo motors on the fretboard.

***PROPOSED SOLUTION* :**

:> DATA PROCESSING AND CODE WORK :
The code reflecting the positions of servo motors, desired angle setting , Input node pattern is edited using an Arduino IDE.
Next, it is burned on the Arduino, the data or code needs to be processed and classified into particular chords with the amount of delay that needs to be given in between the chords.

:> OUTPUT :

The servo motor corresponding to the chord is set to ON mode and given the desired rotation so that the string is plucked as a result.

:> INTERFACING :

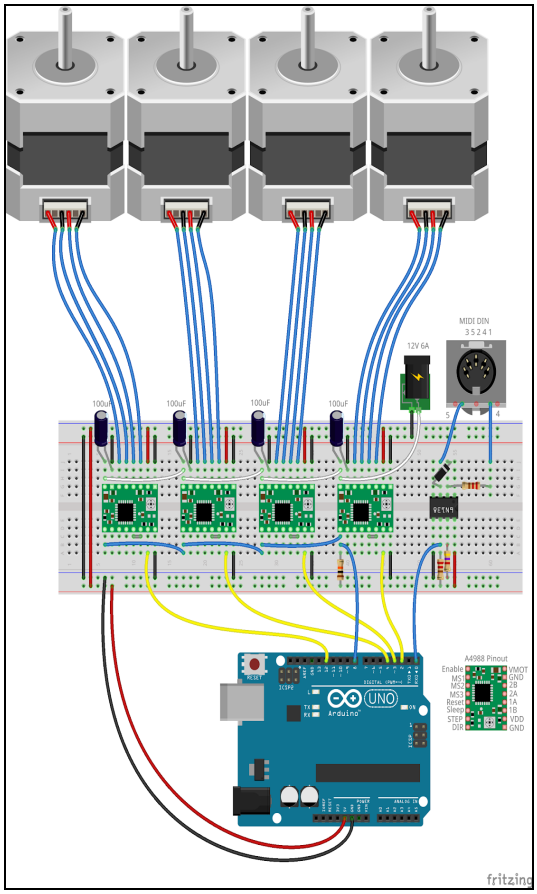
The interfacing is done by male to female connectors. The microcontroller is fed with the appropriate code. This microcontroller can be powered by an external 5V supply or can be auxiliary powered by the laptop's USB port.

:> We used a single string instead of two and made six nodes on fretboard using servo motors.

SOLUTION :

SCHEMATIC DIAGRAM :

(OUR PROJECT)



(CONNECTING MULTIPLE SERVO MOTORS TO ARDUINO USING BREADBOARD)

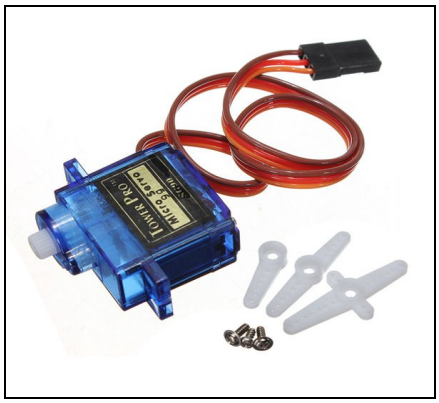
LIST OF COMPONENTS :

HARDWARE :

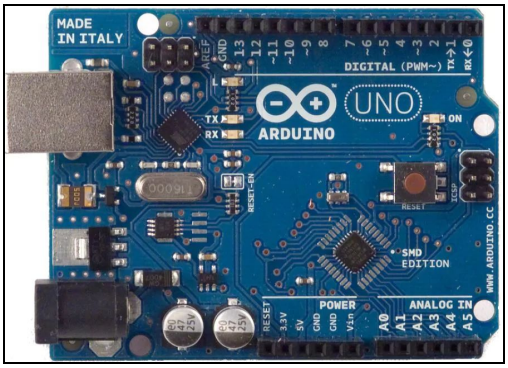
:> Guitar

:> String

:> 7 Servo Motors

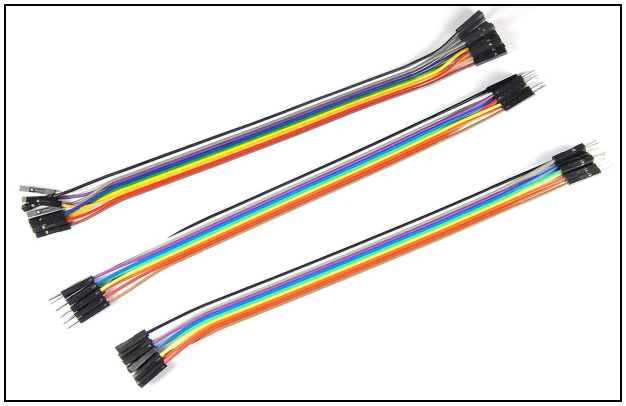


:> Arduino Uno

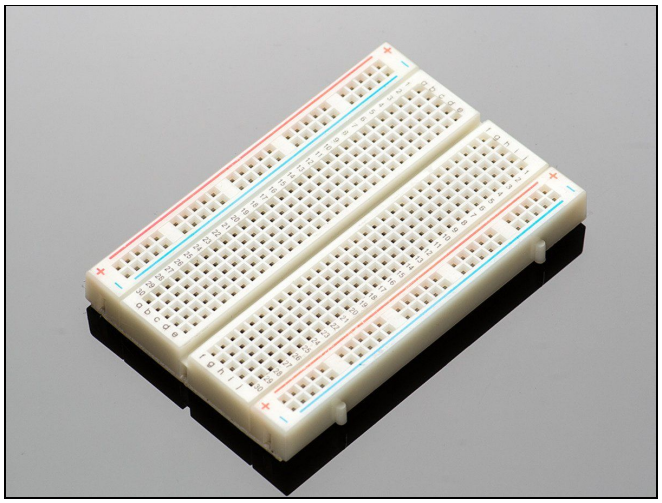


:> Some stationary (glue gun, tapes)

:> Jumper Wires



:> Breadboard(not PCB)



:> Connecting Cable

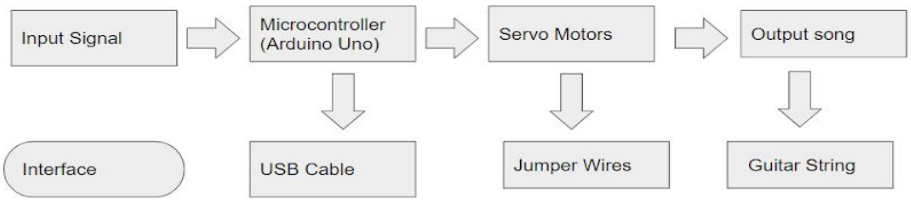
SOFTWARE :

:> Arduino IDE

METHODOLOGY :

:> Working and the detailed solution :

Block Diagram:



An Arduino is proposed as the control unit of the device. An Arduino has only 13 digital input/output pins and three were necessary for each string. Individual strings can be plucked reliably. However, as the code is currently set up, two strings cannot be plucked at the same time. This is due to the nature of the Arduino code which runs its primary function in a continuous loop.

:> Breadboard and Electrical Circuits :

With the basic circuitry kit that was used to create our device, manual wiring of the circuitry was required and this allowed for wires to come out of the breadboard and prevent the device from working normally. The Arduino and breadboard are attached using the connected wires.

:> Servo motors :

Usually, a servo motor turns 90 Degree in either direction hence maximum movement can be 180 degrees. However a normal servo motor cannot rotate any further to a built-in mechanical stop.

We take three wires as out of a servo: positive, ground and control wire. A servo motor is controlled by sending a pulse width modulated(PWM) signal through the control wire. A pulse is sent every 20 milliseconds. Width of the pulses determines the position of the shaft. When we command a servo motor to move by applying a pulse of appropriate width, the shaft moves to and holds the required position of the shaft. The rotation of servo motors is taken here to be around 70 Degree.

:> <u>Code explanation</u> :
Firstly we included the library used for servo motors.
Setup :
We attached the 7 servo motors.

Functions :
Some extra void functions are included. One function for plucking up the pick and second for plucking down the pick. Other functions are for pressing and releasing the servo motors which are present at the fretboard to play different nodes.

Loop :
Now playing that servo motor present on fret as a node which is required for the song at that moment and so on.

At a time only two motors will be working since we didn't provide the battery and only the Arduino voltage(5 volts) is provided as an input so all the motors cannot work at the same time.

<u>SYSTEM EVALUATION</u> :

The initial idea involved making use of 2 strings with 6 nodes. We tried the configuration but due to space constraints, it failed. Finally, we made use of a single string with 6 nodes corresponding to which 6 servo motors(to press the string) were used and yet another to pluck the string. A couple of songs were played. The corresponding musical notes were coded on the IDE.
A list of songs that we played are as follows:
1. Happy Birthday Song
2. Maggie advertisement tone
3. Twinkle-Twinkle song
The system worked pretty well with a melodious sound being produced, however, the sound created due to servo motor rotation lowered the melody a bit. The physical assembling of the motors on the fretboard was a bit challenging as they came out frequently due to less adhesivity.
We proposed the use of an *amplifier* to amplify the output for a clearer and neater tune.
Further, we supplied power to servo motors from the Arduino board which limited the use of servo motors to only two at a time. As an improvement, we made use of 9V external power supply to feed the servo motors.
:>*Use of Bluetooth module Arduino :*
It's an Arduino app that can be used to send serial data input to the Arduino Bluetooth module(when a button is pressed on the app). The Arduino Bluetooth module at the other end receives the data and sends it to the Arduino through the TX pin of the Bluetooth module(connected to RX pin of Arduino).

CONCLUSION :

The goal of this project was to design a self-playing guitar and allow users to play a guitar through a variety of interfaces. To accomplish this goal, a microcontroller was used and tested. The device successfully plucked the strings on an acoustic guitar and attached to the guitar as per the code fed.

Recommendations *:*

Optimation of time delays:
Testing should be done to optimize the length of the time delays in the code. If the time delays were to be optimized, the time it takes for the Arduino to read consecutive signals could be reduced. This could help to minimize the response time of the device to the user input.

Plucking Multiple Individual Strings at Once :
The code should be adjusted to allow for multiple individual strings to be plucked at once. This increases the amount of complexity of chords played while using the device.

SUBMITTED BY :

AMAN SINGH KUSHWAHA : 2018CSB1071
ANAMIKA SHARMA : 2018CSB1072
ASHNA V AJITH : 2018CSB1076
AYUSH VERMA : 2018CSB1078
DIKSHA KAPOOR : 2018CSB1084