Design Review for Electric Air Ukulele

ECE 445: Senior Design at University of Illinois at Urbana – Champaign

Spring 2012

Developers

Ivan Setiawan & Satyo Iswara

February 18, 2012

TA : Jane Tu

Table of Contents

1. **Introduction ……………………………………………………….2**
2. **Objective ………………………………………………………..2**
3. **Benefits ………………………………………………………….2**
4. **Features …………………………………………………………2**
5. **Block Diagrams …………………………………………………3**
6. **Block Descriptions ……………………………………………...4**
7. **Flow Charts ……………………………………………………..7**
8. **Theory and Calculations ……………………………………….8**
9. **Performance Requirements and Verifications ………………………..12**
10. **Tolerance Analysis …………………………………………………16**
11. **Project Cost ………………………………………………………..18**
12. **Schedule ……………………………………………………………19**
13. **Ethics Considerations ……………………………………………..21**
14. **Appendix ………………………………………………………..22**

Introduction

Our final project in this class is to implement an electric air ukulele. We changed our project from developing electric air guitar to electric air ukulele. The main reason we develop air ukulele is to improve the detail. Since implementing guitar with six strings would have an ambiguity for the sixth and five strings, we decide to change our design into a ukulele, which have four strings. The air ukulele then can be more realistic with the feature that each string can be pick individually with a finger correspond to a string. Other advantage is a real ukulele needs to be calibrated often while air ukulele doesn't which make it users friendly. We also hope that the air ukulele will make ukulele training easier.

Objective

Our objective with this project is that we can play a music with an ukulele by using the sensing distance. The distance between our left and the right hands will determine which notes are being played. Then, the notes will be sent into the microcontroller for noise filtering. After filtering process, the smooth sound will be generated to the speaker.

Benefits

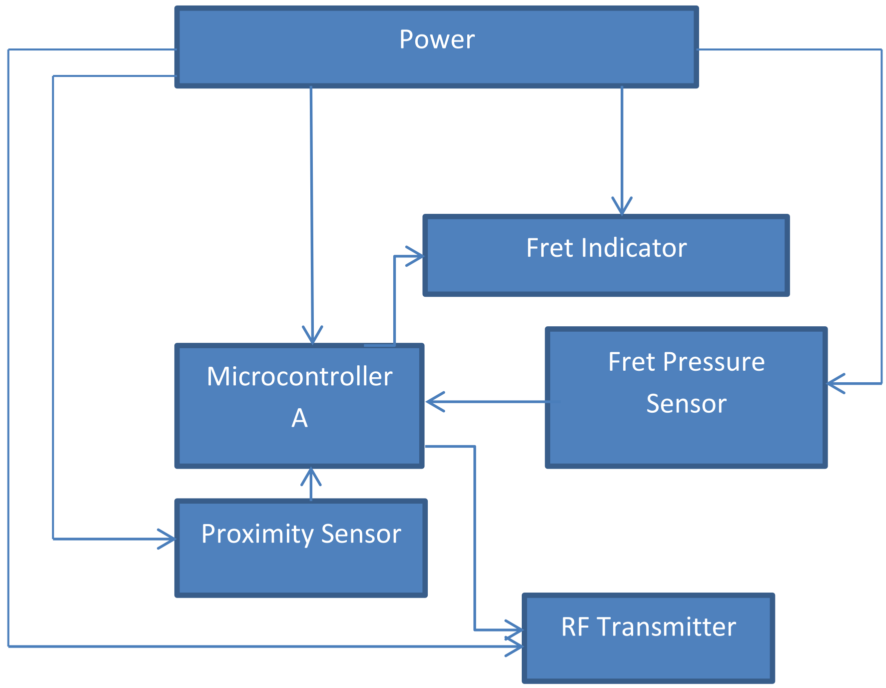
- Lightweight and compact
- Simplify the music playing since the frets are determined by the distance between our hands
- Strings are pre-calibrated and does not need other calibration

Features

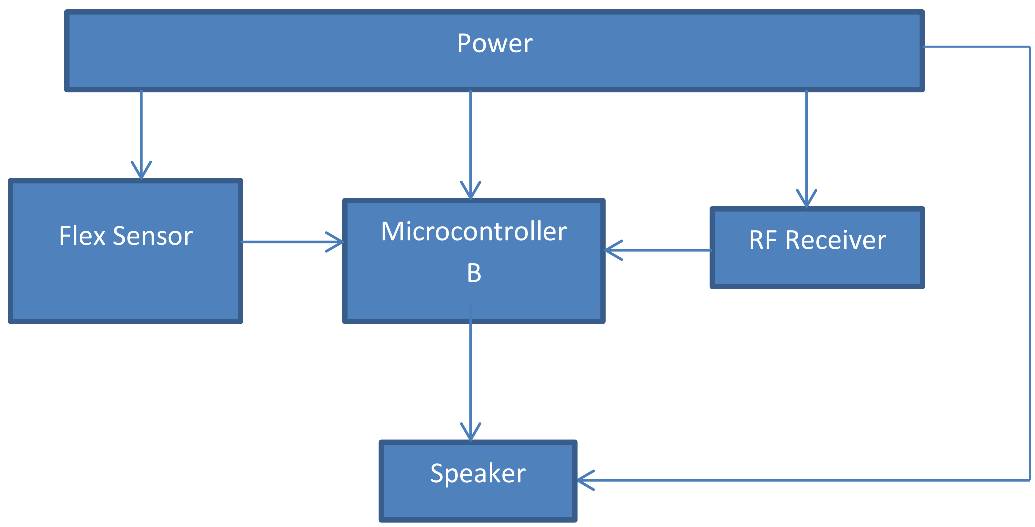
- The device can recognize which string is being picked
- Realistic fret width
- Led array for fret indicator

Block Diagram

Left Hand Block Diagram



Right Hand Block Diagram



Block Diagram Description

Power

For this project we are using AA batteries. We are using AA batteries because most of the component use 5V DC input. The power then connected to resistor to ensure current draw would not exceed each component limit.

RF Transmitter

We use RF transmitter WRL- 10534 with frequency of 434MHz. This transmitter will receive data from the fret board. WRL- 10534 connected to Microcontroller which sends serial digital data. Microcontroller, Arduino Uno will use VirtualWire library to implement digital signal send.

RF Receiver

We use RF transmitter WRL-10532 with frequency of 434MHz. RF receiver will receive the data sent from the transmitter wireless and will send the data serially to Microcontroller. Microcontroller, Arduino Uno will use VirtualWire library to decode the digital signal received.

Microcontroller

We use two microcontroller and both are Arduino Uno. The main functionality of Microcontroller A is to calculate distance, fret position, fret width and fret played. The inputs to Microcontroller A are proximity sensor and touch pad. The outputs of Microcontroller A are the fret position which use to calculate fret border condition and also the LED for user display. Microcontroller A then send the pressed note through RF transmitter. Microcontroller B is mainly use to play the sound. Microcontroller B takes input from RF receiver and flex sensor. Using the digital data from RF receiver Microcontroller B decode it and depends on flex sensor input sends digital signal to the speaker. The complete function of both microcontroller are described in flowchart.

Speaker

We use AS5008-32 manufactured by PUI Audio. After the notes played are filtered with Karplus-String Strong synthesis, the filtered notes will be sent to the speaker to generate the sounds.

Flex Sensor

Four Flex sensors will be placed in the fingers to determine which string is being picked. The flex sensor changes its resistance when bent. So, when the point finger is bent, we know that the first string is being picked. The flex sensor will then input the string to be picked into the microcontroller B so that the data can be decoded. For the flex sensor, we use SEN-08606 with the bend resistance range 60K - 110K ohms.

Proximity Sensor

Proximity sensor will be used to determine the frets being played based on the distance. For proximity sensor we use MB1040 LV-MaxSonar. The proximity sensor works by sending ultrasonic wave (42kHz) and reading the echo. The output of proximity sensor is digital signal in form of ASCII code. The output codes are based on the distance detected. The output then will be serially transmitted and send to microcontroller.

Fret Indicator

For fret indicator we are using linear LED array. The LED will receive signal from microcontroller to determine which LED is on. This LED analogous to the dots appears on the top side of ukulele as fret markers.

Fret Pressure Sensor

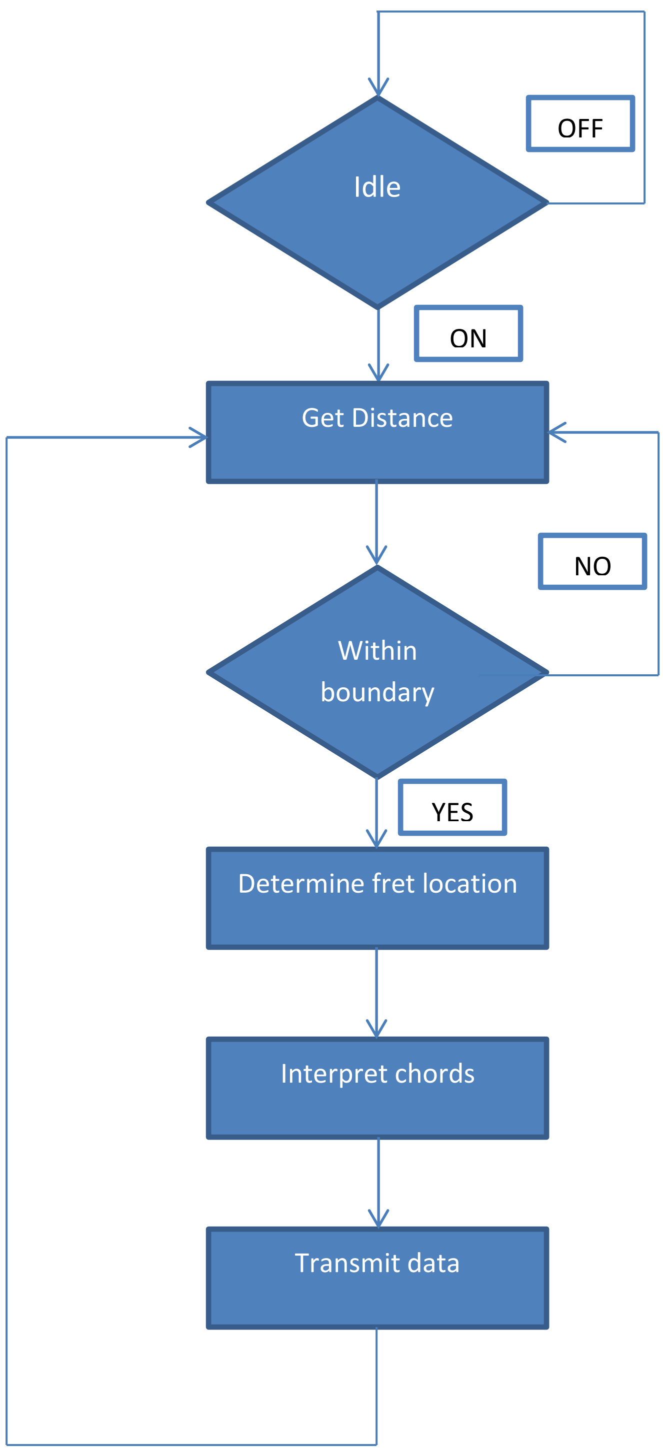
For fret pressure sensor we are using membrane potentiometer SEN-08607. The membrane potentiometer will detect chords by detecting hand pressed. When a part of membrane potentiometer is pressed, the resistance will vary between .1kOhm to 10kOhm. The membrane potentiometer then connected to microcontroller to determine which fret pressed.

Schematics

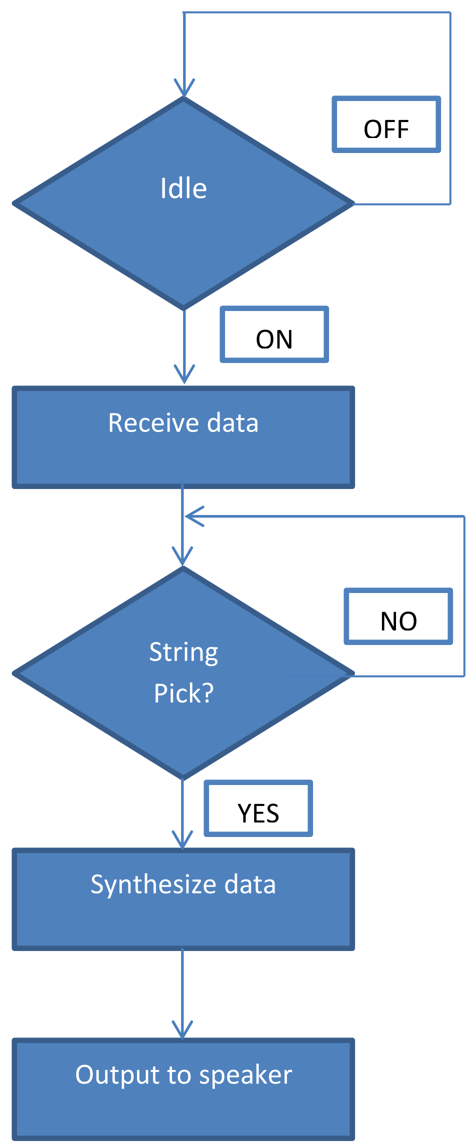
See attachment

Flowcharts

Left-Hand flowchart



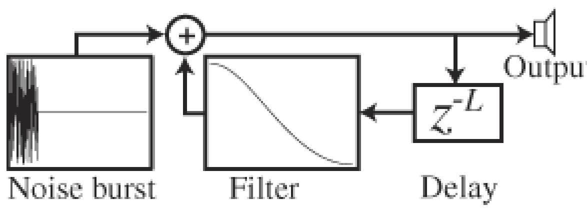
Right-hand flowchart



Calculation and Theory

Karplus-Strong String Synthesis

Stringed instruments like guitars are the most simulated and synthesized musical instruments. One of the way to make a physical simulation model of the strings is the Karplus Strong synthesis model. This model is based on the feedback loops. Waveforms are looped through a delay line filter, which simulates the sounds of plucked strings. The Karplus-Strong method starts out with the generation of a short excitation waveform. The noise burst then gets averaged out The excitation serves as the output, but at the same time is fed back into a delay line. The delay line is typically the same sample length as the excitation waveform.The delay line output then passes through a filter (typically low pass filter), and then simultaneously goes back into the delay line and back into the output



For our design, it is critical to use the Karplus-Strong String synthesis because the sound generated will will have the property of sounds of a real guitar. Also, averaging the signals from the strings will effectively get rids of the high frequency. This synthesis will be implemented in the microcontroller.

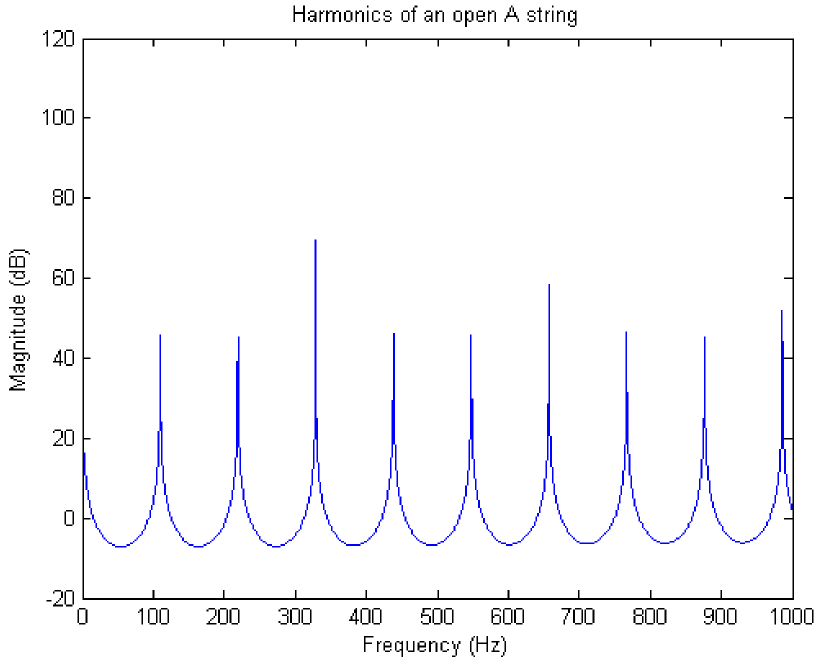
The plucked string algorithm can be implemented as below:

$$y[n] = x[n] + \frac{y[n-N] + y[n-(N+1)]}{2}$$

where N is the felay line. N = fs / fo

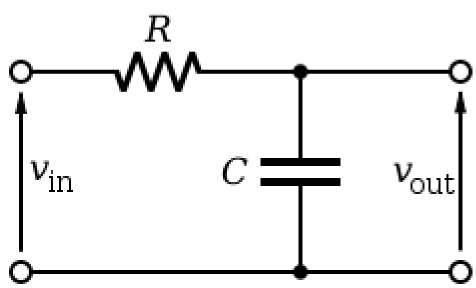
fs is the sampling rate and fo is the fundamental frequency of the string.

Below is the simulations waveform of the guitar chords of C harmonics using MathLab:



Low Pass Filter

The Karplus Strong string synthesis uses the RC low pass filter since the frequency won't exceed 100kHz. This filter has a purpose to pass the desired frequencies signals and block the unwanted high frequencies signals. Below is the schematic of the RC low pass filter:



The cutoff frequency can be calculated as:

$$f_c = 12RC$$

The fundamental frequency is the frequency where the filter changes its behaviors from passing the low frequency and stopping the high frequency.

Power Safety

For our power source we are using four AA batteries connected in series so that we get the desired 6 V DC input. Each component will be connected to resistor to reduce the voltage input and also to match the current input each component draw.

For Arduino microcontroller, the battery connected directly to the Vin. Arduino have its own power control   For the logic voltage the input have to be at 5V so simple voltage divider will be use.

$$V_{out} = V_{in} * R_2 / (R_1+R_2) \quad (1)$$

$$6V = 5V * R_2 / (R_1+R_2)$$

$$I = V_{in} / (R_1+R_2) \quad (2)$$

$$40 \text{ mA} = 6 / (R_1+R_2)$$

so,

$$R_1+R_2 = 150 \text{ Ohm}$$

solving for R1 and R2 we get

$$R_1 = 25 \text{ Ohm}$$

$$R_2 = 125 \text{ Ohm}$$

For proximity sensor we need 5V Vin and 3mA current. the same simple voltage divider circuit is also needed. Using equation (1) and (2) the value of R1 and R2 are

$$R_1 = 333.33 \text{ Ohm}$$

$$R_2 = 1666.67 \text{ Ohm}$$

The pressure sensor act like variable resistor that have a constant .1kOhm when not active (i.e pressed). The resistance will increase up to 10kOhm. The input voltage should be 5V and the current max should be 40mA because the output from pressure sensor goes directly to Arduino. Using equation (1) and (2) the value of R1 and R2 are

$$R_1 = 25 \text{ Ohm}$$

$$R_2 = 125 \text{ Ohm}$$

For both RF transmitter and receiver both use 5V input and draw 8mA of current. Using equation (1) and (2) the value of R1 and R2 are

$$R_1 = 125 \text{ Ohm}$$

$$R_2 = 625 \text{ Ohm}$$

Flex Sensor have the same function as pressure sensor. Flex sensor will have resistance of 10kOhm if it straight and with bending the resistance will increase up to 60kOhm. The resistances uses for flex sensor implementation are

$$R_1 = 25 \text{ Ohm}$$

$$R_2 = 125 \text{ Ohm}$$

For speaker AS5008-32 the maximum input power is 0.4Watts. The correlation between power and voltage are given.

$$P = V^2 / R \tag{3}$$

$$.4 \text{ W} = (6\text{V})^2 / R$$

$$R = 90 \text{ Ohm}$$

Performance Requirement and Verifications

Requirement	Verification
Microcontroller (A)	
1. The microcontroller functions correctly under voltage range of 7V – 12V 2. Be able to send the signal at 4800 bps. a. Correctly decode one octave frequency range (262 Hz – 523 Hz) b. The delay time between the data input from the proximity sensor must be less than 200 ms c. Successfully output the analog signal into the fret indicator in the range of 0 – 5 volts with input above 2.5 V considered as high 3. Be able to receive the signal from the fret potentiometer under 0 – 5 volts	1. Connect to the power supply at 7, 9, 12 V, and verify the correct input with function generator and correct output with LED and oscilloscope 2. Code the microcontroller with 4800 bps data. Then, we connect the digital pin out of the microcontroller to the oscilloscope. Using the trigger function and sample the time in 10ms, we can see the data bits are generated. a. Code the microcontroller with one octave range every 1 second, and then connect the digital pin out to the oscilloscope. Using math function in the oscilloscope and convert the signal into frequency domain, we can measure the peak to peak voltage. b. Program the microcontroller with the delay phase less than 200ms and connect it to the oscilloscope to convert the signal into frequency domain. Then, we can use the delay sweep function to verify if the delay is less than 200ms. c. Wire the digital output pin of the microcontroller to the digital high (5V) and digital low (0V), and wire the LED at the output pin to check if it is on or off. 3. Wire the digital output pin of the microcontroller to the digital high (5V) and digital low (0V), and wire the LED

	at the output pin to check if it is on or off
Microcontroller (B)	
1. The microcontroller functions correctly under voltage range of 7V – 12V 2. Be able to generate correct notes within one octave in frequency range of 262 Hz – 523 Hz a. Correctly generate note C (262 Hz) b. Correctly read the digital encoding of note C (262 Hz) c. The sounds must be damped after 10ms using the Karplus Strong synthesis d. The delay time between the input and the output must be less than 200ms e. Successfully receive analog signal from flex sensor in range 0 – 5 V with input above 2.5 V consider as high	1. Connect to the power supply at 7, 9, 12 V, and verify the correct input with function generator and correct output with LED and oscilloscope 2. Connect the output to the speaker and be able to hear the correct sound of the entire octave in a sequence and the expected output must be match to the tuner a. Code the microcontroller to generate note C every 1 second and connect it to the oscilloscope. Using the math function in the oscilloscope and convert the signal to frequency domain, we can measure the peak of the frequency pulse b. Wire the digital input pin of the microcontroller to the digital high (9V) and digital low (0V), and wire the LED at the output pin to check if it is on or off. c. Program the synthesizer to the microcontroller and connect it to the oscilloscope. If the amplitude of the note C is reduced by 50% of the initial and less than 10% after 10ms, then the synthesizer behaves correctly. d. Program the microcontroller with the delay phase less than 200ms and connect it to the oscilloscope to convert the signal into frequency domain. Then, we can use the delay

	sweep function to verify if the delay is less than 200ms. e. Wire up the analog input pin of the microcontroller with 0, 2.5, and 5V. Then, wire the LED at the analog pin output to check if it is on or off.
Power	
1. Four AA batteries will be used to generate 12V DC input power since most of the components must have voltage range of 5V – 12V	1. Connect battery to volt meter and check the output voltage with voltmeter. The voltage measure should stay constant at 12 V and current draw 100mA.
Proximity Sensor	
1. Proximity sensor function correctly under 5 V input 2. Be able to generate 2.5 – 5 volts output for distance between 30 cm to 2 m. a. Be able to generate correct output at fixed distance of 50 cm	1. Connect to power supply and read output change indicating device working 2. Connect to power supply and vary distance between proximity sensor and the object. The voltage will vary 0.98mV / inch
Fret Potentiometer	
1. Fret potentiometer able to generate resistance ranging from 1 kΩ to 10 kΩ. a. Generate 1 kΩ when not pressed	1. Connect the potentiometer into the digital multi-meter, and then check the resistance output. a. Press the potentiometer then checks the output resistance with Digital Multimeter. The output resistance is 1 kΩ.
Fret Indicator	
1. LED light up under 5V input	1. Connect the LED into the proto board, and then gives an input of 5 volts. Check the LED if it is on (5 volts input) or off (low input)
RF Transmitter	
1. RF transmitter will accept an input and transmit it through the antenna. Signal should be receivable at least 1 meter away. The turn on time for the	1. Digital signal will be sent from the microcontroller to the transmitter. A distance test will measure the signal to noise ratio of the receiver to determine

transmitter after being powered up should be 1 ms.	the maximum RF transmission distance. Additionally, the unit will be powered on and data will be inputted. Then, the antenna output will be measured to ensure it can transmit within 1ms power-on.
RF Receiver	
1. The RF receiver should receive signal from RF transmitter for at least 1 meter away. 2. All of the bit patterns transmitted should be recognized by RF receiver	1. We will continuously transmit data and then measure the output vs. distance. The point at which the data outputted cannot be recognized will be the maximum distance. 2. At the specific distance, all byte patterns from 0x00 to 0xFF will be sent to ensure that each bit pattern can be read correctly
Flex Sensor	
1. Flex sensor able to generate resistance ranging from 10 kΩ to 30 kΩ. a. Generate 10kOhm when not bend	1. Connect the flex sensor into the digital multi-meter, and then check the resistance. a. When the flex sensor is not bent down then checks the output resistance with the DMM. The output resistance is 10kΩ.
Speaker	
1. Speaker are able to generate correct notes in one octave range (specific range) a. Be able to generate note C (262 Hz)	1. Connect the speaker into the function generator, and then vary the frequency input to see the peak to peak. a. Code the C notes into the microcontroller, and then connects it to the speaker.

Tolerance Analysis

Block Name	**Testing Focus**	**Acceptable Result Ranges and Confirming Operation**
Power • Supply power to the rest of the system	The most important for power are the voltage and the maximum current. The voltage and the current will determine total power supplied. We need to ensure that each device is supplied with appropriate power in order to function properly.	The voltage has 9V output and maximum current of 100mA for the power module.
Microcontroller (A) • Compute the location of the fret • Send output signal to transmitter	For microcontroller A, the computation of the fret location is the most important part because the fret will be sent into the right hand for further decoding. In order to compute the fret location, we will code the microcontroller.	The microcontroller A can send the data to the transmitter within one octave range (262 Hz – 523 Hz).
Microcontroller (B) • Decode the data receive • Send output signal to generate sound	For microcontroller B, the decoding is the most important part because the microcontroller will output the generated the sound into the speaker.	The microcontroller B can decode and receive the data within one octave range (262 Hz – 523 Hz).
Fret Indicator • Indicate fret number	Voltage is the most important parameter in the fret indicator because when the input is low (0V), the indicator is off. When it is high, 5V, the indicator is on.	The fret indicator has a voltage range of 0 – 5 volts.
Proximity Sensor • Proximity sensor is used to detect certain ranges	The proximity sensor must be able to detect object in the range of 30 cm – 2 meters. The output of the sensor is the voltage where the closer the object to the sensor, the voltage read will be smaller.	The proximity sensor has .772V output for object detection at 2m and .116 V for 30 cm detection. So the expected range of proximity sensor should be between .116V and .772V with output increase as object detected farther.
Fret Potentiometer • Act as voltage divider to detect fret being press	Resistance is the most important parameter. The change of resistance on the sensor depends on whether the sensor is pressed or not. And, the change of resistance will determine which fret is being pressed.	The sensor has 1kΩ - 10kΩ resistance range. When the sensor is not pressed, the resistance is 1 kΩ. When the sensor is pressed, the maximum resistance is 10 kΩ.
RF Transmitter – Receiver	For the transceiver, frequency is	The transceiver can transmit and

• Mainly use to transfer data wirelessly • Data contain which fret is being pressed	the most important parameter because the receiver must be able to receive the data from transmitter with specific frequency. For example, if note C (262 Hz) is sent from transmitter, the receiver must be able to receive the data.	receive the data within one octave range (262 Hz – 523 Hz).
Speaker • Speaker is used to generate sound	The most important parameter in the speaker is the resistance because resistance determines the power output delivered to the speaker.	The speaker has 8Ω resistance with tolerance of ± 0.01Ω.
Flex Sensor • Act as voltage divider to detect string pick	Resistance is the most important parameter. The change of resistance on the sensor depends on the amount of bend on the sensor. And, the more the bend, the more the resistance value. The change in resistance will determine which string is being picked.	The flex sensor has 10kΩ - 30kΩ resistance range. When the sensor is not bent, the resistance is 10 kΩ. When the sensor is bent, the maximum resistance is 30 kΩ.

Project Cost

Part Cost

No	Component	Part Number	Manufacturer	Quantity	Total Cost
1	Proximity Sensor	MB 1040 LV MaxSonar	MaxBotix	1	$29.95
2	Battery	AA	Alkaline	4	$4.00
3	Speaker	AS5008-32	PUI Audio	1	$12.00
4	Flex Sensor	SEN-08606	Spectra Symbol	4	$51.80
5	RF Receiver	WRL-10532	Wenshing	1	$4.95
6	RF Transmitter	WRL-10534	Wenshing	1	$3.95
7	Microcontrollers	Arduino Uno	Arduino	2	$80.00
8	Resistors, Capacitance, Inductance	Various	Various		
9	Membrane Potentiometer	SEN-08607	Softpot	4	$52.00
10	LED	COM-00533		20	$7.00

Labor Cost

Members	Salary	Hours	Total x 2.5
Ivan	$28	150	$10,500
Satyo	$28	150	$10,500

Total Cost = Total part cost + total labor cost

= $214.60 + $21,000

= $21,214.60

Schedule

Week	Tasks	Group Members
19 Feb - 25 Feb	Finish up the design review	Ivan, Satyo
	Generate with noise for speaker for initial starting point	Ivan
	Implement the button to get frets	Satyo
26 Feb - 3 Mar	Filter white noise to get certain frequency	Ivan
	Work on the transmitter and receiver	Satyo
	Filtering out noise from RF transmitter and receiver	Satyo
4 Mar - 10	Finish up with the Karplus-Strong synthesis filtering	Ivan

Mar		
	Testing the transmission data from RF transmitter to RF receiver	Satyo
11 Mar - 17 Mar	Individual progress report	Ivan, Satyo
	Work on the flex sensor	Ivan
	Work on the proximity sensor to get the specific note based on the distance	Satyo
18 Mar - 24 Mar	Spring Break	Ivan, Satyo
	Finish up with the flex sensor	Ivan
	Finish up with the proximity sensor	Satyo
25 Mar - 31 Mar	Integrate all of the sounds and combine multiple chords to be outputted into a single speaker	Satyo
	Work on the interface between the microcontroller and the speaker	Satyo
	Mock up demo	Ivan, Satyo
1 Apr - 7 Apr	Continue working with the interface between microcontroller and the speaker	Satyo
	Integrate all of the sounds and combine them	
8 Apr - 14 Apr	Debugging each components	Satyo
	Assemble all of the components together	Ivan

15 Apr - 21 Apr	Extreme case behavior test	Ivan
	Algorithm checking	Satyo
22 Apr - 28 Apr	Finish testing and debugging	Ivan, Satyo
29 Apr - 5 May	Preparing for final Demo	Satyo
	Write final report	Ivan
6 May - 12 May	Final week	

Ethics Considerations

There are several ethical issues to consider when designing our idea. Our group adheres to the IEEE Code of Conduct and Ethical Guidelines. Our project is intended to improve the understanding of technology, applications, and the consequences. For example, this project combine the flex sensor, proximity sensor, and as well as microcontroller to design such a music instrument. Furthermore, all voltages will be safely regulated and all components will be shielded from user contact. So, this project accepts the responsibility in making decisions with the safety, health, and welfare of the public.

Appendix

"2500 mAh NiMH Battery - AA - SparkFun Electronics." *News - SparkFun Electronics*. N.p., n.d. Web. 21 Feb. 2012. <http://www.sparkfun.com/products/335>.

"Arduino Uno SMD - SparkFun Electronics." *News - SparkFun Electronics*. N.p., n.d. Web. 12 Feb. 2012. <http://www.sparkfun.com/products/10356>.

"Basic LED - Red - SparkFun Electronics." *News - SparkFun Electronics*. N.p., n.d. Web. 21 Feb. 2012. <http://www.sparkfun.com/products/533>.

"Flex Sensor." *Sparkfun.com*. spectrasymbol.com, n.d. Web. 2 Nov. 2012. <www.sparkfun.com/datasheets/Sensors/Flex/FLEXSENSOR(REVA1).pdf>.

"Generating guitar chords using the Karplus-Strong Algorithm." *MathWorks - MATLAB and Simulink for Technical Computing*. N.p., n.d. Web. 15 Feb. 2012. <http://www.mathworks.com/products/demos/signaltlbx/guitar/filterguitardemo.html>.

"Karplusâ€“Strong string synthesis - Wikipedia, the free encyclopedia." *Wikipedia, the free encyclopedia*. N.p., n.d. Web. 15 Feb. 2012. <http://en.wikipedia.org/wiki/Karplus%E2%80%93Strong_string_synthesis>.

"LV-MaxSonar Â® -EZ." *maxbotix.com*. N.p., n.d. Web. 10 Feb. 2010. <www.maxbotix.com/documents/MB1040_Datasheet.pdf>.

"Low-pass filter - Wikipedia, the free encyclopedia." *Wikipedia, the free encyclopedia*. N.p., n.d. Web. 14 Feb. 2012. <http://en.wikipedia.org/wiki/Low-pass_filter>.

"RF module datasheet." *Sparkfun.com*. Sparkfun.com, n.d. Web. 12 Feb. 2012. <dlnmh9ip6v2uc.cloudfront.net/datasheets/Wireless/General/RWS-371-6_433.92MHz_ASK_RF_Receiver_Module_Data_Sheet.pdf>.

Serafin, Stefania. "Implementing a basic Karplus Strong algorithm." *Implementing a basic Karplus Strong algorithm*. Stanford University, 8 Nov. 2001. Web. 20 Feb. 2012. <https://ccrma.stanford.edu/~serafin/320/assign5/Implementing_basic_Karplus.html>.

"TWS-BS RF MODULE Series." *Sparkfun.com*. Sparkfun.com, n.d. Web. 2 Dec. 2012. <dlnmh9ip6v2uc.cloudfront.net/datasheets/Wireless/General/TWS-BS-3_433.92MHz_ASK_RF_Transmitter_Module_Data_Sheet.pdf>.

"Variable Potentiometer Membrane Strip, 50mm (SoftP." *MicroControllerShop.com - MicroController Development Tools*. N.p., n.d. Web. 20 Feb. 2012. <http://microcontrollershop.com/product_info.php?products_id=3803>.