Air Scratch:
An Innovative *Touch-Free* DJ Controller

Senior Project

California Polytechnic State University, San Luis Obispo
Electrical Engineering Department

Nixon Vandel

6/7/2011

Advisor: Dr. Bryan Mealy
ABSTRACT

This project explores the design and development of a touchless DJ interface. It requires fundamental knowledge of the operation of DJ software and hardware. Two theremins and two microcontrollers are integrated into the design and encompass the analog/digital aspect of the project. The design was created after researching hands free and touchless electronic controllers. The features incorporated into Air Scratch are very common and useful for DJ’s. The principle of a touchless DJ controller was first tested using software, and then later developed into hardware; primarily for copyright issues. The DJ software used by Air Scratch is found online for free, while the hardware was produced after successfully finding compatible technologies. The first stage was a semi-functional software implementation of a single turntable, and consisted solely of a theremin, A/D software, and VST instruments. As the project matured, references for the theremin and A/D controllers were found and then assembled. The parts for the theremins and microcontrollers were ordered online, assembled, tested, and then troubleshot. The components for the theremins were soldered onto copper PC board to save time, while the microcontrollers were fabricated onto etched PC boards. All four parts were then placed into a protective and aesthetic enclosure for the final product. This project explains how to develop a non-tangible object into a marketable product.
## Contents

Abstract .................................................................................................................................................. 2  
Table of Figures and Tables .................................................................................................................. 4  
Introduction ........................................................................................................................................... 6  
Background ........................................................................................................................................... 6  
Design ..................................................................................................................................................... 6  
Controls .................................................................................................................................................. 13  
Development ........................................................................................................................................ 16  
Setup and Tuning .................................................................................................................................. 36  
Performance .......................................................................................................................................... 38  
Future Modifications ............................................................................................................................... 38  
Public Reception .................................................................................................................................... 39  
Challenges ............................................................................................................................................. 39  
Conclusion ............................................................................................................................................ 41  
Cost ......................................................................................................................................................... 41  
Software Used ....................................................................................................................................... 41  
Lab Equipment Used .............................................................................................................................. 41  
Time schedule ....................................................................................................................................... 42  
Bibliography .......................................................................................................................................... 42
TABLE OF FIGURES AND TABLES

Figure 1: Software Based Interface for VSTi Control........................................................................ 7
Figure 2: Pitch to MIDI Conversion in WIDISOFT ........................................................................ 7
Figure 3: Virtual DJ Software ........................................................................................................ 8
Figure 4: An example of Mapping MIDI to Virtual DJ VDJ Script ................................................ 8
Figure 5: Theremin Controlling DJ Software .................................................................................. 9
Figure 6: Microcontroller Based Interface with DJ Software.......................................................... 9
Figure 7: Top View of the A/D Interface Box .................................................................................. 10
Figure 8: Front and Rear View of the A/D Interface Box ............................................................... 10
Figure 9: Top View of the DJ Controller ....................................................................................... 11
Figure 10: Side View of the DJ Controller ..................................................................................... 11
Figure 11: Inside View of the DJ Controller ................................................................................... 12
Figure 12: Total System Overview: Two Theremins, Microcontroller D/A, & Laptop .................... 12
Figure 13: Turntable Controls: Play, Scratch, Pause, Rewind ....................................................... 13
Figure 14: DJ Mixer Enables Turntable Controls On/Off ............................................................... 13
Figure 15: Control Box Protoype ................................................................................................... 14
Figure 16: DJ Track Selection Controls.......................................................................................... 15
Figure 17: Theremin Schematic Diagram ...................................................................................... 16
Figure 18: Voltage Regulator Circuit ............................................................................................. 17
Figure 19: Pitch Reference Oscillator ............................................................................................ 17
Figure 20: Pitch Variable Oscillator Circuit ................................................................................... 18
Figure 21: Mixer ............................................................................................................................... 18
Figure 22: Volume Control Circuit ................................................................................................ 19
Figure 23: Pre-amplifier Circuit ..................................................................................................... 19
Figure 24: Theremin 1 (T1) ............................................................................................................. 20
Figure 25: Output Waveform of the Theremin .............................................................................. 20
Figure 26: Two of the Six of T2 circuits .......................................................................................... 21
Figure 27: Testing the Fully Assembled T2 Circuit Board .............................................................. 21
Figure 28: Theremin 1 and Theremin 2 Boxes .............................................................................. 22
Figure 29: The Wiring Diagram for the Microcontroller ................................................................. 23
Figure 30: Original CAD Trace ...................................................................................................... 24
Figure 31: Modified cad trace ........................................................................................................ 24
Figure 32: Laminate transparency, and cad drawing ...................................................................... 25
Figure 33: The ABS plastic box for the A/D microcontrollers ......................................................... 25
Figure 34: Copper Etching Machine ............................................................................................... 26
Figure 35: PC Board: Trimming Loose Wires ................................................................................ 26
Figure 36: The Final MCU Board .................................................................................................. 27
Figure 37: Frequency to Midi Conversion ...................................................................................... 30
Figure 38: Frequency-MidI Note # ............................................................................................... 30
Figure 39: Original MIDI–USB interface ....................................................................................... 31
Figure 40: Modified MIDI-USB interface ...................................................................................... 31
Figure 41: Wiring Diagram for MIDI-USB Interface ...................................................................... 32
Figure 42: Final Enclosure for A/D Box ......................................................................................... 32
Figure 43: Box Construction ......................................................................................................... 33
Figure 44: Final Design ................................................................................................................ 34
Figure 45: Turnable and Mixer Design .......................................................................................... 34
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>46</td>
<td>Air Scratch Logo</td>
<td>35</td>
</tr>
<tr>
<td>47</td>
<td>Left Hand Location - Function Data Map</td>
<td>36</td>
</tr>
<tr>
<td>48</td>
<td>Virtual DJ Midi Scripts</td>
<td>37</td>
</tr>
<tr>
<td>49</td>
<td>Virtual DJ Functions</td>
<td>37</td>
</tr>
<tr>
<td>50</td>
<td>Music Video Mixing</td>
<td>38</td>
</tr>
<tr>
<td>51</td>
<td>Public Demonstration</td>
<td>39</td>
</tr>
<tr>
<td>52</td>
<td>Happy Engineer</td>
<td>40</td>
</tr>
</tbody>
</table>
INTRODUCTION

As DJ technology advances beyond the use of the traditional turntable, demands set by the DJ community require simpler and more intuitive control surfaces. Similar to a turntable, the Air Scratch instrument will accurately allow users to scratch sound samples in the air using hand gestures; just like the “Air Guitar” that is popular amongst guitar enthusiasts. However, unlike a turntable, Air Scratch does not require expensive equipment and is more forgiving of the user’s incorrect hand gestures. Another great feature of Air Scratch is that the user can choose to mix mp3 audio or mp4 music videos- DJ or VJ.

BACKGROUND

The “scratch” is a combination of two hand gestures. 1- moving a record 2- moving a fader. Using these two hand gestures simultaneously results in an almost infinite combination of sounds. To truly be a DJ the user must be able to play, pause, scratch, rewind, and stop. The use of these functions lets the user decides to scratch or mix.

DESIGN

Air Scratch consists of:

2 Theremins
2 Arduino Microcontrollers (Atmega 328)
1 Virtual DJ Software
1 RCA audio cable
2 USB cables
2 Midi-USB Adapters

Initially, a theremin is used to control VST instruments on a laptop by way of Virtual Studio Technology instruments. VSTi’s are software based virtual instruments that offer a wide array of computer generated sound effects. The Theremin outputs an audio signal; more specifically a sine wave. The generated sine wave frequency is used to determine the pitch by means of laptop software, which converts the pitch to MIDI notes. The MIDI notes are then used to control VST instruments on the laptop to create polyphonic sounds and play audio samples. Figure 1 shows the block diagram of the theremin- VSTi setup.
The theremin is connected to a PC by using a 3.5mm mini-jack to mini-jack connector. Afterwards, WIDISoft Pitch to MIDI software is used to convert the frequency to MIDI notes. WIDISoft also allows the user to select several instruments to play the theremin with. The Steel Drums are chosen in Figure 2 however there are approximately 50 instruments to control.

The next step involves substituting the VSTi with DJ software; this will allow the user to control music. The Virtual DJ GUI is shown in Figure 3.
After the pitch is converted into MIDI, the VDJScript language in the Virtual DJ Software allows the user to map the MIDI functions into the DJ Software to perform the “Scratch” function. The GUI for VDJ is shown in Figure 3 and an example of the VDJScript is shown in Figure 4.
Following a proof of concept that a theremin can control DJ software, a hardware solution is devised. Figure 6 shows the proposed concept of the hardware implementation.

An Atmega 328 microcontroller will be used to interface the theremin Pitch and Volume to the laptop. The Atmega 328 detects the frequency of the input signal and uses a look up table to output the appropriate MIDI note. Once in MIDI form, a USB-to-MIDI box connects to the laptop and allows the MIDI notes to become visible to Microsoft Windows.

A very important rule when using two theremins on the same MIDI bus is that the two theremins must always have non-overlapping MIDI note ranges. This essentially means:

$$T_1 \text{MIDI Notes} \neq T_2 \text{MIDI Notes}$$

The Arduino MCU code will have to be modified to shift T1 out of the range of T2.
Next is the A/D box design. Figure 7 shows the microcontroller enclosure. The analog and digital connections are planned out before any parts are ordered.

**FIGURE 7: TOP VIEW OF THE A/D INTERFACE BOX**

**FIGURE 8: FRONT AND REAR VIEW OF THE A/D INTERFACE BOX**
The plan is to hide the theremins inside a wooden box to look like DJ turntables and mixer. This will immerse the user in the experience. Figure 9 shows the proposed design.

**FIGURE 9: TOP VIEW OF THE DJ CONTROLLER**

Figure 10 and Figure 11 show the inside view of the box.

**FIGURE 10: SIDE VIEW OF THE DJ CONTROLLER**
FIGURE 11: INSIDE VIEW OF THE DJ CONTROLLER

FIGURE 12: TOTAL SYSTEM OVERVIEW: TWO THEREMINS, MICROCONTROLLER D/A, & LAPTOP
CONTROLS

The manipulation of records will be controlled by the location of the user's hand. A MIDI note associated command is triggered in the DJ software whenever the user's hand crosses a red control line. The controls are mapped and placed onto the turntables according to the location of where the MIDI note is ON. The markings in Figure 13 correspond to where MIDI notes are turned ON; thus it lets the users know where to place their hands to play, pause, scratch, and rewind.

![Turntable Controls: Play, Scratch, Pause, Rewind](image)

FIGURE 13: TURNTABLE CONTROLS: PLAY, SCRATCH, PAUSE, REWIND

The DJ mixer is designed to turn the turntable controls on and off. This gives the user the ability to pause, play records. *Which ultimately allows them to mix and scratch;* this little insight is vital to the success of Air Scratch. The initial design for the mixer controls are shown in Figure 14.
The final product skins are applied to the proto box in Figure 15 to get a general idea of the appearance. The finished DJ box wiring will be tucked inside the box, and the turntable skins will be printed on colored sticker material. The finished box will be made out of wood to hide the theremins and then the final controls will be tuned and marked on the box.
Track selection is accomplished with the use of a PC keyboard. This allows the user to easily choose songs and load songs into the left /right deck in Virtual DJ. The modified controls are in Figure 16.

![Figure 16: DJ Track Selection Controls](image)
DEVELOPMENT

The first phase in the development is to understand how the theremin works. A theremin is basically made of seven separate circuits working as a whole system. An analysis of the theremin circuit will allow for a better understanding of just how Air Scratch works. The wiring diagram for the theremin is shown in Figure 17.

Table 1 shows the location of the circuit and its function.

<table>
<thead>
<tr>
<th>Letter</th>
<th>Circuit Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Pitch Reference Oscillator</td>
</tr>
<tr>
<td>B</td>
<td>Mixer</td>
</tr>
<tr>
<td>C</td>
<td>Pitch Variable Oscillator</td>
</tr>
<tr>
<td>D</td>
<td>Volume Control Reference Oscillator</td>
</tr>
<tr>
<td>E</td>
<td>Volume Control Variable Oscillator</td>
</tr>
<tr>
<td>F</td>
<td>Pre-amplifier</td>
</tr>
<tr>
<td>G</td>
<td>Voltage Regulator</td>
</tr>
</tbody>
</table>

TABLE 1: THEREMIN CIRCUITS
The first circuit to analyze is the voltage regulator. Power to theremin is supplied by an 8 pin adjustable output voltage regulator, National Semiconductor LP2951ACN. Input voltage is 9V DC from a battery and outputs 7.5VDC rail voltage throughout the theremin. The voltage regulator is shown in Figure 18.

The Theremin consists of four oscillators. The first two oscillators are the Pitch reference and Pitch Variable Oscillators. Figure 19 shows the Pitch Reference Oscillator.

The Pitch Reference Oscillator is assembled and connected to an O-scpe at connection A. The frequency of oscillation is measured to be \( f_0 = 357.143 \text{KHz} \) with a period of \( T=2.8 \mu\text{S} \) and amplitude 4.8V p-p.

It is important to note that the Pitch Reference Oscillator is a constant frequency- unlike the Pitch Variable Oscillator which varies in frequency.
The next stage is the Pitch Variable Oscillator. As the user’s hand is waved over the pitch reference antenna it varies the capacitance in a variable oscillator circuit. Changing the capacitance changes the frequency of oscillation. With no hand present, the measured frequency of oscillation is recorded as $f_0 = 357.160$ KHz with a period of $T = 2.8\mu$S and amplitude $4.5V_{p-p}$. The Pitch Variable Oscillator is shown in Figure 20.

![Figure 20: Pitch Variable Oscillator Circuit](image)

It is observed that the frequency of oscillation for the Pitch Reference and the Variable Oscillator circuits are fairly close together. This is important because the difference of the two frequencies cannot exceed any more than 20KHz. Reason being - the two signals are subtracted in a mixer and the difference between the two must not exceed the audible range of what human hearing is ~ 20Hz-20KHz. The signals from the Pitch Reference and Pitch Variable oscillators enter the heterodyning Mixer circuit shown in Figure 21.

![Figure 21: Mixer](image)
The mixer down converts the difference of the two frequencies. The Pitch Reference enters at Point A and the Variable signal enters into Point B. The output is taken off the Low-Pass circuit at Point C. A sinusoid is measured at Point C with varying frequency and amplitude. At 50Hz the amplitude was measured to be 80mVp-p and at 5KHz the amplitude is 40mVp-p. The Low-Pass filter is important because it prevents higher harmonics from being passed on.

The third and fourth oscillators are in the Volume Control circuit. The second theremin antenna controls the switching frequency of a FET, which in turn acts like a Pulse Width Modulator; this determines the amplitude of the voltage for the volume control. Figure 22 shows the Volume Control circuit.

![Volume Control Circuit](image)

**FIGURE 22: VOLUME CONTROL CIRCUIT**

The output of the Volume Control circuit switches a FET on and off, which controls the amplitude of the audio signal that was generated from the mixer. The control signal and the audio signal are "mixed" by the FET and sent to the pre-amp in Figure 23 which boosts the signal to the necessary line-level value of 700mV.

![Pre-Amplifier Circuit](image)

**FIGURE 23: PRE-AMPLIFIER CIRCUIT**
Since the Air Scratch system uses two turntables, two theremins are made— one for each turntable. The first theremin (T1) is assembled and shown in Figure 24.

An oscilloscope is connected to the line output of the T1 pre-amplifier and the signal is captured. Figure 25 shows the sinusoidal output of Theremin 1. The output is approximately 200mv p-p.
The process for assembling a theremin from the wiring diagram is as follows. First, each individual (of the six) circuit is soldered onto small copper boards. Figure 26 shows the wiring diagrams and the fully assembled circuit.

![Figure 26: Two of the six of T2 circuits](image1)

After all six of the small circuits are soldered, they are joined together on a larger copper board. Figure 27 shows the complete board with all six small circuits soldered on it.

![Figure 27: Testing the fully assembled T2 circuit board](image2)
Similar to the first theremin, the circuit is put in a box for protection. Theremin 1 controls the left deck and theremin 2 controls the right deck. Now, the user can decide to scratch or mix. Figure 28 shows the boxes for the theremins.

Next to build is the microcontroller. Doing some research online reveals that a hardware implementation of the software Pitch-MIDI is possible. The boards that are chosen are Arduino replica’s with an AVR Atmega 328 microcontroller. The Atmega 328 is chosen over the Atmega 128 because of its large memory size. The wiring diagram for the hardware implementation is shown in on the next page in Figure 29.
FIGURE 29: THE WIRING DIAGRAM FOR THE MICROCONTROLLER
The EAGLE CAD drawing for the MCU design is obtained and modified in Photoshop so that the traces are etchable with the resolution of the Cal Poly IME 156 Lab etching machines. The original and modified CAD drawings are shown in Figure 30 and Figure 31.
The CAD traces are then transferred to a laminate. All three are shown in Figure 32.

A box for the microcontrollers (MCU) is ordered to guarantee protection. A paper circuit is cut out to make sure that the final PC boards will fit. The proposed design is shown in Figure 33.
Next the transparency is copied over to a copper board and then etched to reveal the traces. This step is shown in Figure 34.

![Figure 34: Copper Etching Machine](image)

About 100 holes are drilled for the components. Next is to solder the parts onto the board. Extreme care should be taken to make sure conductivity is traced throughout the board, so whenever a new part is soldered, a conductivity test should be made. This little step of attention will help to save time because there could be a few IC pins that are not properly soldered. Soldering the IC pins require extra care because they are so tiny and spaced close together. Parts are ordered and then soldered onto the PC Board. This is shown in Figure 35.

![Figure 35: PC Board: Trimming Loose Wires](image)
The first MCU board is placed into its protective case. It is trimmed to fit securely as shown in Figure 36.

FIGURE 36: THE FINAL MCU BOARD

The code for MCU finds the input frequency and generates a MIDI ON note. The code had to be modified for one of the MCU's to put the theremins on separate MIDI channels. This is important, because it allows the theremins to operate in the same frequency without controlling one another. This little modification is essential to get around

\[
\text{Theremin}_{1\text{MIDI Notes}} \neq \text{Theremin}_{2\text{MIDI Notes}}
\]

One of the theremins is put on MIDI channel 1 and the other theremin is on MIDI channel 3. This eventually solves the problem of having conflicting MIDI commands. Having conflicting commands makes turntable 1 control turntable 2 in Virtual DJ (Which is undesired). Once again, it essential to separate the theremins so that they don’t operate the opposite turntable in the Virtual DJ software. The portion of the code that converts frequency to MIDI and outputs it on MIDI channel 3 is shown in Figure 37.
uint16_t lower_bound_period[] = {
    /*22282,*/ 21052, 19869, 18748, 17696, 16706, 15768, 14880, 14046, 13259,
    12515, 11814, 11152, 10525, 9931, 9374, 8849, 8351, 7882, 7442,
    7024, 6631, 6260, 5908, 5577, 5264, 4968, 4689, 4426, 4177,
    3943, 3722, 3513, 3316, 3130, 2955, 2789, 2632, 2485, 2346,
    2214, 2090, 1972, 1862, 1757, 1658, 1565, 1478, 1395, 1317,
    1244, 1174, 1107, 1045, 987, 932, 880, 830,
    784, 740,
    699, 659, 622, 588, 555, 524, 495, 467, 441, 421
};

int detune_values[]  = {
    /*123,*/ 118, 112, 105, 99, 94, 89, 83, 79, 75,
    70, 66, 63, 60, 56, 53, 50, 47, 44, 42,
    39, 37, 35, 33, 31, 30, 28, 26, 25, 24,
    22, 21, 20, 19, 18, 17, 16, 15, 14, 13,
    13, 12, 11, 11, 10, 9, 9, 8, 8, 7,
    7, 7, 6, 6, 6, 5, 5, 5, 4, 4,
    4, 4, 4, 3, 3, 3, 3, 3
};

prog_char string_1[] PROGMEM   = "F#-3";
prog_char string_2[] PROGMEM   = "G-3";
prog_char string_3[] PROGMEM   = "Ab-3";
prog_char string_4[] PROGMEM   = "A-3";
prog_char string_5[] PROGMEM   = "Bb-3";
prog_char string_6[] PROGMEM   = "B-3";
prog_char string_7[] PROGMEM   = "C-2";
prog_char string_8[] PROGMEM   = "C#-2";
prog_char string_9[] PROGMEM   = "D-2";
prog_char string_10[] PROGMEM  = "Eb-2";
prog_char string_11[] PROGMEM  = "E-2";
prog_char string_12[] PROGMEM  = "F-2";
prog_char string_13[] PROGMEM  = "F#-2";
prog_char string_14[] PROGMEM  = "G-2";
prog_char string_15[] PROGMEM  = "Ab-2";
prog_char string_16[] PROGMEM  = "A-2";
prog_char string_17[] PROGMEM  = "Bb-2";
prog_char string_18[] PROGMEM  = "B-2";
prog_char string_19[] PROGMEM  = "C-1";
prog_char string_20[] PROGMEM  = "C#-1";
prog_char string_21[] PROGMEM  = "D-1";
prog_char string_22[] PROGMEM  = "Eb-1";
prog_char string_23[] PROGMEM  = "E-1";
prog_char string_24[] PROGMEM  = "F-1";
prog_char string_25[] PROGMEM  = "F#-1";
prog_char string_26[] PROGMEM  = "G-1";
prog_char string_27[] PROGMEM  = "Ab-1";
prog_char string_28[] PROGMEM  = "A-1";
prog_char string_29[] PROGMEM  = "Bb-1";
prog_char string_30[] PROGMEM  = "B-1";
prog_char string_31[] PROGMEM  = "C 0";
prog_char string_32[] PROGMEM = "C# 0";
prog_char string_33[] PROGMEM = "D  0";
prog_char string_34[] PROGMEM = "Eb 0";
prog_char string_35[] PROGMEM = "E  0";
prog_char string_36[] PROGMEM = "F  0";

prog_char string_37[] PROGMEM = "F# 0";
prog_char string_38[] PROGMEM = "G  0";
prog_char string_39[] PROGMEM = "Ab 0";
prog_char string_40[] PROGMEM = "A  0";
prog_char string_41[] PROGMEM = "Bb 0";
prog_char string_42[] PROGMEM = "B  0";
prog_char string_43[] PROGMEM = "C  1";
prog_char string_44[] PROGMEM = "C# 1";
prog_char string_45[] PROGMEM = "D  1";
prog_char string_46[] PROGMEM = "Eb 1";
prog_char string_47[] PROGMEM = "E  1";
prog_char string_48[] PROGMEM = "F  1";

prog_char string_49[] PROGMEM = "F# 1";
prog_char string_50[] PROGMEM = "G  1";
prog_char string_51[] PROGMEM = "Ab 1";
prog_char string_52[] PROGMEM = "A  1";
prog_char string_53[] PROGMEM = "Bb 1";
prog_char string_54[] PROGMEM = "B  1";
prog_char string_55[] PROGMEM = "C  2";
prog_char string_56[] PROGMEM = "C# 2";
prog_char string_57[] PROGMEM = "D  2";
prog_char string_58[] PROGMEM = "Eb 2";
prog_char string_59[] PROGMEM = "E  2";
prog_char string_60[] PROGMEM = "F  2";

prog_char string_61[] PROGMEM = "F# 2";
prog_char string_62[] PROGMEM = "G  2";
prog_char string_63[] PROGMEM = "Ab 2";
prog_char string_64[] PROGMEM = "A  2";
prog_char string_65[] PROGMEM = "Bb 2";
prog_char string_66[] PROGMEM = "B  2";
prog_char string_67[] PROGMEM = "C  3";
prog_char string_68[] PROGMEM = "Eb 3";
prog_char string_69[] PROGMEM = "D  3";

PROGMEM const char *note_name_table[] =
{
  string_1, string_2, string_3, string_4, string_5, string_6,
  string_7, string_8, string_9, string_10, string_11, string_12,

  string_13, string_14, string_15, string_16, string_17, string_18,
  string_19, string_20, string_21, string_22, string_23, string_24,

  string_25, string_26, string_27, string_28, string_29, string_30,
  string_31, string_32, string_33, string_34, string_35, string_36,

  string_37, string_38, string_39, string_40, string_41, string_42,
  string_43, string_44, string_45, string_46, string_47, string_48,
For comparison, the actual Frequency-MIDI-# conversion reference chart is shown in Figure 38.

<table>
<thead>
<tr>
<th>MIDI number</th>
<th>Note name</th>
<th>Frequency Hz</th>
<th>Period ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>A0</td>
<td>27.503</td>
<td>36.36</td>
</tr>
<tr>
<td>23</td>
<td>B0</td>
<td>30.869</td>
<td>30.58</td>
</tr>
<tr>
<td>24</td>
<td>C1</td>
<td>32.703</td>
<td>29.155</td>
</tr>
<tr>
<td>25</td>
<td>D1</td>
<td>36.706</td>
<td>30.56</td>
</tr>
<tr>
<td>26</td>
<td>E1</td>
<td>41.205</td>
<td>27.34</td>
</tr>
<tr>
<td>27</td>
<td>F1</td>
<td>45.954</td>
<td>28.86</td>
</tr>
<tr>
<td>28</td>
<td>G1</td>
<td>49.999</td>
<td>25.71</td>
</tr>
<tr>
<td>29</td>
<td>A1</td>
<td>55.000</td>
<td>22.91</td>
</tr>
<tr>
<td>30</td>
<td>B1</td>
<td>61.735</td>
<td>21.26</td>
</tr>
<tr>
<td>31</td>
<td>C2</td>
<td>69.296</td>
<td>19.26</td>
</tr>
<tr>
<td>32</td>
<td>D2</td>
<td>73.445</td>
<td>17.16</td>
</tr>
<tr>
<td>33</td>
<td>E2</td>
<td>77.792</td>
<td>15.29</td>
</tr>
<tr>
<td>34</td>
<td>F2</td>
<td>82.407</td>
<td>13.86</td>
</tr>
<tr>
<td>35</td>
<td>G2</td>
<td>87.307</td>
<td>11.45</td>
</tr>
<tr>
<td>36</td>
<td>A2</td>
<td>92.499</td>
<td>10.18</td>
</tr>
<tr>
<td>37</td>
<td>B2</td>
<td>103.85</td>
<td>9.631</td>
</tr>
<tr>
<td>38</td>
<td>C3</td>
<td>116.54</td>
<td>8.981</td>
</tr>
<tr>
<td>39</td>
<td>D3</td>
<td>133.64</td>
<td>8.642</td>
</tr>
<tr>
<td>40</td>
<td>E3</td>
<td>159.57</td>
<td>8.161</td>
</tr>
<tr>
<td>41</td>
<td>F3</td>
<td>185.00</td>
<td>7.717</td>
</tr>
<tr>
<td>42</td>
<td>G3</td>
<td>207.65</td>
<td>7.405</td>
</tr>
<tr>
<td>43</td>
<td>A3</td>
<td>233.08</td>
<td>7.190</td>
</tr>
<tr>
<td>44</td>
<td>B3</td>
<td>251.63</td>
<td>6.909</td>
</tr>
<tr>
<td>45</td>
<td>C4</td>
<td>277.18</td>
<td>6.625</td>
</tr>
<tr>
<td>46</td>
<td>D4</td>
<td>311.13</td>
<td>6.304</td>
</tr>
<tr>
<td>47</td>
<td>E4</td>
<td>349.23</td>
<td>6.214</td>
</tr>
<tr>
<td>48</td>
<td>F4</td>
<td>392.00</td>
<td>6.025</td>
</tr>
<tr>
<td>49</td>
<td>G4</td>
<td>435.95</td>
<td>5.857</td>
</tr>
<tr>
<td>50</td>
<td>A4</td>
<td>480.00</td>
<td>5.703</td>
</tr>
<tr>
<td>51</td>
<td>B4</td>
<td>522.95</td>
<td>5.565</td>
</tr>
<tr>
<td>52</td>
<td>C5</td>
<td>574.88</td>
<td>5.432</td>
</tr>
<tr>
<td>53</td>
<td>D5</td>
<td>627.35</td>
<td>5.315</td>
</tr>
<tr>
<td>54</td>
<td>E5</td>
<td>681.46</td>
<td>5.207</td>
</tr>
<tr>
<td>55</td>
<td>F5</td>
<td>734.99</td>
<td>5.111</td>
</tr>
<tr>
<td>56</td>
<td>G5</td>
<td>788.39</td>
<td>5.024</td>
</tr>
<tr>
<td>57</td>
<td>A5</td>
<td>842.29</td>
<td>4.945</td>
</tr>
<tr>
<td>58</td>
<td>B5</td>
<td>905.77</td>
<td>4.870</td>
</tr>
<tr>
<td>59</td>
<td>C6</td>
<td>971.25</td>
<td>4.800</td>
</tr>
<tr>
<td>60</td>
<td>D6</td>
<td>1038.56</td>
<td>4.737</td>
</tr>
<tr>
<td>61</td>
<td>E6</td>
<td>1109.01</td>
<td>4.678</td>
</tr>
<tr>
<td>62</td>
<td>F6</td>
<td>1183.37</td>
<td>4.620</td>
</tr>
<tr>
<td>63</td>
<td>G6</td>
<td>1260.26</td>
<td>4.564</td>
</tr>
<tr>
<td>64</td>
<td>A6</td>
<td>1336.53</td>
<td>4.508</td>
</tr>
<tr>
<td>65</td>
<td>B6</td>
<td>1413.46</td>
<td>4.453</td>
</tr>
<tr>
<td>66</td>
<td>C7</td>
<td>1492.00</td>
<td>4.401</td>
</tr>
<tr>
<td>67</td>
<td>D7</td>
<td>1572.00</td>
<td>4.350</td>
</tr>
<tr>
<td>68</td>
<td>E7</td>
<td>1653.00</td>
<td>4.301</td>
</tr>
<tr>
<td>69</td>
<td>F7</td>
<td>1734.00</td>
<td>4.252</td>
</tr>
<tr>
<td>70</td>
<td>G7</td>
<td>1816.00</td>
<td>4.205</td>
</tr>
<tr>
<td>71</td>
<td>A7</td>
<td>1899.00</td>
<td>4.160</td>
</tr>
<tr>
<td>72</td>
<td>B7</td>
<td>1982.00</td>
<td>4.115</td>
</tr>
<tr>
<td>73</td>
<td>C8</td>
<td>2066.00</td>
<td>4.071</td>
</tr>
</tbody>
</table>
Once the MIDI # command is outputted of the MCU it is registered in Microsoft Windows through a MIDI-USB adapter. The adapter is purchased online through Amazon.com and retails for $5.

Since the MCU can only output MIDI and cannot talk to Windows, and to avoid writing custom drivers, a MIDI-USB device is modified. The MIDI-USB converter is modified by means of a 220 Ω resistor to provide +5V connectivity from the Arduino MCU to a Windows PC. The hacked device passes the MIDI note commands to Windows and will also provide +5V power to the MCU. Design requirements dictate the use of only one USB connection- providing power and data on the same cable.

A 220 Ω resistor was added to provide a small load and the +5V line connects to the Arduino +5V in. The -Vin connects to TX, and the G goes to GND. Figure 39, Figure 40, and Figure 41 show the adapter.

FIGURE 39: ORIGINAL MIDI-USB INTERFACE

FIGURE 40: MODIFIED MIDI-USB INTERFACE
The next step is to house the microcontrollers and the MIDI-USB adapters in a safe enclosure. Figure 42 shows the final clear ABS plastic case with the RCA connectors.
The wooden box that houses the theremins is shown in Figure 43. It was made out of an old Ping-Pong table.
Next, the box is spray painted black and control graphics are printed onto the surface. Then theremins are secured into the box using angle irons. Using only one angle iron is important because it allows for the theremin to be adjusted (rotated) and returned backing to the correct location inside the box.

Analog RCA connections are installed inside the box provide connectivity to the A/D box. RCA are chosen because they are the industry standard connection.

The final box design is shown in Figure 44 and Figure 45.
The final step was to come up with a catchy name.

“Air Scratch” best describes the features of the product, so it is chosen.

A logo is designed and shown in Figure 46.
SETUP AND TUNING

The Air Scratch system is susceptible to radio interference. To counter this obstacle a simple setup and tuning procedure is put into place.

First, both theremin frequencies must be set to 1.864 Khz (MIDI # 94) by means of a variable capacitor on the variable oscillator circuits (Figure 20). This only needs to be done once upon purchase, and then never touched again.

Step two involves tuning Air Scratch to the room interference. This is accomplished by means of MIDI-Ox software. The process is as follows:

A. Download and install MIDI-Ox at [www.MIDI-Ox.com](http://www.MIDI-Ox.com)
B. Monitor the PC input/output
C. Create a data map

The MIDI-Ox data map is used to easily map ranges of MIDI #'s into one number. MIDI #'s range from 0-127, the # 94 is chosen for convenience because it’s easily audible.

Table 2: Data Map shows the key to creating the correct MIDI-Ox data map.

<table>
<thead>
<tr>
<th>Function</th>
<th>Pause</th>
<th>Scratch</th>
<th>Play</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIDI Input # Range</td>
<td>0-87</td>
<td>88-93</td>
<td>94-127</td>
</tr>
<tr>
<td>MIDI-Ox Output #</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

**TABLE 2: DATA MAP**

Figure 47 shows the function for the location of the user’s left hand when crossing the red lines.
The MIDI # outputs (1-3) are then linked to Virtual DJ Play, Pause, and Scratch function using VDJ Script language. The VDJ scripts are shown in Table 3 and in Figure 48.

<table>
<thead>
<tr>
<th>MIDI-Ox Output #</th>
<th>VDJ Script</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>pause</td>
</tr>
<tr>
<td>2</td>
<td>scratch-125ms</td>
</tr>
<tr>
<td>3</td>
<td>play_sync</td>
</tr>
</tbody>
</table>

**TABLE 3: MIDI-OX TO VIRTUAL DJ MAP**

**FIGURE 48: VIRTUAL DJ MIDI SCRIPTS**

Functions for the Virtual DJ Scripts are shown in Figure 49.

- **deck**: specifies which deck the verb act upon by adding "deck xxx" in front of the verb. xxx can be "1", "2", "left", "right", "default" or "active". ("1"/"left" and "2"/"right" are synonyms).
- **scratch**: 'scratch +120ms' to scratch 120ms forward.
- **play_sync**: play the song instantly synchronized with the other deck
- **pause**: pause the deck.

**FIGURE 49: VIRTUAL DJ FUNCTIONS**
PERFORMANCE

Air Scratch proves to incredibly accurate and the feel of the virtual records is incredible precise.

The final feature for Air Scratch was to add the ability to mix videos with hand movement. This is demonstrated in Figure 50.

![Figure 50: Music Video Mixing](image)

FUTURE MODIFICATIONS

Track selection:

A way to select tracks without using the laptop would be ideal. Currently, all DJ controllers use the laptop to select tracks; it would be nice to do it hands free.

Crossfader:

An adjustable crossfader curve will blend the amplitude of the signals.

Equalizer:

An Air FX equalizer will be built into the box module.

1-1 Hand Mapping:

More code will be developed to provide a more accurate, real-time mapping of hand gestures to record control.
PUBLIC RECEPTION

Air Scratch was openly demonstrated to the public at the Cal Poly Senior Project Expo on June 2\textsuperscript{nd} 2011. Public reaction was amazingly warm and enthusiastic. Attendees were fascinated with the system’s ease, accuracy, design, and technology. Figure 51 is a picture of the public’s reception.

![Public Demonstration](image)

FIGURE 51: PUBLIC DEMONSTRATION

CHALLENGES

Initially, the system could only be used for scratching. The biggest challenge occurred because a DJ has to Scratch and Mix. Since Mixing requires to turntables, and only one Theremin was originally present, it wasn’t possible to be a fully functional DJ “Scratch and Mix” system. At first, the system could only be used for scratching and not mixing.

Deciding on DJ software was also difficult. There are many available including; PC DJ, Virtual DJ, Serato Scratch, Traktor Studio, and MIXX. Virtual DJ was chosen based upon its GUI, scripting language, and its ability to mix music videos.

A lot of time was spent contemplating control schemes, software, and hardware. The most difficult part was deciding on the software/language to use to process the MIDI logic statements. Some of the software I came across includes; Processing 2.0, MIDI OX w/ Visual Basic, VDJ Script, MIXX w/ C++, and Java w/ MIDI Bus plugin. Due to the limited resources available for each method it was decided to go with Virtual DJ Script.
There was no guarantee that the MCU’s will perform the required tasks and it is nerve wrecking to say the least. A fair amount of time was spent researching the hardware and even more time was spent trying to keep cost down. The uncertainty involved in the MCU’s made ordering parts daunting task because price is a large factor of the project.

It took a total of 4 days to make the second theremin (T2) compared to the 6 weeks it took to make the first theremin (T1). Soldering has become almost second nature and all 6 of T2 circuits were soldered within 5 hours. Constructing the box, drilling holes, and finding antennas at the local dump took a day all together. The last two days were spent troubleshooting and polishing.

Troubleshooting the second theremin (T2) proved to be more difficult than initially thought. One day was spent troubleshooting, disassembling, and reassembling one mixer circuit alone, while the second day was dedicated to wiring the +7.5V rails, tracing signals, checking schematics, and re-soldering loose connections. The biggest setback came from the pre-amp circuit- a lose 220 ohm resistor that wasn’t properly soldered to ground, this one resistor alone cost 5 hours of troubleshooting time. On the fourth and final day T2 was wired up and put in its box, then tuned to performance specs. It nearly outperforms T1. Figure 52 shows the author celebrating.

Etching the PC board required a lot of work. Proffesor Rinzels of the Industrial Manufacturing Engineering department and my friend Louis Goughli helped out considerably with the board. It was decided that the traces on the PC board were too close to the IC pins, so I had to use Photoshop to revamp the circuit and make 7 custom jumpers to replace the faulty traces. After that it took about a day to etch the pc board in the IME 156 lab. The most difficult part was trying to get the right exposure on the transparency so that the traces don’t bleed together.

After the PC board was hand etched and fabricated parts were soldered on, troubleshot, and code was loaded. It took about 2 days of continuous work for the board to be completed.
CONCLUSION

The Air Scratch project allowed me to learn a lot about analog and digital electronic instruments. Different stages of research and development were phased through, and the product life cycle was experienced. It was rewarding to see an application of my engineering education come to life.

COST

Theremin – $90
Microcontroller - $60 each

**Total System – $300**

SOFTWARE USED

Virtual DJ 6.0
WidiSoft Pitch to Midi
Fruity Loops Studio 9
Midi-Ox

LAB EQUIPMENT USED

Dual DC Power supply
Oscilloscope
Digital Multi-meter
Grabber Wires
Breadboard
Soldering Iron
Solder
Drill
TIME SCHEDULE

Research: 5 Hours

Design: 5 Hours

Lab Testing and Troubleshooting: 25 Hours

Layout Design and Ordering: 10 Hours

Enclosure Design and Drilling: 8 Hours

Soldering and Mounting: 30 Hours

Testing Final Product: 5 Hours

BIBLIOGRAPHY


