Ukulele Instrument Tutor

Design Review
October 5th, 2012
ECE 445- Senior Design
Team 8

Ammar Faiz
Matt DiLiberto
Udit Sharma
TA: Lydia Majure

TABLE OF CONTENTS

PAGE NO.

I. Introduction

1. Motivation:

There are not many musical instruments in the market today which 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.

2. Objectives:

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.

Benefits:

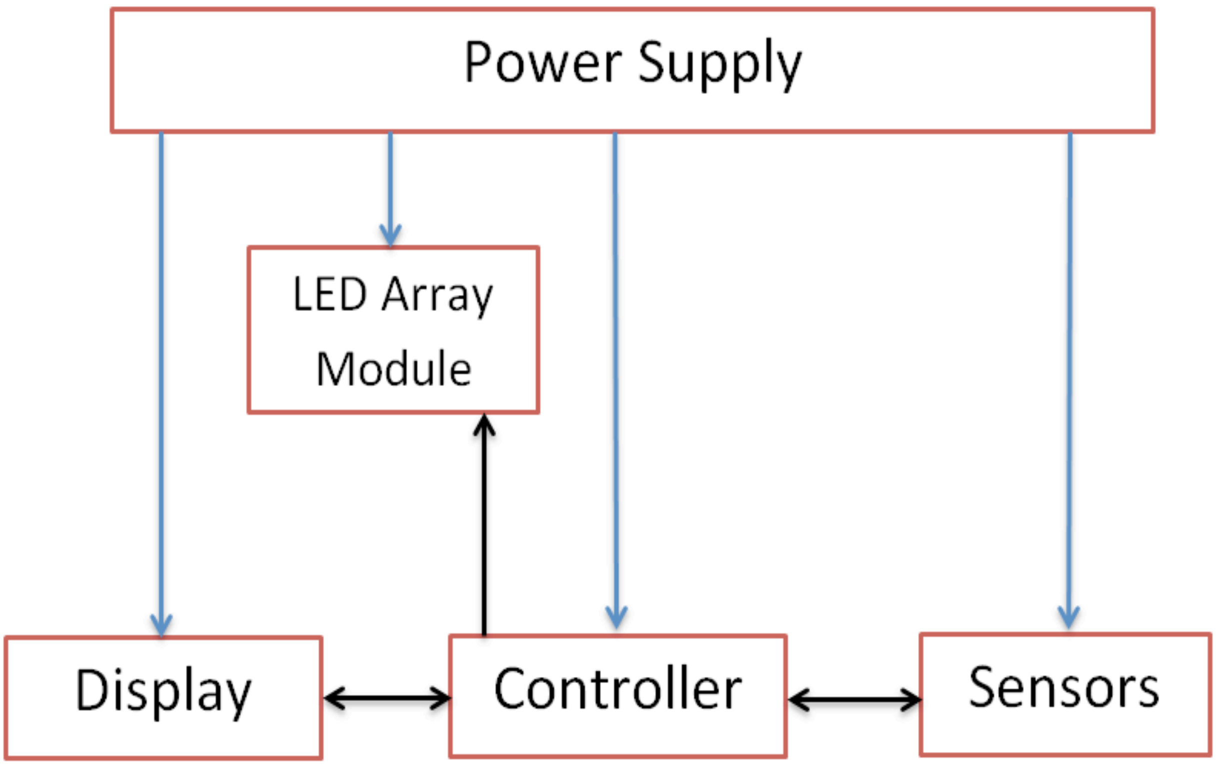
- Helps the user learn how to play the correct note quicker than conventional methods.
- It is a portable system and can be taken wherever desired.
- LEDs display system undermines the need for a musical score sheet.
- Cheaper than other products available on the market, such as Fretlight.
- Its fun!

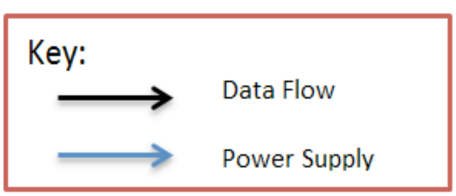
Features:

- 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.

II. Design

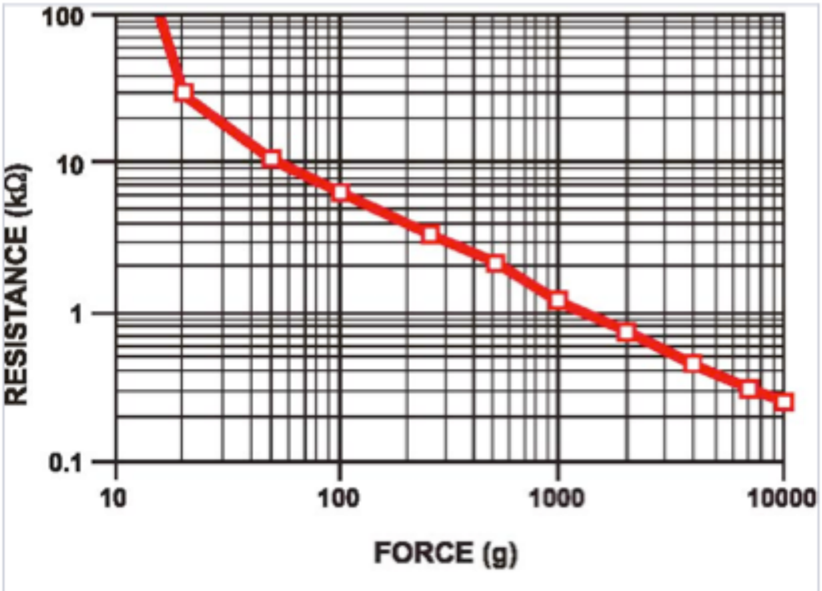
Block Diagram





Block Descriptions

Sensing Unit



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.

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 fretboard. 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 fretboard. 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 which 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 which 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.

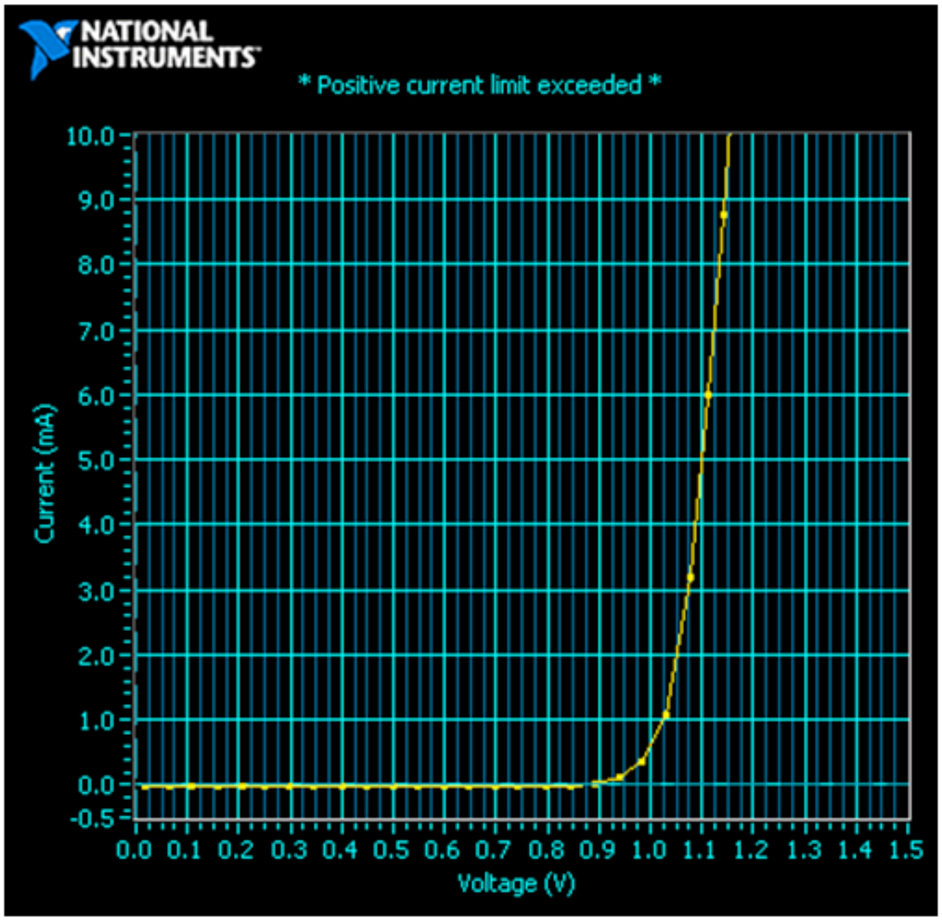
LCD

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 coloured display.

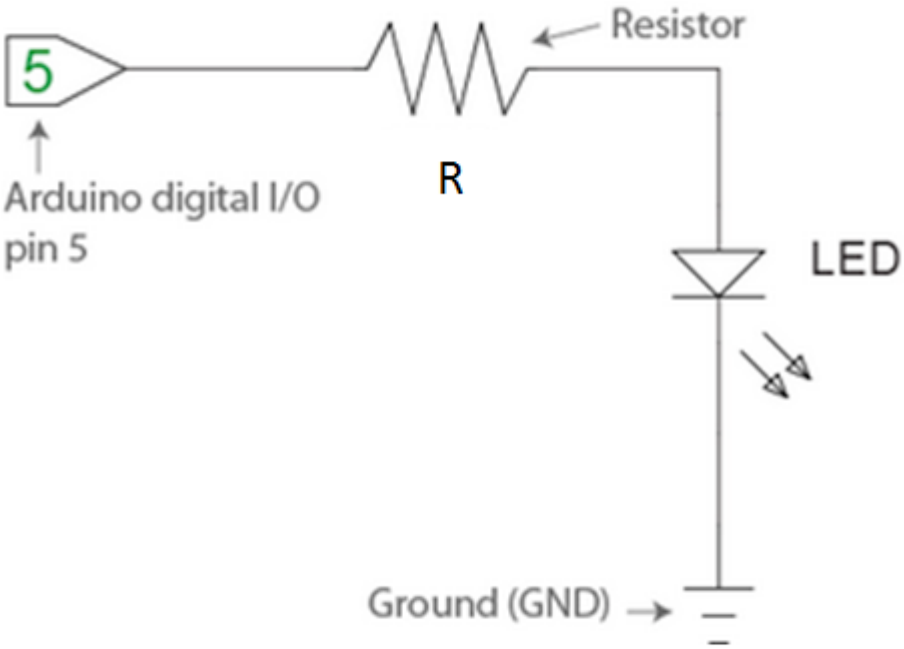
LED Array Module

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 1.7V-2.5V. the peak operational current needed for the LED is 100mA, but for continuous operation, we need 30mA of current to flow through each LED. To provide the LED with 30mA, we need to attach a resistance in series between the battery and the LED.
LED IV curve:



Below is an example of connection for a LED.



To calculate the value of the resistor R, we used

$$R = \frac{5V - LEDVoltage}{30mA}$$
$$R_{min} = \frac{5V - 2.5V}{30mA} = 83.3[?][?][?]$$
$$R_{max} = \frac{5V - 1.7V}{30mA} = 110[?][?][?]$$

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 colour chosen for the LED is red, because it draws the most attention from human beings and also instigates them to work faster.

Power Supply

A 9V battery would be the source of power for all the 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.

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.

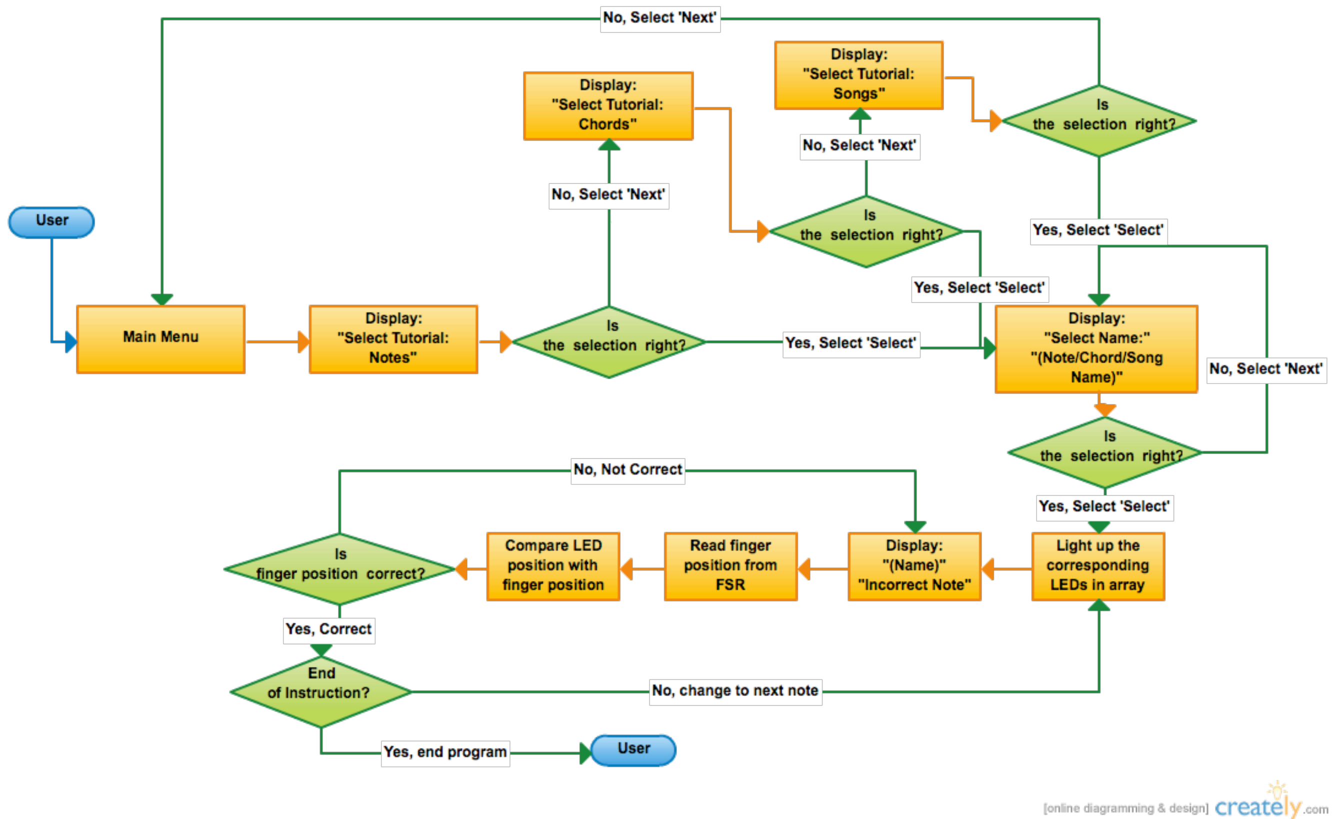
Using the information from the power table, voltage required by the system is

$$V = \frac{Pavg}{Iavg} = \frac{1753.14mW}{220.91mA} = 7.93599V$$

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

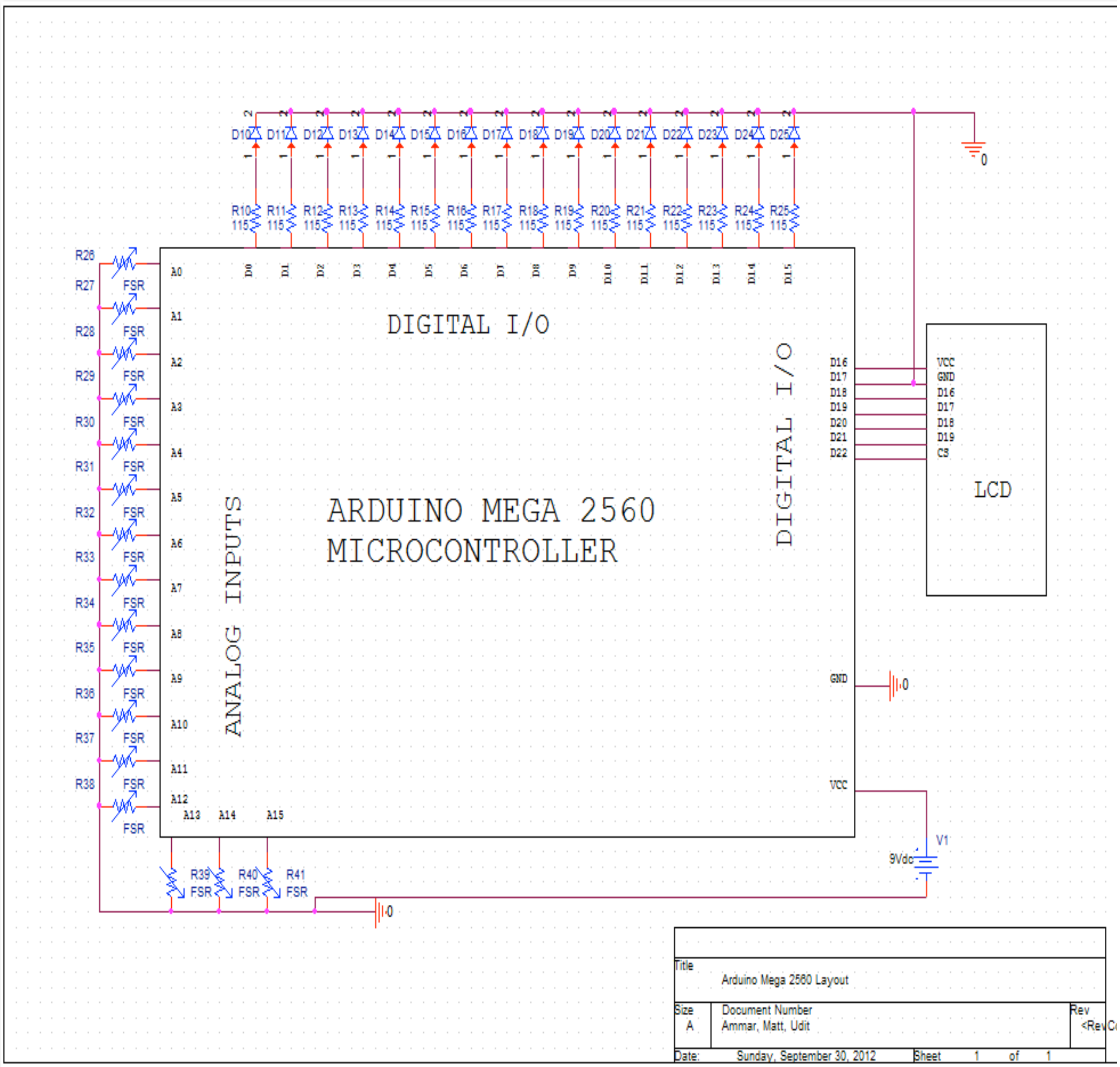
Schematics and Code

Software Flowchart

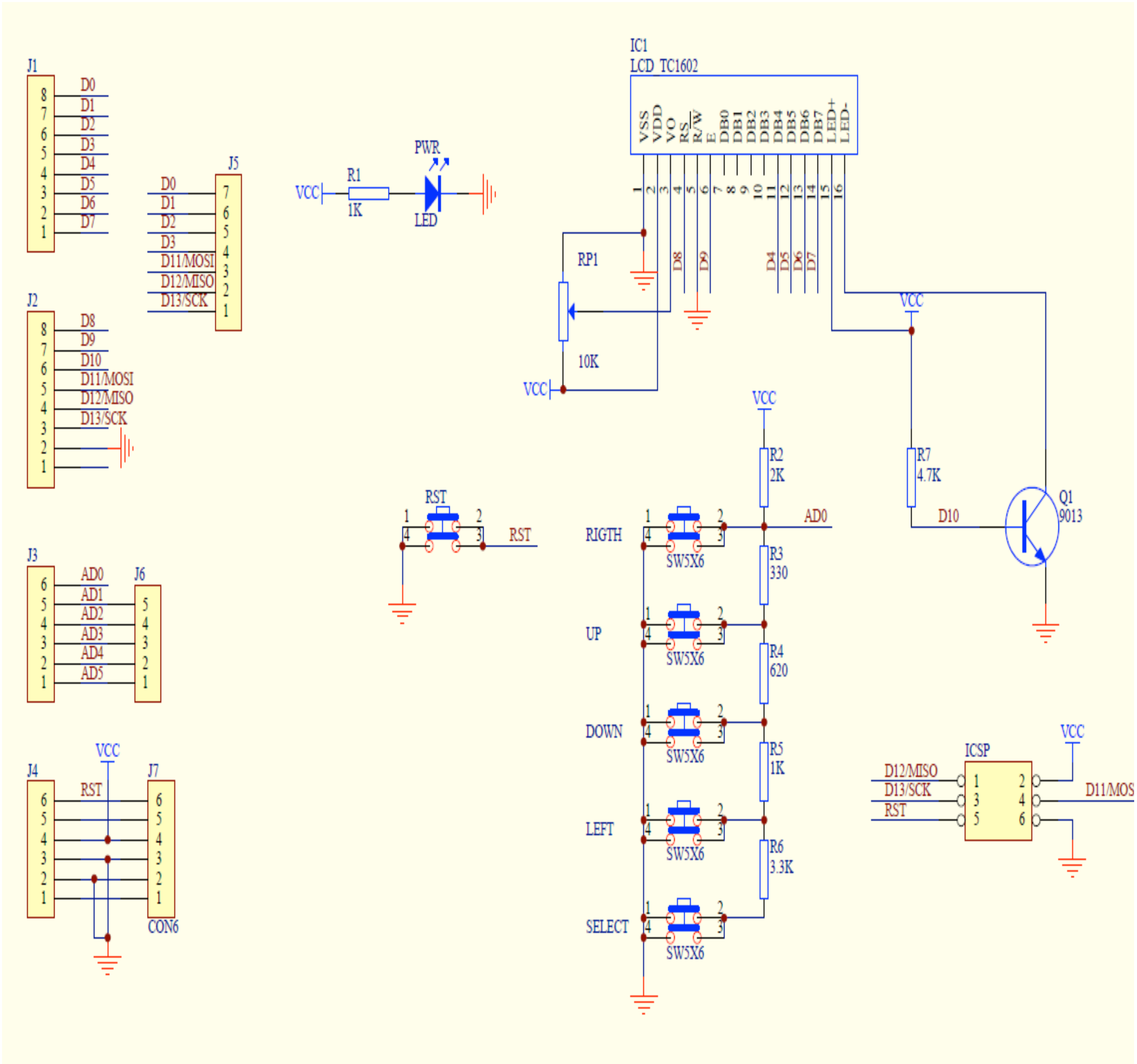


The push buttons on the LCD allow the user to select what option they prefer. They include "Select", "Up", "Right", "Down", "Left" and "Reset".

System Schematic



Display Schematic



Pin Allocation

Pin	Function
Analog 0	Button (select, up, right, down and left)
Digital 4	DB4
Digital 5	DB5
Digital 6	DB6
Digital 7	DB7
Digital 8	RS (Data or Signal Display Selection)

Digital 9 Enable
Digital 10 Backlit Control

Testing codes

Code for LCD:
Below is an example code that is used to test the operation of the LCD.

```
Example use of LCD4Bit_mod library
//
#include <LCD4Bit_mod.h>
//create object to control an LCD.
//number of lines in display=1
LCD4Bit_mod lcd = LCD4Bit_mod(2);
//Key message
char msgs[5][15] = {"Right Key OK ",
                    "Up Key OK    ",
                    "Down Key OK  ",
                    "Left Key OK  ",
                    "Select Key OK" };
int  adc_key_val[5] ={30, 150, 360, 535, 760 };
int NUM_KEYS = 5;
int adc_key_in;
int key=-1;
int oldkey=-1;
void setup() {
  pinMode(13, OUTPUT);  //we'll use the debug LED to output a heartbeat

  lcd.init();
  //optionally, now set up our application-specific display settings, overriding whatever the lcd did in
lcd.init()
  //lcd.commandWrite(0x0F);//cursor on, display on, blink on.  (nasty!)
   lcd.clear();
  lcd.println("KEYPAD testing... pressing");
}
void loop()
{
adc_key_in = analogRead(0);    // read the value from the sensor
digitalWrite(13, HIGH);
key = get_key(adc_key_in);    // convert into key press
            if (key != oldkey)            // if keypress is detected
            {
    delay(50);                  // wait for debounce time
    adc_key_in = analogRead(0);    // read the value from the sensor
    key = get_key(adc_key_in);    // convert into key press
    if (key != oldkey)
    {
```

```
      oldkey = key;
      if (key >=0){
      lcd.cursorTo(2, 0);  //line=2, x=0
            lcd.println(msgs[key]);
      }
    }
  }
  digitalWrite(13, LOW);
}
// Convert ADC value to key number
int get_key(unsigned int input)
{           int k;
            for (k = 0; k < NUM_KEYS; k++)
            {
                        if (input < adc_key_val[k])
                        {  return k;  }
            }
    if (k >= NUM_KEYS)
        k = -1;     // No valid key pressed
    return k;
}
```

Code for FSR:

Below is a simple code for detecting the pressure applied to to an FSR and display its value.

```
int fsrAnalogPin = 0; // FSR is connected to analog 0
int fsrReading;      // the analog reading from the FSR resistor divider

void setup(void) {
  Serial.begin(9600);   // We'll send debugging information via the Serial monitor
 //Set serial boudrate to 9600
}

void loop(void) {
  fsrReading = analogRead(fsrAnalogPin);
  Serial.print("Analog reading = ");
  Serial.println(fsrReading);

  delay(100);
}
```

Code for LEDs

Below is a simple test code that turns a LED on for a second and then turns it off.

```
void loop()                    // run over and over again
{
  digitalWrite(ledPin, HIGH);   // sets the LED on
  delay(500);                  // waits for a second
  digitalWrite(ledPin, LOW);    // sets the LED off
  delay(500);                  // waits for a second
}
```

Code for LEDs, FSR and LCD used with an arduino

Below is a code for a simple program that tests all the three components together.

```
// Force Sensitive Resistor(FSR) to light LED analogly according to pressure
// and display amount of pressure on an 16*2 LCD screen
// Includes public domain LCD display code from official Arduino website

#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // set pins for LCD, set global variables

const int LED = 9;                 // LED connected to pin 9 for analog write
const int FSR = 0;                 // FSR connected to analog pin 0

int level = 0;                     // variable for the level of force applied to the FSR

void setup()                       // run once to setup

{
pinMode (LED, OUTPUT);        // LED set as output, analog pins are automatically set to input

lcd.begin(16,2);                   // set up LCD number of columns and rows
lcd.print(“Pressure Level:”);      // display message on LCD
}
```

```
void loop()                         // repeated routine
{
// first, read the pressure level and set brightness of LED
level = analogRead(FSR)/4;  // read level of pressure on the FSR

// divided by 4 as analog read is 4x sensitive as digital write

analogWrite(LED, level);            // set brightness of LED to pressure level
delay(100);                         // wait to see the effect

//display pressure level on the LCD screen

lcd.setCursor(0,1);                 // set cursor to second line, first spot
lcd.print(level);                   // display level of pressure
}
```

III. Block Level Requirements and Verification

Requirements:

The goals of this project are to develop a device which helps a user learn how to play the ukulele from the start to end of a song. There will be songs stored in the microprocessor and the user inputs will be processed, analyzed and display the subsequent notes to play. For more specific requirements please refer to the 'Verification & Tolerance' section of this report.

Verification and Tolerance

Testing and Verification:
Following table explains what is expected of each of the components and what test procedures will be employed to make sure that the components function properly.

Requirement	Testing and Verification Process
LED 1. The LED is expected to light up when an input voltage of 5V is applied to the series connection of resistor and LED.	**Test:** 1. Using a digital multimeter, supply 5V 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 5V and there is no significant change in brightness over the duration of the test. **Negative Result:** 1. The LED does not light up with 5V input, the LED is too dim or there is a huge change in the brightness of the LED.
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	**1. Testing analog input signal:** 1.1 Using a multimeter, the FSR should have an initial resistance that exceeds 10^6 ohms. As more pressure is applied, the resistance should decrease to less than 10^3 ohms. This will be implemented as an "on/off switch" by recognizing an applied human finger pressure exceeds the breaking point. .

convert it to a digital reading. 1.3 FSR recognizes correct human finger pressure so that it can determine if correct finger placement is achieved.	1.2 Test the voltage across the FSR by attaching it to a 9V battery and measure it using a DMM. The analog input voltage must always fall in the range of 7-12V which is required by the arduino. 1.3 Bench tests will be taken to determine the analog input voltage range. The code will detect the experimentally determined range +- 10%. This is done through sampling. The range will then be used in the program code to verify that the correct note was played. **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 (7V) 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.

Microcontroller: **1. Testing peripherals:**	**1. Testing peripherals:**
1.1 The microcontroller functions as expected at voltages ranging from 8V to 9V and current draw does not exceed 200 mA	1.1 Using a bench DC supply, the circuit will be powered with voltages of 8V, 8.25V, 8.50V, 8.75V and 9 V. At each voltage the behavior of the circuit(current and voltage at I/O pins) will be observed and measured.
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.	1.2 This test will involve connecting the force sensitive resistors to the microcontroller and seeing the voltage inputs by probing the pins with a multimeter to see the voltage that they will provide as inputs. If the requirements specified above for FSRs are met, the test is passed.
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.	1.3 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 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 The microcontroller will output information to the LCD using 7 pins (6 data in pins)	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 LEDs light up as specified using high or low input voltage to them.	1.5 A program will be developed to activate the LEDs for various locations on the fretboard at various times (delay). Additionally, the outputs on the I/O pins for the LEDs will be checked with an oscilloscope to see proper voltage characteristics exhibited.
2. Requirements for ukulele tutoring program	**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.	2.1 Incorrect sensor pressing (which will be determined in our testing) will be performed for a specific note and then the pins for the microcontroller will be probed by oscilloscopes as well as a DMM.
2.2 The LEDs light up for a particular note and if the proper note is played, light up for the subsequent note.	2.2 The same test mentioned above is performed except this time the LEDs are also connected to the microcontroller.
2.3 The microcontroller properly functions with the buttons and LCD integrated into the system.	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

LCD	**Test:**
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.	1. Using a digital multimeter, 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. 2. Numbers, symbols and letter do not get displayed on the screen or there is too much flickering and disruption when the test is carried out. 3. The buttons do not perform the task that it is supposed to do. For example, when “Right” button is pressed, the button does not respond.

Battery	**Test**:
1. The battery should be able to supply a voltage of 9V (±1V) for approximately 1 hours. 2. The battery will be able to supply a minimum current rating of 30mA and a maximum current rating of 200mA	1. A DMM will be used to monitor the voltage across the battery and to measure the current drawn. The time frame is based on the expected maximum playing time. 2. The required minimum current powers the LED lights and the maximum current should not exceed 200mA for the microcontroller. These will be monitored with a DMM so that we don't damage any of the components. Positive Result: 1. The battery will function as expected and be able to supply power to all components. 2. The current does not exceed the maximum described current and no damage is done to the microcontroller or components. Negative Result: 1. The battery is completely depleted before the stipulated time of 1 hours and is not practical for our application. 2. The current drops significantly over time and does not supply enough current to each application. The current exceeds the maximum allotted current and damages the components.

IV. Cost and Schedule
1. Cost Analysis

i. 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

ii. 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
100Ω Resistors	002-100	$0.65	16	$10.40
15Ω Resistors	016-15	$0.54	16	$8.64
TOTAL				$180.06

2. Schedule

	Ammar Faiz	Matt DiLiberto	Udit Sharma
9/16-9/23	Order microcontroller, work on schematics for the microcontroller system, research ways to build Ukulele tutor and attach the microcontroller to the ukulele.	Order parts, work on schematics for system, research ways to build Ukulele tutor	Order parts, work on schematics for system, research ways to build Ukulele tutor
9/24-10/1	Design schematics for microcontroller. Order parts.	Design schematics. Order parts.	Design schematics for LEDs and LCDs. Order parts.
10/2-10/14	Microcontroller-Program and test sensors inputs	Research FSR implementation	Research on ways to code the LEDs and LCD
10/15-10/21	Design microcontroller arrangement on ukulele	Design sensor arrangement on ukulele. Solder FSR to the instrument.	Begin soldering the LEDs and the pressure sensors to the Ukulele.
10/22-10/28	Power system- See requirements for power for microcontroller, test circuit without battery	Calibrate inputs from the FSR to microcontroller to interpret correct finger pressure	Finish up soldering and start calibration of the LEDs
10/29-11/2	Power System- Design PCB, run circuit on battery power, ensure microcontroller is hooked up properly.	Program microcontroller to verify finger placements match the desired notes. Add to PCB design. Program at minimum 3 songs.	Write the program that controls both the LEDs and the LCD.
11/3-11/4	Get ready for Mock Up demo with the circuit on PCB	Get ready for Mock Up demo with the circuit on PCB	Get ready for Mock Up demo with the circuit on PCB
11/5-11/18	Final Debugging and testing	Final Debugging and testing	Final Debugging and testing
11/19-11/25	Thanksgiving Vacation	Thanksgiving Vacation	Thanksgiving Vacation

11/26-12/2	Final paper write up	Final paper write up	Final paper write up
12/3-12/16	Perform Demos and Presentation (singer)	Perform Demos and Presentation (play instrument) (Somewhere Over the Rainbow)	Perform Demos and Presentation (drums)
12/17-12/23	Finish final paper and lab notebook	Finish final paper and lab notebook	Finish final paper and lab notebook

Ethical Considerations:

IEEE Code of Ethics	Considerations
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).
4. To reject bribery in all its forms.	Not applicable.
5. To improve the understanding of technology, its appropriate application, and potential consequences.	Not applicable.
6. To maintain and improve our technical competence and to undertake	Since we are not very experienced in this field of building a product from scratch we will try our best

technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations.	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 others 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.
9. To avoid injuring others, their property, reputation, or employment by false or malicious action.	Not applicable.
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.

APPENDIX:

FSR:

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>.

datasheet

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>.

LEDs:

"SMD LED." *LC-LED.com*. LC LED Corporation, n.d. Web. 01 Oct. 2012. <http://www.lc-led.com/View/itemNumber/101>.

datasheet

SLT-776TUR: POWER TOP RED SMD (3.5mm x 2.8mm) LED. *Asiaopto Corporation*. Asiaopto Industrial LTD. Web. 30 Sept. 2012. <http://www.asiaopto.com/specification/led_smd/SLT-776TUR.pdf>.

LCD:

"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>.

Microcontroller:

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

"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>.

Resistor:

"100 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>.

"15 Ohm 10W Resistor Wire Wound 5% Tolerance." *Parts-express.com*. Parts Express, 2012. Web. 01 Oct. 2012. <http://www.parts-express.com/pe/showdetl.cfm?partnumber=002-100>.

LED test code

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

LCD Test Code

"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>.

FSR Test Code

"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>.

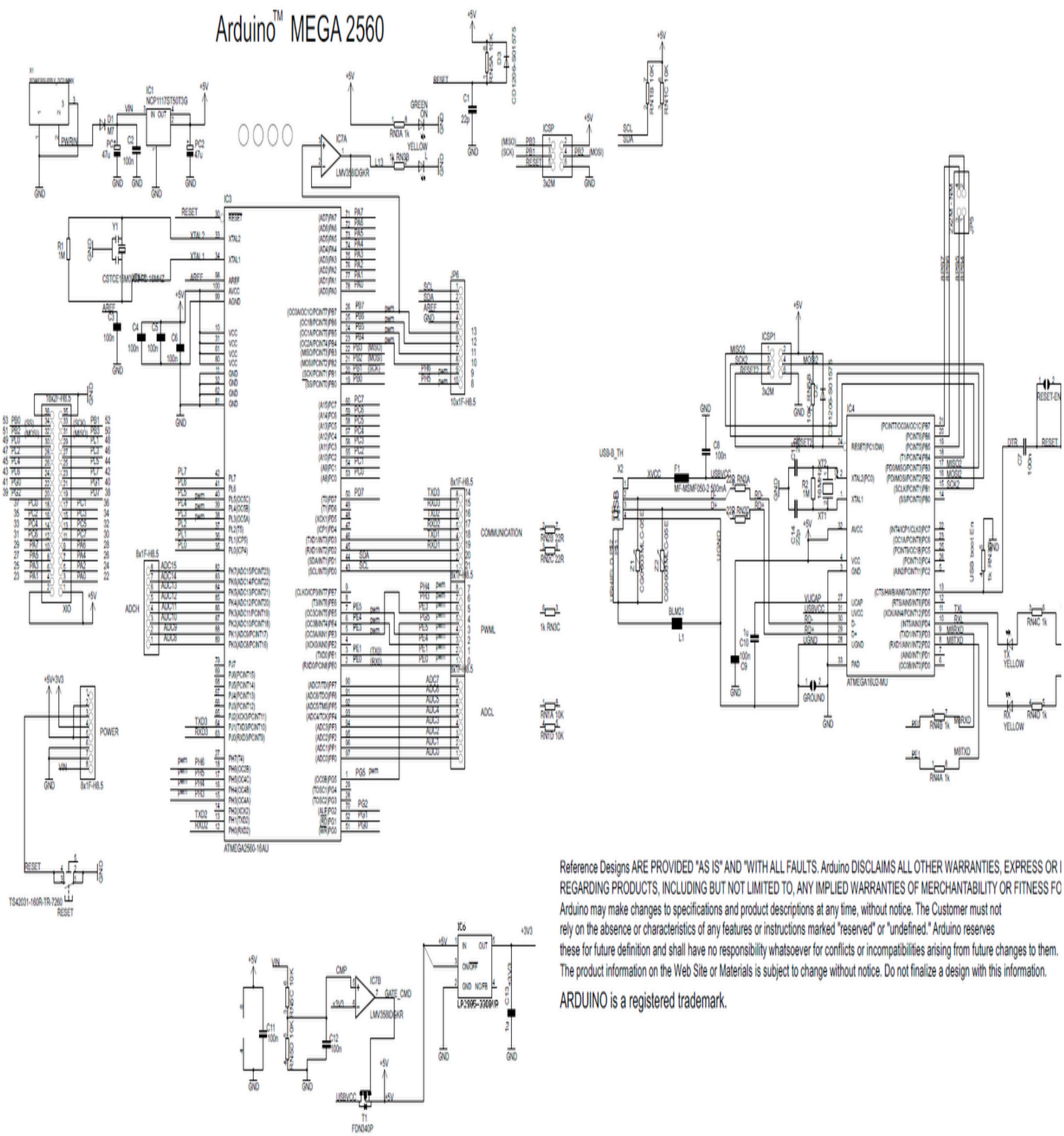
LCD, LED and FSR Test code

"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/>.

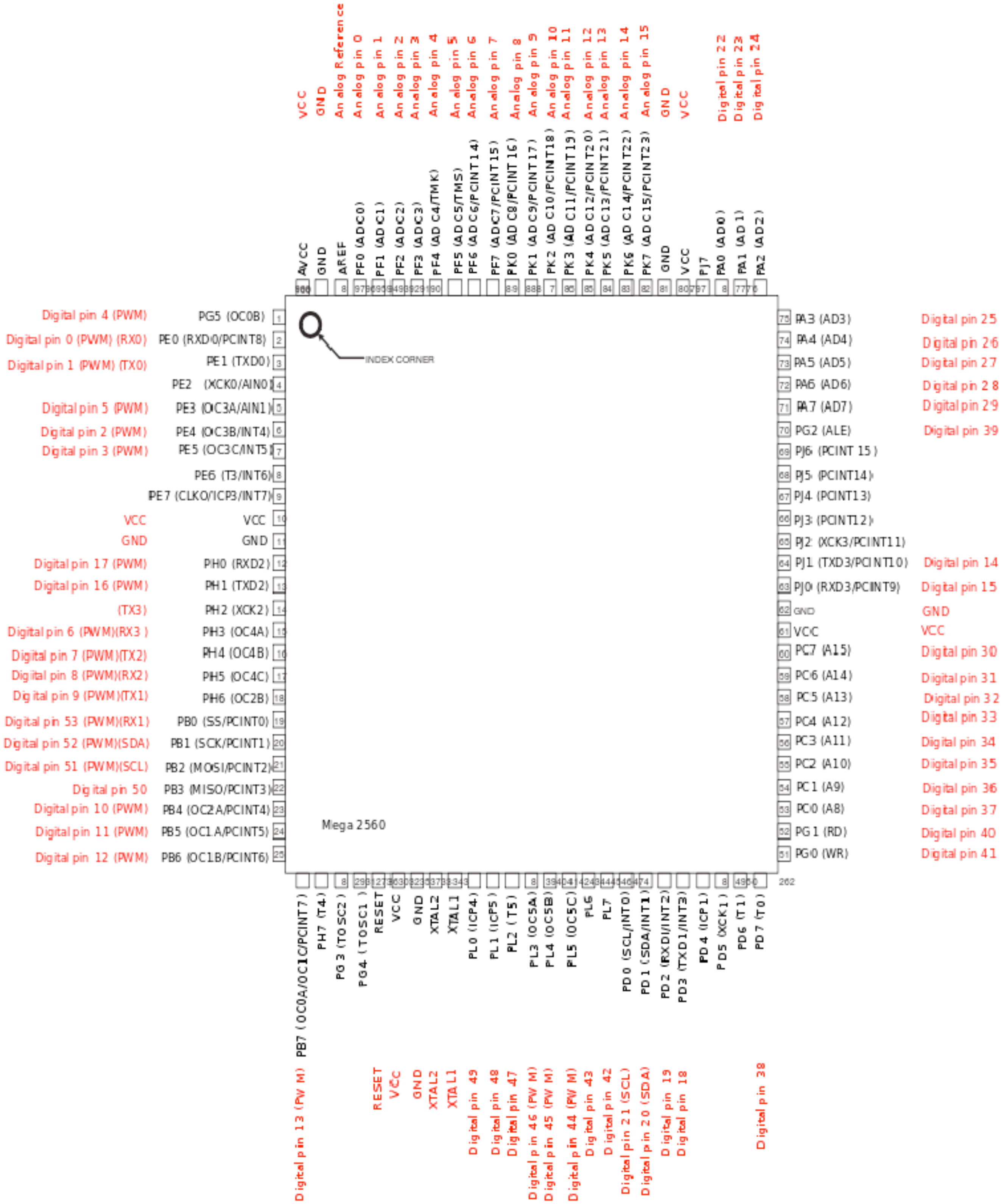
LED curve

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

Arduino Schematic:



Arduino Pin Mapping:



Mega 2560
INDEX CORNER
Digital pin 4 (PWM)
PG5 (OC0B)
Digital pin 0 (PWM) (RX0)
PE0 (RXD0/PCINT8)
Digital pin 1 (PWM) (TX0)
PE1 (TXD0)
PE2 (XCK0/AIN0)
Digital pin 5 (PWM)
PE3 (OC3A/AIN1)
Digital pin 2 (PWM)
PE4 (OC3B/INT4)
Digital pin 3 (PWM)
PE5 (OC3C/INT5)
PE6 (T3/INT6)
PE7 (CLKO/ICP3/INT7)
VCC
GND
Digital pin 17 (PWM)
PH0 (RXD2)
Digital pin 16 (PWM)
PH1 (TXD2)
(TX3)
PH2 (XCK2)
Digital pin 6 (PWM)(RX3)
PH3 (OC4A)
Digital pin 7 (PWM)(TX2)
PH4 (OC4B)
Digital pin 8 (PWM)(RX2)
PH5 (OC4C)
Digital pin 9 (PWM)(TX1)
PH6 (OC2B)
Digital pin 53 (PWM)(RX1)
PB0 (SS/PCINT0)
Digital pin 52 (PWM)(SDA)
PB1 (SCK/PCINT1)
Digital pin 51 (PWM)(SCL)
PB2 (MOSI/PCINT2)
Digital pin 50
PB3 (MISO/PCINT3)
Digital pin 10 (PWM)
PB4 (OC2A/PCINT4)
Digital pin 11 (PWM)
PB5 (OC1A/PCINT5)
Digital pin 12 (PWM)
PB6 (OC1B/PCINT6)
PA3 (AD3) Digital pin 25
PA4 (AD4) Digital pin 26
PA5 (AD5) Digital pin 27
PA6 (AD6) Digital pin 28
PA7 (AD7) Digital pin 29
PG2 (ALE) Digital pin 39
PJ6 (PCINT15)
PJ5 (PCINT14)
PJ4 (PCINT13)
PJ3 (PCINT12)
PJ2 (XCK3/PCINT11)
PJ1 (TXD3/PCINT10) Digital pin 14
PJ0 (RXD3/PCINT9) Digital pin 15
GND GND
VCC VCC
PC7 (A15) Digital pin 30
PC6 (A14) Digital pin 31
PC5 (A13) Digital pin 32
PC4 (A12) Digital pin 33
PC3 (A11) Digital pin 34
PC2 (A10) Digital pin 35
PC1 (A9) Digital pin 36
PC0 (A8) Digital pin 37
PG1 (RD) Digital pin 40
PG0 (WR) Digital pin 41
AVCC VCC
GND GND
AREF Analog Reference
PF0 (ADC0) Analog pin 0
PF1 (ADC1) Analog pin 1
PF2 (ADC2) Analog pin 2
PF3 (ADC3) Analog pin 3
PF4 (ADC4/TMK) Analog pin 4
PF5 (ADC5/TMS) Analog pin 5
PF6 (ADC6/PCINT14) Analog pin 6
PF7 (ADC7/PCINT15) Analog pin 7
PK0 (ADC8/PCINT16) Analog pin 8
PK1 (ADC9/PCINT17) Analog pin 9
PK2 (ADC10/PCINT18) Analog pin 10
PK3 (ADC11/PCINT19) Analog pin 11
PK4 (ADC12/PCINT20) Analog pin 12
PK5 (ADC13/PCINT21) Analog pin 13
PK6 (ADC14/PCINT22) Analog pin 14
PK7 (ADC15/PCINT23) Analog pin 15
GND GND
VCC VCC
PJ7
PA0 (AD0) Digital pin 22
PA1 (AD1) Digital pin 23
PA2 (AD2) Digital pin 24
PB7 (OC0A/OC1C/PCINT7) Digital pin 13 (PWM)
PH7 (T4)
PG3 (TOSC2)
PG4 (TOSC1)
RESET RESET
VCC VCC
GND GND
XTAL2 XTAL2
XTAL1 XTAL1
PL0 (ICP4) Digital pin 49
PL1 (ICP5) Digital pin 48
PL2 (T5) Digital pin 47
PL3 (OC5A) Digital pin 46 (PWM)
PL4 (OC5B) Digital pin 45 (PWM)
PL5 (OC5C) Digital pin 44 (PWM)
PL6 Digital pin 43
PL7 Digital pin 42
PD0 (SCL/INT0) Digital pin 21 (SCL)
PD1 (SDA/INT1) Digital pin 20 (SDA)
PD2 (RXD1/INT2) Digital pin 19
PD3 (TXD1/INT3) Digital pin 18
PD4 (ICP1)
PD5 (XCK1)
PD6 (T1)
PD7 (T0) Digital pin 38