UKULELE INSTRUMENT TUTOR

By

Matt DiLiberto

Ammar Faiz

Udit Sharma

Final Report for ECE 445, Senior Design, Fall 2012

TA: Lydia Majure

12 December 2012

Project No. 08

Abstract

The Ukulele Instrument tutor is design to be a self-teaching device to learn how to play the Ukulele. It provides direct assistance to the user to learn how to play music without the use of score sheets. An Arduino Mega 2650 drives the logic that controls an LED array. This array displays various songs, notes, and chords that can be selected by the user. This project utilizes the force-sensitive resistor do detect whether or not the user has played the correct note or chord by sensing a change in applied pressure. The device is portable and is powered by a 9V battery, which allows users to practice this instrument at their convenience.

Contents

1. Introduction

1.1 Purpose/ usefulness of project

There are not many musical instruments in the market today that provide direct assistance to a musician in playing a new instrument. Our idea is to utilize our experience in the field of electrical engineering to build a ukulele with integrated LEDs in it which will help the musician figure out which fret to press and then strum it for the required sound. We will program various notes, chords, and songs into the microprocessor for the musician to play as well as utilizing force resistance sensors to ensure the proper notes are played. An LCD will act as a user interface and allow the user to determine what is going to be played. The project will be unique if it has inputs like note detection or finger position sensing, LED outputs, as well as stand-alone operation.

1.2 Project functions

The goal of this project is to develop a tutor that will allow the user to play a ukulele proficiently. This will help them develop the skills needed to play without the aid of a tutor in the future. The project will be accomplished by using LEDs to direct the user on where his or her fingers need to be placed. After that the sensors will determine if the note played matches the correct note in the song upon which it will display the next note to play. This will be an easier alternative to learning to play an instrument in a class where attention to individuals might not be emphasized.

<u>Benefits:</u>

- Helps the user learn how to play the correct note quicker than conventional methods.
- It is a portable system and can be taken wherever desired.
- LEDs display system undermines the need for a musical score sheet.
- Cheaper than other products available on the market, such as Fretlight.
- Its fun!

<u>Features:</u>

- LEDs to direct proper finger placement.
- Import songs to device which can be viewed on the LCD screen.
- Displays the note being played on the LCD screen.
- Portable and easy to use.

1.3 Blocks/ Subprojects

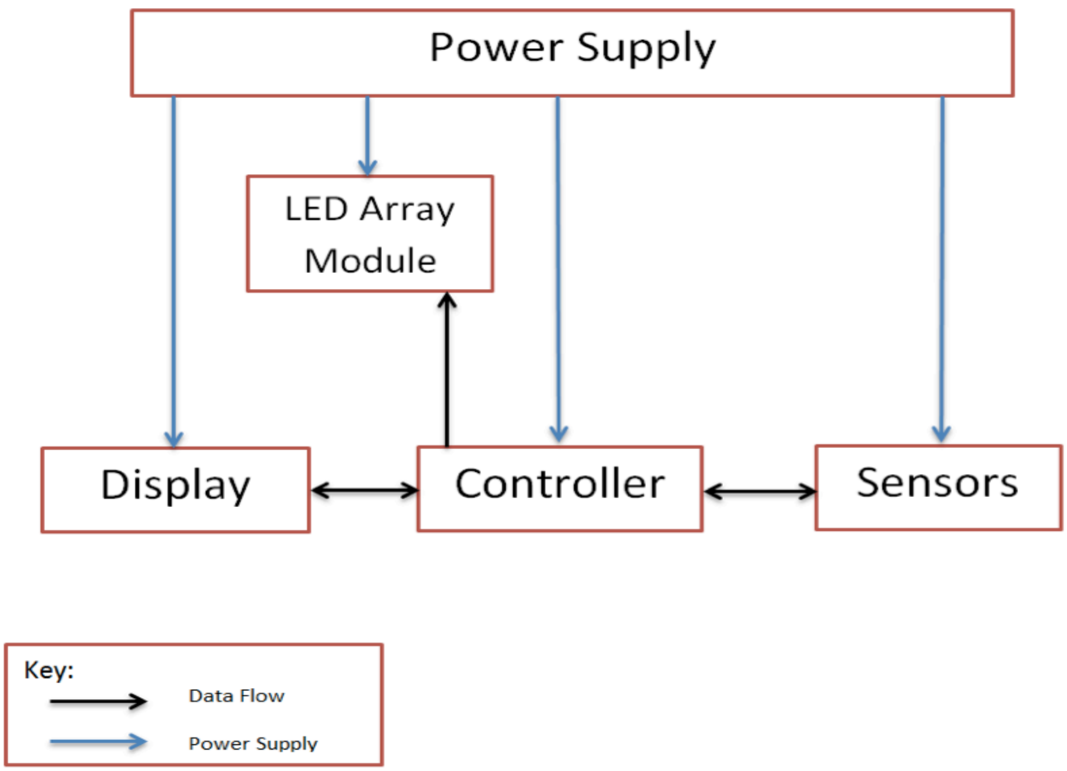

Figure 1 Block diagram of the project. This shows our independent modules interacting with one another

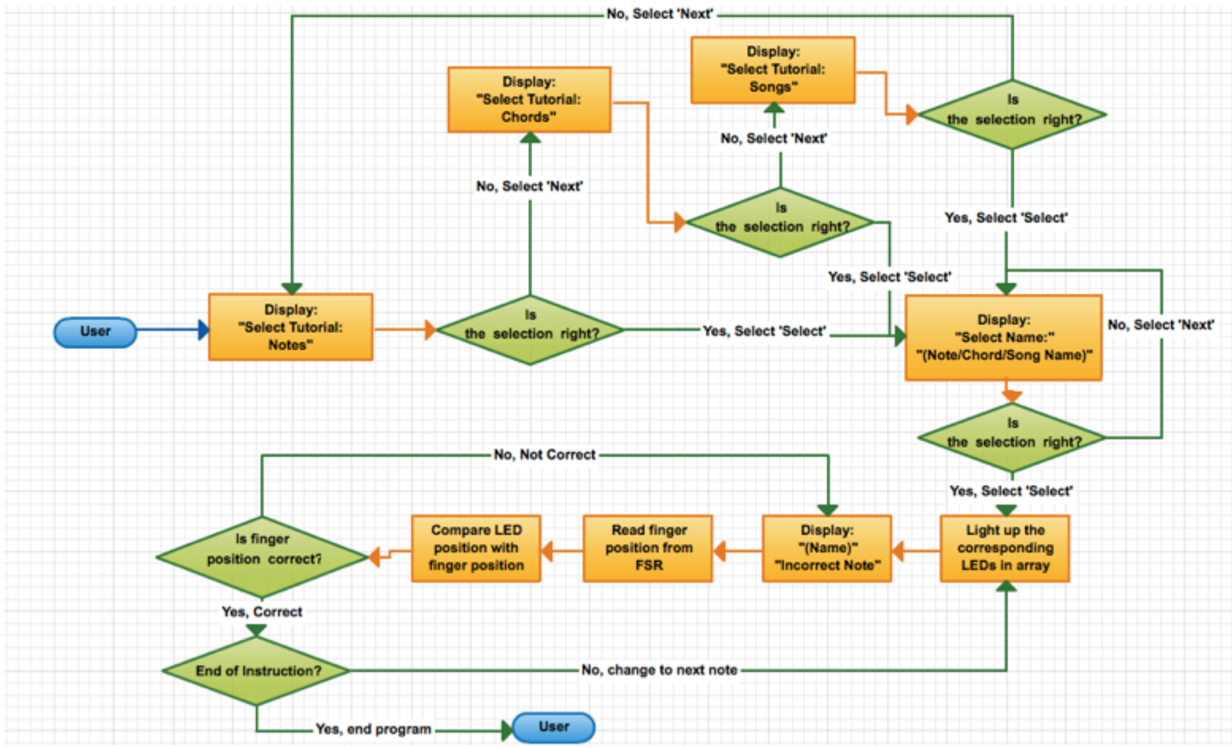

Figure 2 Flow logic for user interface

2. Design

2.1 General design alternatives

The main alternative that had to be made in our design was to do with our LEDs. The LEDs that we intended on using for our project did not turn to be a good fit. There were three main issues with those LEDs, firstly, these LEDs were just about the size of the fret wire, which made getting the string and the fret wire to touch while pressing a note really cumbersome. Secondly, these LEDs were to be used with a shift register which turned out to be causing delays in our system. This was not acceptable as we needed the change in LEDs to be instant. Third issue we had was that these LEDs did not allow us to calibrate the FSRs consistently. Loads of calibration issues were encountered because of the previously chosen LEDs. Hence to solve all the issues, we used surface mount, current dependent LEDs. The new LEDs were small enough to fit below the fret wire and also small enough to easily allows us to calibrate the FSRs. Also the LEDs were individually attached to the arduino, which meant each LED was controlled independently. This allowed changes to be instantaneous and perfect for the scope of our project.

2.2 Equations/ Simulations/ General Circuits

2.2.1 Power Requirements:

Part	Voltage(V)	Min Current (mA)	Max Current (mA)	Min Power (mW)	Max Power (mW)	%age of Usage	Average Current (mA)	Average Power (mW)
LED	5	0	12	0	60	20%	2.4	12
Arduino Mega 2560	8.4	1	200	8.4	1680	100%	200	1680
LCD	3.3	17.5	18.5	57.75	61.05	100%	18.5	61.05
FSR	9	0	0.02	0	0.18	100%	.010	0.09
Total		18.5	230.52	65.97	1801.2		220.91	1753.14

*Parts not displayed in this list have negligible effects on the power system, and therefore are left out.

Using the information from the power table, voltage required by the system is

$$V=P_{avg}I_{avg}=1753.14mW220.91mA=7.93599V$$

Hence for all the components to function, we need at least 8V of input voltage.

2.2.2 LED Simulations and Calculations

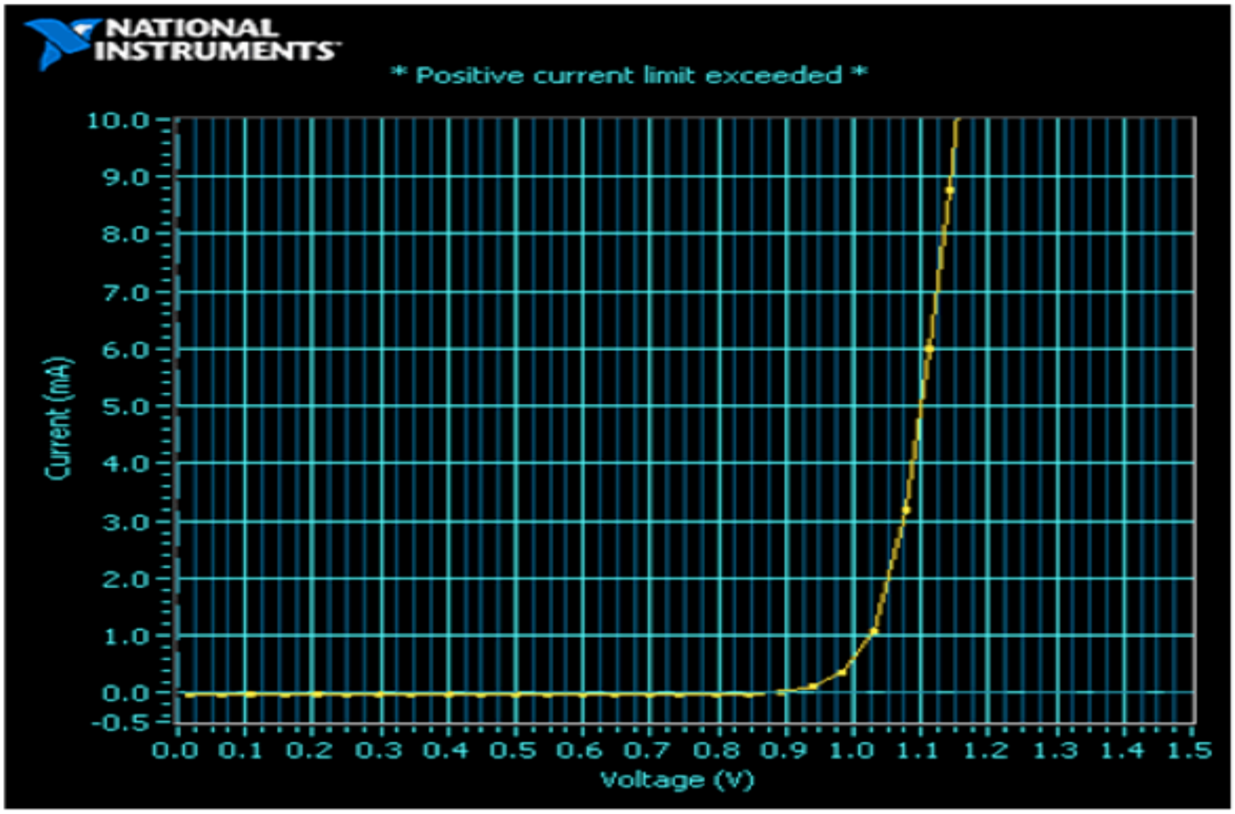

Figure 3 IV Relationship for an LED

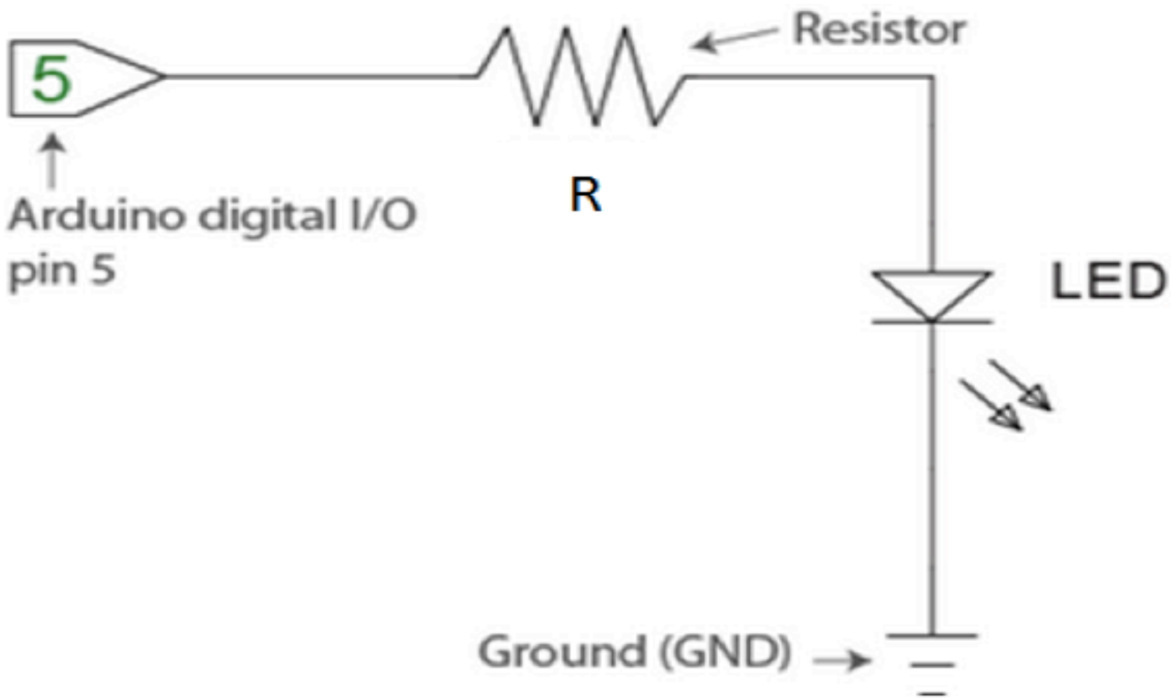

Figure 4 Sample circuit for LED

To calculate the value of the resistor R, we used

$R \;\; = 5V - (V_{LED})(30mA)$

$R_{min} = 5V - (2.5V)(30mA) = 83.3\Omega$

$R_{max} = 5V - (1.7V)(30mA) = 110\Omega$

It is not recommended to use the absolute minimum voltage, hence the value chosen for resistance in 115Ω, which gives us a 24.8mA operational current, which will be fine. The color chosen for the LED is red, because it draws the most attention from human beings and also instigates them to work faster.

2.3 Detailed Description of Design

2.3.1 Microcontroller

The microcontroller used for the implementation of the ukulele tutor will be the Arduino Mega 2560. Specifically, this uses the ATMega2560 chip. A music file that has been converted manually into notes and chords will be stored in the memory of the microcontroller and will be programmed so that it is passed to the LED array module to light up the note that is supposed to be played. The pressure sensors also send data back to the microcontroller and from those information, we can determine whether the note that was pressed was the correct note or not. Another feature controlled by the microcontroller is the LCD that displays the song selection screen at the beginning and once the song is selected, it displays the note that is being played.

The operating voltage for this microcontroller is 5 Volts. In addition to that, the microcontroller can take voltage inputs between 7 to 12 Volts, which is acceptable for our design. For the inputs and outputs, the Arduino Mega 2560 consists of 54 digital I/O pins as well as 16 analog input pins which also matches with our requirements for a microcontroller.

For the inputs, 16 analog input pins will be connected to the force sensitive resistors (FSRs) on the fret board. Furthermore, the LEDs, LCD screen and input buttons will be connected to the digital I/O pins. The processing of user inputs will be based on the song to be played and upon execution of that command the tutor will commence its program. The first part of the process will involve taking input data from the FSRs. Following that, the microcontroller will show the current note to be played on the LCD and light up the corresponding LEDs on the fret board. The last step would involve determining whether the user correctly places his fingers to play the particular note, which would be based on comparing data from the sensors to the data stored in the microcontrollers program. If the user improperly plays a note, the LEDs will not progress on to the next note of the song until the correct note is played. This will ensure proper mastery of the song as well as the user being able to take his time in learning the proper positioning for a note. For memory, Arduino Mega has 256KB flash memory, 8 KB of SRAM and 4 KB of EEPROM. This memory will be implemented to hold the program as well as songs that will be played by a user. Additionally, the memory will store sensor and LED locations for a note in order to ensure that the proper note is played.

This microcontroller is ideal for our implementation because of its memory that can store our program as well as the songs and sensor locations as well as the large amount of analog and digital I/O pins that are required for the implementation of our ukulele tutor. The Arduino Mega Kit also includes features which make it easily compatible with a windows based PC, which will make programming more user friendly and efficient.

2.3.2 LCD

The LCD we will be using is SainSmart 1602 LCD Keypad Shield. The screen has two functions, one, at the start, it displays the main menu where the user can either select a chord, a note, or a song to play. Once the user makes an option, say for example the user selects the option playing a song, then a list of songs would be displayed for the user to select. Finally when the user selects a song the he/she wants to play, the first note that needs to be played is displayed on the screen and the LEDs for that note lights up simultaneously. LCD has 4 data pins and one power pin, which would be attached to the digital output of the Arduino kit. The display can be used in 5V/3.3V IO Arduino system directly, so no external components are required. What needs to be displayed on the screen will be programmed directly onto the microcontroller. The reason this LCD was chosen was because it is compatible with an arduino so it can utilize a library to allow us to easily display text and is relatively cheap for a colored display.

2.3.3 User Interface

The user interface combines the functionality of the microcontroller, LCD, and the keypad. The six buttons available to the user are up, down, left, right, select, and reset. The left and right buttons are used to scroll through the menu. This assigns vales to the note and chord select variables so the logic that controls the LEDs and FSRs can be driven. The select button selects an option and advances to the next menu level. The up button will return to the main option screen. The interface allows the user to select the initial function, be it playing a note, playing a chord, or directly diving in to a song. If the user decided to play a note, they are taken to the next level of the user interface where they select which exact note they want to learn. There are 14 notes, 14 chords and in our case, one song to select from.

2.3.4 LED Array Module

16 red LEDs will be placed on the fretboard, that cover the first 4 frets on the Ukulele. This is because almost all the songs can be played with the notes represented by the first 4 frets. Under each of these LEDs, pressure sensors would be placed that measure the amount of pressure applied. The LEDs light up to show the user what note he needs to play and it stays until the user plays the correct note. Once the user plays the correct note, the next note to be played immediately lights up. 16 LEDs would be attached to the digital output of the arduino programming kit. Each of these connections would be made in parallel to each other, so that each of the output is controlled individually. With the program that would be written in the microcontroller, we can then individually control which LED to light up based on the note that needs to be played. Using the data sheet for a surface mount LED, provided in the appendix, the forward voltage is 2V, for continuous operations, we need 20mA of current to flow through each LED. Turns out that there is about 25mA of current that comes out of each digital pin of an arduino, hence a resistor was not needed between the LEDs and the arduino.
LED IV curve:

2.3.5 Power Supply

A 9V battery would be the source of power for the entire major component. The negative terminal of our power supply will be grounded while the positive terminal will be connected to the Vin port of our microcontroller. Also, the microcontroller contains a voltage regulator, which outputs 3.3 Volts. This will be used to power our LEDs and FSRs. The power supply would be attached in parallel with all the other units and provide the desired voltage to each component. For example, the microcontroller requires a power input of 5V for it to function; hence the battery would be attached in parallel to the controller with a resistor to regulate the incoming voltage. It will come with a battery clip so that we can attach it to our circuitry. For detailed analysis of power consumption of the system, refer to the table below.

2.3.6 Force Sensitive Resistor

The sensing unit consists of the Model 400 Force Sensitive Resistor(FSR) that would be placed behind the LEDs on the fretboard. The FSR is 0.2” [5mm] in diameter and will register a pressure value when the user presses the desired note. We will be implementing the FSR as an on/off switch and use the breakpoint located at approximately 30,000 ohms to determine whether or not the user is applying pressure. This data will be sent to the analog input of the microcontroller, which then determines whether the correct note was played or not based on the amount of pressure that was exerted. Various input pressure levels would be read and the software will determine whether or not enough pressure is applied based on reasonable pressure levels that are needed to play an acoustic instrument. The resistor will act as a voltage divider to limit the maximum input voltage that will go into the microcontroller.

2.4 Schematics with components

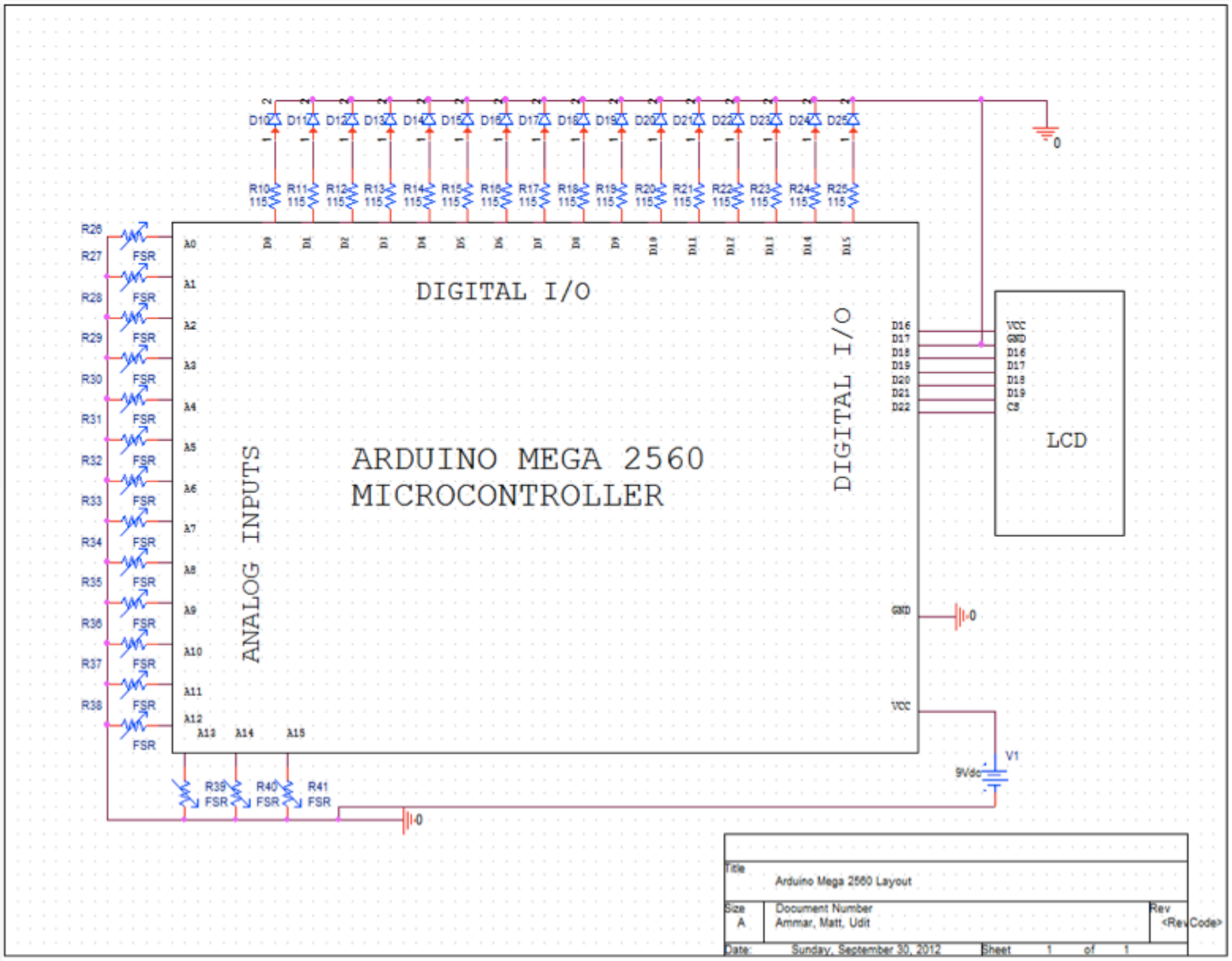

Figure 5 Shown in this figure is the circuit layout for the design of the system. This shows the connections between all of the devices and how all of the individual components are connected to the microcontroller. There are 5 main components to our circuit: Arduino Mega, FSRs, LEDs, LCD, and power supply.

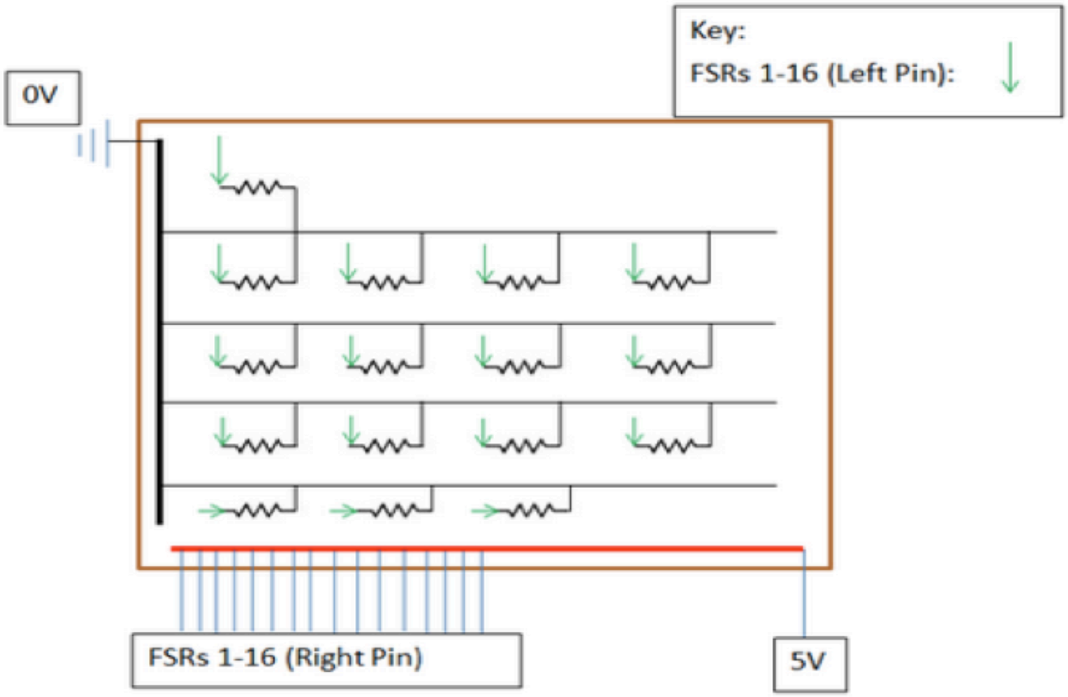

Figure 6 Shown above is a simple layout that shows how we soldered onto the vector board. The red line in the diagram represents the 5V power line and the black line represents the ground. Both the right pin and the left pin on the FSRs were extended using a thin gauge wire. Right pins of all the 16 FSRs were attached to the power line as shown in the schematic and the left pins of each FSR was soldered to one end of a resistor, which was grounded at the other end. Another wire was soldered onto the resistor; FSR connection and this wire went directly into the analog pins of the arduino mega. With the inclusion of an appropriate resistance, the fluctuations in the FSRs reading are steadier when a pressure is applied onto them. Through many calibration tests, it was noted that the best resistance to use for our projects was 5K Ohms.

3. Design Verification

3.1 Testing procedure

Functional tests for all of our components are shown in the Appendix. These tests were performed on all of our parts. On top of these initial tests that were in the design review, other tests that were deemed relevant during the course of building our project are included. These mainly consisted of testing the vector board and the functionality of the user interface. The vector board had to be repeatedly tested if there was a malfunction to ensure that the solder joints were mechanically and electrically sound and the appropriate connections were being made on the board. The continuity function of a multimeter was utilized for this purpose.

For the user interface, the tests involved making sure all the buttons functioned perfectly, the user was able to navigate easily through the menu and that the appropriate LEDs turned on when the user made his/her selection. Further tests involved making sure the FSRs work properly at the programmed pressure readings. To do the button testing, we tested the peripherals like the FSRS on the serial monitor and once we perfected the code, tested on them in accordance with the display to make sure the menu looks good and the system functions properly.

Additional testing we did was on the FSRs after soldering connections to their pins as this was a delicate process prone to errors and needed to be done with the utmost care. The testing involved hooking up the FSRs to the multimeter and making sure that the resistance reacted as designed when pressure was applied.

3.2 Discussion of results and failed verifications

3.3.1 LCD Results

Test for the LCD indicates proper operation with 5 volts at the power pin for the LCD logic. There was no flickering on the screen when characters were displayed. This test was cleared easily and problems were encountered. Buttons on the LCD shield were also tested and again they worked perfectly with buttons responding in a manner they should. The only issues that arose were related to the coding for the LCD. At times there would be garbage characters displayed on the LCD when we would expect some phrase. These issues were sorted out through trial and error method and it turned out that we needed to set the cursor on the display to make sure that the phrase fits completely in the screen if not garbage characters are displayed towards the tail end of a phrase.

3.3.2 LED Results

Testing of the LEDs indicated that they were functioning correctly. A simple test program was run to light up rows of LEDs at the same time and all the LEDs lit up perfectly. The only issue that arose with the LEDs was that they heat up if left on for a substantial amount of time, roughly more than 20mins. However this is not a major issue for our project since any LED would realistically be ON for at most 2-3mins at a time when the program is running. Therefore no attempts were made to fix this issue.

3.3.3 FSR Results

The FSRs needed to be tested individually. When first constructing our circuit on a breadboard, every FSR acted exactly as expected and there were almost no variations between any of them, which made it easier to calibrate. However, once they were mounted on top of the fret board of the ukulele using a particular adhesive, we realized that this had begun to damage them and separate the layers of the FSRs, which prevented them from having the characteristics, needed to detect an applied pressure.

Some of the FSRs were still salvageable and were still used in the final prototype. However, their properties were now different than the FSRs that weren't exposed to the adhesive and we had to change the calibration of the FSRs. This didn't pose to be too much of a problem because our pressures were used to act as an on/off switch rather than used to get an accurate reading of pressure. The FSR component passed all of the tests and verifications listed in Appendix A.

3.3.4 User Interface Results

We went over the menu and the entire user interface was tested to ensure proper functionality. It involved cycling through all of the options to make sure there were no inherent glitches in the code or program. Proper operations of all the options were checked thoroughly. We scrolled through the notes, chords and the songs on the LCD and compared the LEDs and FSR operation relative to what our code was designed to do. Furthermore, the pressures of the FSRs were once again checked on the serial monitor. Everything matched on the serial monitor with the code we had designed.

Other tests we performed for the user interface involved making sure that everything reset at the proper time i.e. if a user successfully plays a note then the LEDs will turn off and the corresponding FSRs will stop taking data. This ensured that there were no background components running which would lead to power loss as well as overheating of components.

After going through the code several times and rectifying any flaws that we observed, both logical and FSR pressure related, we were able to go through all of the features and ensure that the overall system was working properly.

3.3.5 Microcontroller Results

The verification of the microcontroller required a lot of time since there were so many things to test. After passing the peripheral tests of the LCD, LEDs and the FSRs (explained above) we had to start testing the interfacing between the microcontroller and the aforementioned components. Some of the ways we tested our system was through the sketches we had developed in order to ensure that the analog values from the FSRs were interpreted by the microcontroller correctly as well as proper output was performed to the LEDs and the LCD.

Once we ascertained that the data from the sensors was correct and there were no communication errors between the other devices, we started running the code on the complete mounted system on the ukulele. Our approach to testing the system was to cycle through multiple notes and chords, perform them properly and then do some notes properly while pressing other notes at incorrect LEDs(which were unlit) to make sure that there were no rogue FSRs taking values at times that they were not supposed to register any pressure value into the program. We documented the tests in videos, which we used in our presentation to provide a better idea of some of the testing procedures.

After going through all of the testing, we confirmed that the sensors were able to properly communicate with the microcontroller, as well as the LEDs and the LCD functioning as expected. The program could instantly distinguish when the proper note was pressed and incorrect pressing on the fret of a note would lead to no operation whatsoever until the user did it correctly. This was exactly how the system was designed to perform so we were very pleased with the results.

4. Costs

4.1 Parts

Part	Part Number	Unit Cost	Quantity	Total
LED	SLT-776TUR	$0.75	35	$26.25
Force-Sensitive Resistor	Pololu #1695	$5.26	16	$84.16
Arduino Mega 2560	A000067	$38.95	1	$38.95
LCD	SainSmart 1602 LCD Keypad	$9.99	1	$9.99
Battery- 9V	PC16049V	$1.46	1	$1.46
9VOLT BATTERY CLIP	233	$0.21	1	$0.21
5kΩ Resistors	002-100	-	16	-
TOTAL				$147.33

4.2 Labor

Name	Hourly Rate	Total Hours Invested	Total= Hourly Rate * 2.5 * Total Hours Invested
Ammar Faiz	$35	150	$13,125
Udit Sharma	$35	150	$13,125
Matt DiLiberto	$35	150	$13,125
TOTAL	$105	450	$39,375

Total Cost of Project: $39,702.39

5. Conclusion

5.1 Accomplishments

We certainly encountered our fair share of obstacles during the course of our project. Some of the obstacles were ensuring proper interfacing of the FSRs with the arduino when set up on the ukulele. Obtaining clear-cut values of pressure was harder than we initially thought and required repetitive testing on the FSRs to make sure that they were giving proper pressure values. In the end, we overcame this challenge and the FSRs were spotless in their performance.

Other accomplishments we made were making sure that the proper notes were detected while improper notes were not. This was a vital feature in our project therefore we put special emphasis on making sure that everything performed accordingly. A third milestone we made in our project was coding the song to the memory of the arduino and running it on the system. This involved playing notes and chords as well as displaying the names of the aforementioned things on the LCD, lighting up the LEDs and reading the pressure from the FSRs. This was the grand finale of the experiment and when it functioned properly, we reached a suitable level in our project for demonstration purposes as well as wrapping up the design.

5.2 Uncertainties

Uncertainties in our project were largely mechanical, such as making sure there were no broken connections or wiring in our system. Other than that we had all of the functional aspects covered and did not notice anything out of the ordinary in all of our functional tests. The system could take a fair amount of beating and was able to perform reliably once everything was in order.

Other uncertainties we faced were not being able to tell if the FSRs were burning out when we initially glued them onto the surface of the ukulele, which we had to overcome by taping them.

5.3 Ethical considerations

By abiding to the IEEE Code of Ethics, we agree:

1. To accept responsibility in making decisions consistent with the safety, health and welfare of the public, and to disclose promptly factors that might endanger the public or the environment.

    Since the user will be touching sensors, and using an electric system attached to the ukulele we must make sure it is safe and there is no chance of electric shock from our components in the system.

2. To avoid real or perceived conflicts of interest whenever possible, and to disclose them to affected parties when they do exist.

    Throughout this course, our team will ensure that we manage our time wisely, coordinate work among the team members and make sure we utilize all available resources. Therefore we will try our best to balance our work and achieve the goals.

3. To be honest and realistic in stating claims or estimates based on available data.

Our system will be advertised accurately in its reliability and how long it lasts (battery).

6. To maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations.

Since we are not very experienced in this field of building a product from scratch we will try our best to learn the necessary skills to complete the project.

7. To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others.

Throughout the course we will critique each other's work to make sure that we create the best product possible.

8. To treat fairly all persons regardless of such factors as race, religion, gender, disability, age, or national origin.

We will treat all our group members fairly and with respect.

10. To assist colleagues and co-workers in their professional development and to support them in following this code of ethics.

We will work together throughout to course to ensure proper understanding and following the codes.

5.4 Future work

Due to some of the fundamental features of our project, we will be able to expand on or existing product or adapt it to new ideas. Initially, we can move to expand this project to a large-scale design. This could mean just adapting the technology to be put on top of a guitars fret board. Next, we would want to add more FSRs and LEDs to include the remaining frets on the instrument. This allows for additional notes and chords to be played as well as more advanced songs. Our design only included a single song on the device due to time restrictions, but since we had a lot of remaining memory, we could have expanded it to include a larger song library.

From our additional testing, we have the capabilities to allow the user to record their own songs directly on device. Once recorded, the will be able to practice their songs in the same fashion they did in our original design. By altering the algorithm for the sequence or chords in our song, we can design various games that could be played that can appeal to a younger audience.

The capabilities of musical devices are extensive. All of the possible ides for future work are practical, but would not likely all be included on a single device. Each idea can be produced and sold as individual products and the marketed towards the most effective markets.

References

[1] Lee, Chang, Anaid Ortegoza, Shagun Singh, and Mustafa Bagdatli. "Sensor Workshop at ITP." :: Reports / Force Sensor Resistor. Sparkfun, 22 Nov. 2010. Web. 01 Oct. 2012. <http://itp.nyu.edu/physcomp/sensors/Reports/ForceSensorResistor>.

[2] FSR Integration Guide. InterLinkElectronics. Digikey. 10 Feb. 2012. Web. 25 Sept. 2012. <http://www.digikey.com/Web%20Export/Supplier%20Content/InterlinkElectronics_1027/PDF/Interlink_Electronics_Integration_Guide.pdf?redirected=1>.

[3] "APT3216EC." Mouser Electronics. N.p., n.d. Web. 11 Dec. 2012.

[4] http://www.mouser.com/ds/1/216/APT3216EC-20327.pdf

[5] "SainSmart 1602 LCD Keypad Shield for Arduino Duemilanove UNO MEGA2560 MEGA1280." SainSmart. N.p., 2010. Web. 30 Sept. 2012. <http://www.sainsmart.com/sainsmart-1602-lcd-keypad-shield-for-arduino-duemilanove-uno-mega2560-mega1280.html>.

[6] "Arduino - ArduinoBoardMega2560." Arduino - ArduinoBoardMega2560. N.p., n.d. Web. 01 Oct. 2012. <http://arduino.cc/en/Main/ArduinoBoardMega2560>.

[7] "The Color Red Is The Best Choice When You Want Attention." The Color Red Is The Best Choice When You Want Attention. N.p., n.d. Web. 01 Oct. 2012. <http://www.colorcombos.com/color-red-article.html>.

[8] "5000 Ohm 1/2W Flameproof Resistor." Parts-express.com. Parts Express, 2012. Web. 01 Oct. 2012. <http://www.parts-express.com/pe/showdetl.cfm?partnumber=002-100>.

[9] "Arduino Tutorial." Ladyada.net. N.p., 27 Apr. 2012. Web. 30 Sept. 2012. <http://www.ladyada.net/learn/arduino/lesson3.html>.

[10] "SainSmart 1602 LCD Keypad Shield for Arduino Duemilanove UNO MEGA2560 MEGA1280." SainSmart. N.p., 2010. Web. 30 Sept. 2012. <http://www.sainsmart.com/sainsmart-1602-lcd-keypad-shield-for-arduino-duemilanove-uno-mega2560-mega1280.html>.

[11] "Force Sensitive Resistor (FSR)." Adafruit.com. Adafruit Industries, n.d. Web. 04 Oct. 2012. <http://learn.adafruit.com/force-sensitive-resistor-fsr/using-an-fsr>.

[12] "Joe's Gizmos." Joe's Gizmos. N.p., n.d. Web. 05 Oct. 2012. <http://joesgizmos.wordpress.com/2012/06/02/project-1-force-sensitive-resistor-led-lcd-display-and-arduino/>.

[13] N.p., n.d. Web. <http://www.ni.com/cms/images/devzone/tut/image3539283546075585677.png>.

[14] Kamakawiwo, Israel. "Somewhere Over The Rainbow Ukulele Chords, Israel Kamakawiwo." Ukulele Chords, Free Chordsheets. N.p., n.d. Web. 2 Dec. 2012.

Appendix A

Requirement and Verification Table

Requirement	Testing and Verification Process	Verification status (Y or N)
LED 1. The LED is expected to light up when an input voltage of 3.3V is applied to the series connection of resistor and LED.	**Test:** 1. Using a dc power supply, supply 3.3V to the series connection of resistor and LED with the positive terminal of the 4mm output socket in the banana wire placed in between the digital output of the arduino and the resistor and the negative terminal of the 4mm output socket placed in between the LED and ground. **Positive Result:** 1. The LED lights up when supplied with an input voltage of 3.3V and there is no significant change in brightness over the duration of the test. **Negative Result:** 1. The LED does not light up with 3.3V input, the LED is too dim or there is a huge change in the brightness of the LED.	**Y**
Force Sensitive Resistor(FSR): **1. Analog input signal:** 1.1 The FSR functions as expected and experiences a change in resistance due to an applied external pressure 1.2 The FSR is able to apply a voltage input to the analog input of the microcontroller and convert it to a digital reading. 1.3 FSR recognizes correct human finger	**1. Testing analog input signal:** 1.1 Using a multimeter, the FSR should have an initial resistance that exceeds powers of 10^6 ohms. As more pressure is applied, the resistance should decrease to powers of 10^3 ohms. This will be implemented as an “on/off switch” by recognizing an applied human finger pressure exceeds the breaking point. . 1.2 Test the voltage across the FSR by attaching it to a 5V output pin of the arduino and measure it using a DMM. The analog input	**Y**

pressure so that it can determine if correct finger placement is achieved.	voltage must always fall in the range of 0-5V which is required by the arduino. 1.3 Using the analogRead() function in the arduino programmer, the analog reading will have to exceed 200. The reading is determined by detecting the input voltage of +5 and divide it by 1023 so that each increment represents 0.004V. **Positive Result:** 1.1 The FSR acts as a highly resistive component in its untouched state and decreases in resistance as pressure is applied. 1.2 The FSR provides an ample input voltage (5V max) to the microcontroller and the input is read. 1.3 Finger placement is recognized by the device only at the desired times. **Negative Result:** 1.1 The FSR does not respond to an applied pressure and there is a flaw in the physical nature of the component and needs to be replaced. 1.2 An insufficient voltage is applied and the resistor needs to be changed. 1.3 Program does not recognize the finger placement and informs user that the not is not being played correctly.	
<u>Microcontroller:</u> **1. Testing peripherals:** 1.1 The microcontroller functions as expected at a voltage of 9V. 1.2 The microcontroller is able to accept a voltage input from the force sensitive resistors and convert it to a digital reading for the microcontroller. 1.3 Microcontroller can accept voltage inputs from the push buttons and distinguish between the button being turned on and off. The button will be detected for the function it provides such as navigation between songs and select the song.	**1. Testing peripherals:** 1.1 Using a bench DC supply, the circuit will be powered with a voltage of 9 V. The behavior of the circuit(current and voltage at I/O pins) will be observed and measured. 1.2 This test will involve connecting the force sensitive resistors to the microcontroller and seeing the resistance output by probing the FSR pins with a multimeter to see the change in resistance. If the requirements specified above for FSRs are met, the test is passed. 1.3 Code will be developed for the arduino for testing purposes to make sure that the arduino is getting the proper input from the buttons. After that the code will be executed and the	**Y**

1.4 The microcontroller will output information to the LCD using 7 pins (6 data in pins) 1.5 LEDs light up as specified using high or low input voltage to them. **2. Requirements for ukulele tutoring program** 2.1 Microcontroller should be able to tell if the proper sensor for a note is triggered according to the note location by checking the real time data with the existing code. 2.2 The LEDs light up for a particular note and if the proper note is played, light up for the subsequent note. 2.3 The microcontroller properly functions with the buttons and LCD integrated into the system.	buttons will be pressed and probed in the input pins with a multimeter to make sure that specified voltage is being input as well as the fact that the buttons perform as expected. 1.4 Code will be developed to test the interface of the LCD with the microcontroller. There will be various strings relevant to our ukulele tutor that will be programmed into the code and then the corresponding output on the LCD will be checked to see proper functionality as well as performance of the display (correct coloring and no flickering). 1.5 A program will be developed to activate the LEDs for various locations on the fretboard at various times (delay). **2. Requirements for ukulele tutoring program** 2.1 Correct sensor pressing (which will be determined in our testing) will be performed for a specific note and will be tested by a sketch in the arduino to ensure that it is being pressed. 2.2 The same test mentioned above is performed except this time the LEDs are also connected to the microcontroller. 2.3 For this part we will simulate a user interface by using connecting the buttons and LCD into the system and ensuring that the use can navigate through the LCD using the buttons through the use of arduino test code. The LCD should also display notes to be played after further development of the program. **Positive Results:** 1.1 Minimal change in current and voltage at I/O pins at the given operating voltages. 1.2 FSRs provide ample voltage (7V) as an input that can be read by the microcontroller for	

	use in the program. 1.3 Buttons produce specified output at pins at arduino inputs. The program correctly processes the keypad inputs for navigation, select and menu. 1.4 The strings in the display correspond with the strings in the test code and the LCD displays correct specified color and there is no distortion in the string displayed. 1.5 The LEDs are activated when they are required to (as specified by the user testing them). 2.1 The program recognizes correct sensor pressing to use for the operation of the program. 2.2 The LEDs are activated when they are required to (also as specified by the program). 2.3 The program interprets the buttons function correctly and displays subsequent changes in the LCD. The notes to be played are also displayed on the LCD. **Negative Result:** 1.1 The 9V operating voltage does not cause the circuit to function properly. 1.2 The FSRs do not provide the necessary voltage input for proper functionality. 1.3 The buttons are not processed properly by the arduino and is unable to tell the state of the button (on or off). 1.4 The LCD does not display strings matching the input strings and it is also not functioning properly. 1.5 The LEDs don’t activate when they are specified by the user to do so.	

	2.1 The system fails to recognize whether a sensor input is correct or wrong. 2.2 The LEDs fail to activate for the current note or subsequent note as specified by the program. 2.3 The program doesn't interpret the button functions pushing properly with the LCD integrated with them. The LCD does not display the notes to play.	
LCD 1. The display functions with an input voltage of 5V. 2. It should display numbers, letters, symbols and musical notes in colour without any lag, flickering or disruption. 3. The buttons on the LCD perform their specific task.	**Test:** 1. Using a dc power supply, supply 5V to the VCC pin on the LCD module through a banana wire with a 4mm male output socket. 2. Run a simple program that displays numbers, letters, symbols and musical notes. The program is written in the arduino with the code provided earlier. 3. Run a test program on the arduino which requires the user to press each of the buttons individually to make sure they function. **Positive Result:** 1. The LCD lights up when attached to the power source under the aforementioned scenario. 2. The LCD displays numbers, letters, symbols and musical notes in colour without any lag, flickering or disruption. 3. The buttons function in the manner they are supposed to. **Negative Result:** 1. The LCD does not light up during the test.	**Y**

<u>**LCD**</u> 1. The display functions with an input voltage of 5V. 2. It should display numbers, letters, symbols and musical notes in colour without any lag, flickering or disruption. 3. The buttons on the LCD perform their specific task.	**Test:** 1. Using a dc power supply, supply 5V to the VCC pin on the LCD module through a banana wire with a 4mm male output socket. 2. Run a simple program that displays numbers, letters, symbols and musical notes. The program is written in the arduino with the code provided earlier. 3. Run a test program on the arduino which requires the user to press each of the buttons individually to make sure they function. **Positive Result:** 1. The LCD lights up when attached to the power source under the aforementioned scenario. 2. The LCD displays numbers, letters, symbols and musical notes in colour without any lag, flickering or disruption.	**Y**

Code for the Arduino software

<see 'file 1' on projects page>