Simple Electronic Guitar

DSP Lab Fall 2020 Final Project Report

Names	**Jihang Xie**	**Dajr Alfred**
Student IDs	N10907771	N10249121
NetIDs	jx1187	dva240

Introduction

The aim of our project is to design an electronic guitar prototype. This will be accomplished by utilizing a Raspberry Pi module and MPR121 capacitive touch sensors to simulate the fretting and strumming of a conventional guitar. Implementation of the Karplus Strong Algorithm on the Raspberry Pi will be used to simulate guitar notes in real-time.

The guitar is a fretted musical instrument that usually has six strings[1]. It is typically played with both hands by strumming or plucking the strings with either a guitar pick or the fingers/fingernails of one hand, while simultaneously fretting (pressing the strings against the frets) with the fingers of the other hand. The sound of the vibrating strings is projected either acoustically, by means of the hollow chamber of the guitar (for an acoustic guitar), or through an electrical amplifier and a speaker. Almost all guitars have frets, which are metal strips (usually nickel alloy or stainless steel) embedded along the fretboard and located at exact points that divide the scale length in accordance with a specific mathematical formula. The ratio of the spacing of two consecutive frets is $\sqrt[12]{2}$ (twelfth root of two). In practice, luthiers determine fret positions using the constant 17.817—an approximation to $1/\left(1-1/\sqrt[12]{2}\right)$. If the n_{th} fret is a distance x from the bridge, then the distance from the (n+1)th fret to the bridge is x-(x/17.817)[2] Guitar tunings assign pitches to the open strings of guitars, including acoustic guitars, electric guitars, and classical guitars. Tunings are described by the particular pitches denoted by notes in Western music. By convention, the notes are ordered from lowest-pitched string (i.e., the deepest bass note) to highest-pitched (thickest string to thinnest).[3,4] Standard tuning defines the string pitches as E, A, D, G, B, and E, from lowest (low E2) to highest (high E4),as shown in Figure 1[5]. Standard tuning is used by most guitarists, and frequently used tunings can be understood as variations on standard tuning.

	0	I	II	III	IV
	Open	**1st Fret (index)**	**2nd Fret (middle)**	**3rd Fret (ring)**	**4th Fret (little**
6th string	$\mathbf{E_2}$	F_2	$F\sharp_2/G\flat_2$	G_2	$G\sharp_2/A\flat_2$
5th string	$\mathbf{A_2}$	$A\sharp_2/B\flat_2$	B_2	C_3	$C\sharp_3/D\flat_3$
4th string	$\mathbf{D_3}$	$D\sharp_3/E\flat_3$	E_3	F_3	$F\sharp_3/G\flat_3$
3rd string	$\mathbf{G_3}$	$G\sharp_3/A\flat_3$	A_3	A♯3/B♭3	B_3
2nd string	$\mathbf{B_3}$	C_4	$C\sharp_4/D\flat_4$	D_4	$D\sharp_4/E\flat_4$
1st string	$\mathbf{E_4}$	F_4	$F\sharp_4/G\flat_4$	G_4	$G\sharp_4/A\flat_4$

Figute 1: ***Chromatic Note Progression***

MPR121 capacitive touch sensors were selected to receive the touch inputs, which simulate the fretting and strumming of a guitar. The signals from the MPR121 module are transmitted to the Raspberry Pi where they are processed to determine which notes are being played. The MPR121 is a Capacitive Touch Sensor module which has 12 input pins. This device utilizes the I2C interface to communicate with the Raspberry Pi microprocessor via it's SCL (Serial Clock) and SDA(Serial Data) lines. In this project, we aim to use 4 MPR121 modules to simulate 48 notes of an acoustic guitar. Each input of the MPR121 is connected to a ⅛" wide Pyralux pad which is a type of conductive copper foil. When this conductive foil is touched, an input is triggered and the note specific to that pad is played..

The Raspberry Pi Model 3B was released in February 2016 with a 1.2 GHz 64-bit quad core processor, on-board 802.11n Wi-Fi, Bluetooth and USB boot capabilities.[6] On Pi Day 2018, the Raspberry Pi Model 3 B+ was launched with a faster 1.4 GHz processor and a three-times faster gigabit Ethernet (throughput limited to ca. 300 Mbit/s by the internal USB 2.0 connection) and 2.4 / 5 GHz dual-band 802.11ac Wi-Fi (100 Mbit/s). The Raspberry Pi is able to run a custom version of Linux known as Raspian. This allowed for the use of the Python programming language to program the Raspberry Pi Model 3B+ for real time analysis of inputs from MPR121 and the output of corresponding notes in realtime.

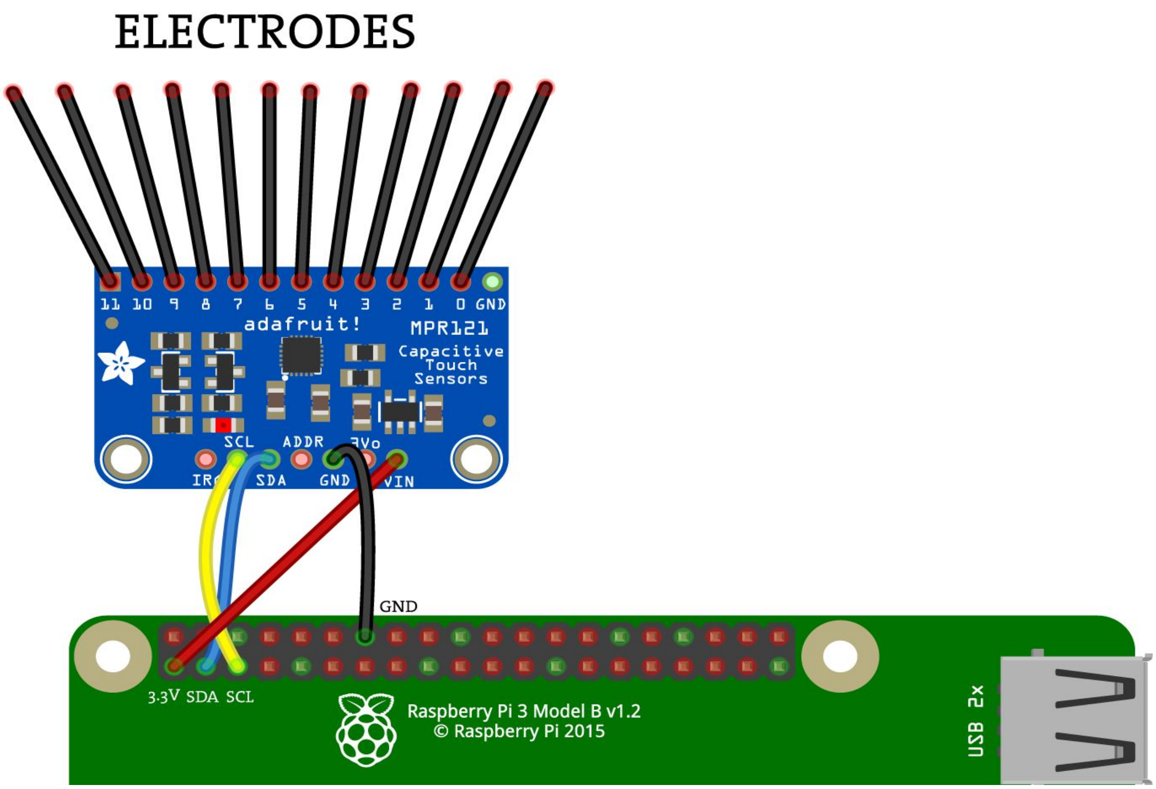

Figure 2: ***Wiring Diagram of Raspberry Pi and MPR121 Module***

In order to facilitate this real-time application, the Karplus-Strong (KS) algorithm was used to simulate an acoustic guitar. In 1983, the Karplus-Strong (KS) plucked-string algorithm was published. This paper presented a relatively simple, efficient model which is based on the earlier work by McIntyre and Woodhouse (1960). In the Karplus-Strong paper, the plucked string was modelled as having a discrete output $y(n)$ produced by a discrete input $x(n)$ plus the two-point average of previous versions of $y(n)$, delayed by N samples. Thus $y(n) = x(n) + ½\, y(n-N) + ½\, y(n-N-1)$. This could be modelled as two filters in series with one providing an N sample delay and the other being a two-point averaging filter. These two filters have a total phase delay of $(N + ½)$ samples which corresponds to a fundamental or pitch frequency, $f0 = Fs/ (N + ½)$ in Hertz.

Physical Model

The physical model of the guitar was modelled in SolidWorks. It was designed to be easily assembled after laser-cutting using ⅛" thick acrylic sheets. The individual parts are held together using M3 screws and stands as well as superglue. The inputs are mounted on individual separable pieces for this initial prototype but will be replaced with a single PCB in future revisions. The guitar assembly features mounting points for the MPR121 modules, the Raspberry Pi as well as a 7" touchscreen display unit.

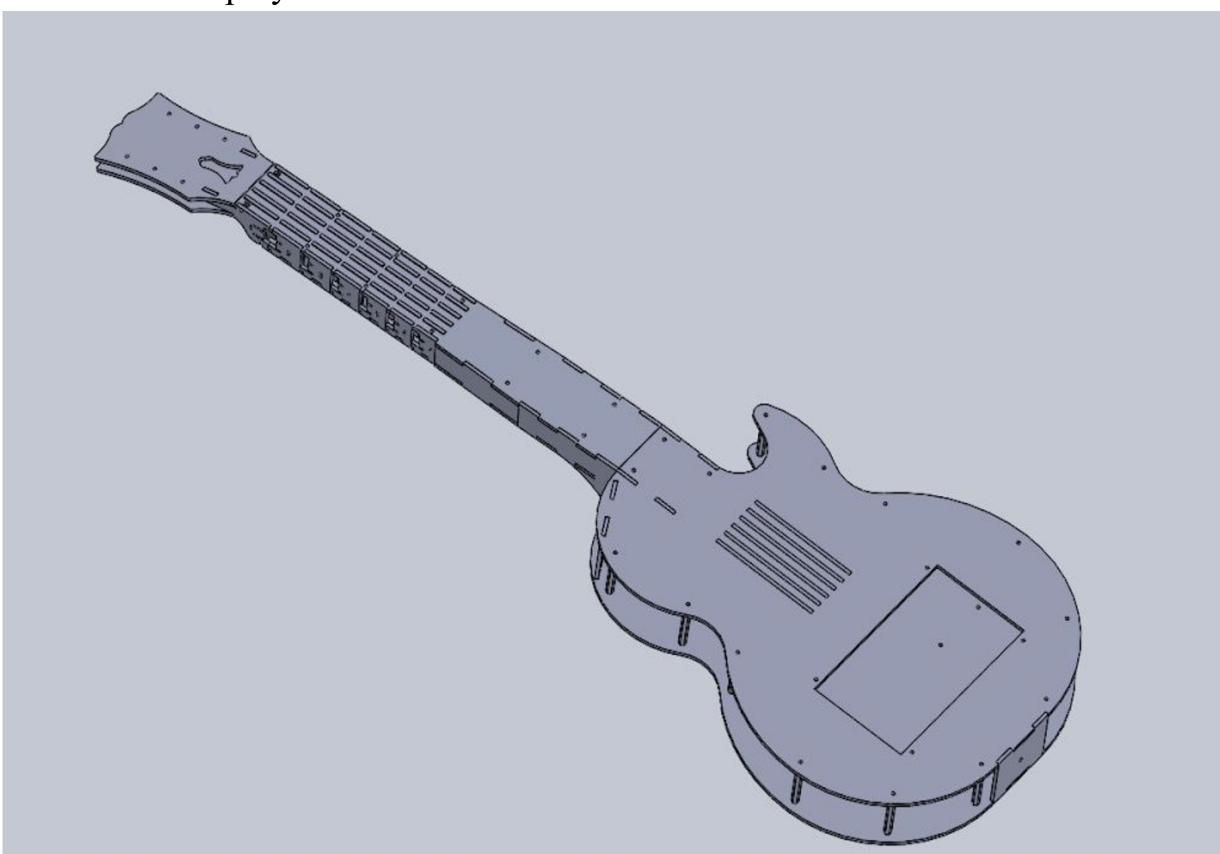

Figure 3: ***Assembled Guitar Model***

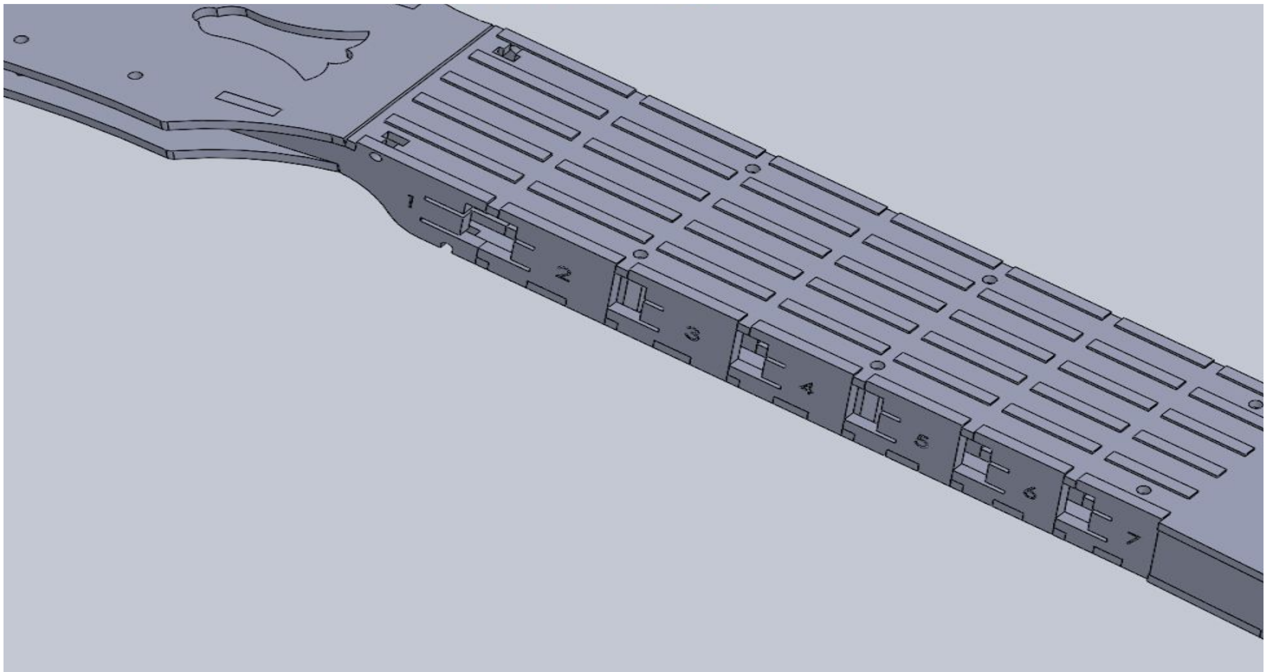

Figure 4: ***Individual Input Pieces (Labelled 1-7)***

Figure 5: ***Acrylic Sheets - Material Used to Build Guitar***

Figure 6: ***Laser-cut & Partially-assembled Physical Model***

Figures 3 to 6 show both virtual and physical components of the project. The physical model was able to be partially assembled by the project deadline. Sadly, due to the fact that the team member with hardware expertise was outside of the US, the model was not able to be completely assembled. Therefore, completion of the guitar assembly will be addressed during future work on this project. The software required to play the guitar, however, was successfully completed and is available on the project GitHub.

Project GitHub

All the code written for this project can be found on the project GitHub which can be accessed via the following link and QR-Code.

GitHub Link: https://github.com/dajralfred/Guitar-Simulation

Figure 7: ***QR-Code with Project GitHub Embedded***

Reference

[1] https://en.wikipedia.org/wiki/Guitar#Frets
[2] *Mottola, R.M.* "Lutherie Info—Calculating Fret Positions"
[3] https://en.wikipedia.org/wiki/Guitar_tunings
[4] Brennan, Maureen (2008). "Linen lites - drThe drones of swedish folk music". Dirty Linen: 15.
[5]https://en.wikipedia.org/wiki/Guitar_tunings#/media/File:Standard_diagonal_shifting_of_C_major_chord.png
[6] https://en.wikipedia.org/wiki/Raspberry_Pi#Generations
[7] Karplus, Kevin, and Alex Strong. “Digital Synthesis of Plucked-String and Drum Timbres.” Computer Music Journal, vol. 7, no. 2, 1983, p. 43., doi:10.2307/3680062.