Department of Electrical & Computer Engineering
University of Central Florida

Electronic Music Interaction:
Modern and alternative interactions with music devices

Group #1:
Anthony Adu[1]
Kenzo Mendoza[2]
Thomas Spalding[3]
Florence Trinh[4]

[1] amtee12@knights.ucf.edu
[2] kenzo_mendoza@knights.ucf.edu
[3] spalding@knights.ucf.edu
[4] ftrinh@knights.ucf.edu

TABLE OF CONTENTS

FIGURES

SECTION 1.0 MOTIVATION & DISCUSSION

Examining the use of ephemeral interaction with technology, our senior design group decided to tie together the theremin, a musical instrument that can be played without touch, to create a technological interaction that took physical bodies as input via the ephemeral, and provide a highly customizable interaction environment. To tie together this highly adaptable environment we lit upon Hochenbaum's case study on the Chronome[1] that notes to provide a virtuosic experience without the years of mastery, one must find a balance between the complexity of an instrument and ease of use. Having found this highly pertinent to our endeavor we decided to use this as the hub of interaction between the MIDI-capable theremin and any other MIDI-capable performance instrument. We set to make each device modular, to expand upon this customizability, in that they can be placed together in many number of combinations to reflect one's personal mental organization. By creating the Expandome we wanted to even further push the easy-grid instrument to the next level. The theremin, as an infinitely variable instrument, that requires years of practice as well as a near perfect pitch ear to master, is an ideal candidate for our intermediary device, a pitch-to-MIDI converter. This device can take any analog instrument and allow pitch-mapping, and arpeggiated control, as well as direct feedback for practice. This complements our theme of ease-in-interaction in proficient playing, as well as gives any analog instrument social and adaptable input to our performance device hub.

SECTION 2.0 OBJECTIVES

Our final goal is to create a virtuosic experience through high customizability, modularity, and simplicity. We shall accomplish this by understanding and developing the following devices: a theremin, a pitch-to-MIDI converter, and at least three Expandome.

Another set of goals revolves around a theme of learning through direct feedback. A virtual instrument interface will be created that provides the user with all the tools to directly control the devices, as well as provide real-time data on the operation of the units. For example, the theremin will be supplemented with a pitch training subprogram. It will provide a novice thereminist immediate pitch and note display on a concert pitch scale.

SECTION 3.0 DEFINITION

3.1 uWave Theremin

Patented in 1928[2], the theremin was the first electronic musical instrument. Although now a classic example, it has always been the most drastic break from any traditional instrument before or since. The theremin is played by moving both hands in the air in the vicinity of two antennas. The player’s hands creates a ‘hand capacitance’ between the antenna and true ground through one’s hand, acting as a plate, and the antenna as the other plate. Typically, the left hand controls the volume, and the right hand the pitch. For the purpose of this project we will call our implementation of this device the uWave

Theremin. The theremin requires an exquisite ear and complete control of one's motion; this brings us to our second device, a pitch-to-MIDI converter.

3.2 SenseBox - A Pitch-to-MIDI Converter

This device takes an incoming audio signal and converts it to MIDI data messages. The audio signal is analyzed by comparing its frequency to a lookup table of musical scale note pitches. The incoming audio signal can be directly pitch-matched in the outgoing MIDI data or used as a controller via data messages. By allowing the infinitely fluid theremin to be mapped to data messages, we will be able to directly see how in-tune the player is. This is useful as direct training feedback. It also allows the theremin to not only be an instrument but dually gives it the capability of a MIDI controller. In the latter faculty, it will be able to control any virtual instruments. In both capacities it will be able to be either an input source, or supplemental controller with our next device, the Expandome.

3.3 Expandome

Hochenbaum et al. ported the Monome[3] to the Arduino platform with the Arduinome[4]. They additionally expanded upon this ideally simplistic, yet complex musical tool with the aim at creating an instrument that could elicit a virtuosic experience with ease. The Monome and its ported version are both very minimalist grid devices that are not actually instruments, but controllers. Upon pushing a button, the information is sent via shift registers to the computer, which being controlled by one of the various programs (described below), then tells it what to do. These MIDI-capable programs can begin tracks, set beat-counts, produce direct noise, and be used as anything really imaginable as an input including an alternative game controller pad, keyboard or spectrum visualizer.

The Expandome is an iterative development[5] on the Arduinome, as was the Chronome on the Arduinome to further this easy-grid concept. It is an effort to add to this highly customizable environment by allowing multiple users to interact together and provide a fluid setup that reflects the user's personal performance organization style. It is a highly adaptable MIDI controller implementing a backlit push-button grid. This will be accomplished by allowing each device to work standalone, or pushed together with another device. The devices will be in three grid forms, an 8x8, 4x8, and 4x4 and each device will have internal storage to save setups and beats, so as not to always rely on computers, and dually be able to aid in spontaneous interaction with other performers.

SECTION 4.0 SPECIFICATIONS

4.1 uWave Theremin

The device will meet the following requirements:

- It shall operate with the supply voltage of ±12v.
- Operate with one toggle switch to turn on or off the device.
- Two coil antennas will be mounted (pitch and volume).
- It shall transmit analog signals to the SenseBox.

- Four potentiometers will be included to control the volume, pitch, brightness and waveform of the uWave Theremin.
- Dimension of cabinet.
- It shall have two output jack) – audio and pitch preview.
- Resonant pitch antenna frequency will be 260 kHz
- Resonant volume antenna frequency will be 450 kHz.
- Output frequencies will range from 0 to 3 kHz.
- This device shall be operated with minimal electro-magnetic interference EMI.

4.2 SenseBox

The device will meet the following requirements:

- It shall operate with the supply voltage of 5V.
- Have one set of MIDI sockets - MIDI IN and MIDI OUT (no MIDI THRU).
- It shall receive analog signals from uWave Theremin and convert it to MIDI data. The MIDI data will include, but not limited to: MIDI note number, NOTE ON, NOTE OFF, Velocity, and Control Mode/Change.
- It shall transmit MIDI data out to another MIDI device or DAW.
- It shall receive MIDI data from another MIDI device and store it.
- Dimension – not less than 6" wide x 6" long x 5" high.
- It shall operate only if the incoming voltage from the uWave is 0.8V RMS.
- It shall be capable of running standalone with all primary functions on the ATmega microprocessor.
- It shall corporate with the uWave Theremin using two input audio jacks & pitch preview.
- It will include one LCD screen panel. The LCD will display but not limited to: current mode operation, note name based on the current pitch, volume value, incoming MIDI note number from a MIDI device.
- Operate toggle rotary switch for choosing the mode operation in the SenseBox.
- Contain five LEDs to display the accuracy of the desired pitch from the uWave Theremin.
- Contain three LEDs to display the intensity of the volume that is emitting from the uWave Theremin.

4.3 Expandome

Each device will comes in an 8x8, 4x8, or 4x4 form. Excluding those related grid and enclosure differences, the requirements and specifications below are equivalent between each device unless otherwise noted.

The device will meet the following requirements:

- It shall have an *n* x *m* backlit push-button grid. It will be the primary interaction with the Expandome.
    - The backlit grid will have the capability of having the appearance of being fully lit due to rapidly multiplexing each row.

    - The device shall be given maximum responsiveness towards multiple button presses.
    - There will be minimum delay in noise responsiveness, between pushing a button and immediate data and noise reply.
- It is capable of running standalone with all primary functions on the ATmega microprocessor.
    - Default programs on microprocessor will allow the Expandome to perform without any virtual instrument programs.
- Implement an SD Card Reader with default beats and beat structure. Default programs associated with other devices as well as saved data messages will be retained on this expandable drive.
- Be capable of general device recognition and interacting, via MIDI and USB, with any typical computer virtual instrument program.
- Works as, or with, any other typical MIDI instrument.
- Draw 5V max and have a maximum cascading current draw of 500mA for up to three devices.
    - Current draw for each row is set with the *Rset* resistor. Two resistances will be chosen, one for maximum brightness, and a second for lower power consumption by minimalizing the current draw.
- Run on batteries when unplugged from a power source.
    - No interruptions shall occur, if devices are working in their standalone capacity.
    - This shall aid in power cascade.
- Have two sets of MIDI sockets in/out for 4x4 and 4x8, four for 8x8 devices.
- Act as the primary Expandome device if an 8x8.
    - If two 8x8 devices are connected, the device connected directly to a computer shall be given higher control.
- Perform in an expanded capacity when two or more Expandome are connected, in that they shall perform as one.
- Connect flush with all other instruments from this project.

SECTION 5.0 RESEARCH

5.1 Theremin

Our search for a uWave Theremin design that would serve as a good reference brought us to a few different kinds of representation of the device. Research for different kinds of theremin devices was done on the web at thereminworld.com. Some theremin designs that the group came across included the vacuum tube theremin also known as the RCA theremin, digital theremin (digital components using transistors and logic gates) and analog theremin (active and passive components). Also, there were some simplified theremin designs that only utilized a pitch antenna and omitted the volume antenna, but the group has decided that a small and compact theremin with all its components available would be more flexible to work with when it came time to experiment with more functionality. Some qualities of the theremin designs listed above will be investigated in this portion of the paper, but because the device as a whole will be modular, the group has decided to base most of the theremin portion to Robert Moog's design seen in his DIY paper. Also, Moog's design was chosen because of his impact in the world of electronic music and his efforts on the synthesizer.

As research on all the different variations of the theremin continued it was noticed that they all had something in common with each other. They were all composed of three main blocks – each capable of producing an output and taking in inputs. These blocks include the antenna circuitry, the beat oscillator, and the volume controllers, which can be seen in more detail in our block diagram below. For our research we focused on these three parts individually to understand how the theremin works. In essence, it gave us an idea of what to expect during design and testing, which will be discussed later.

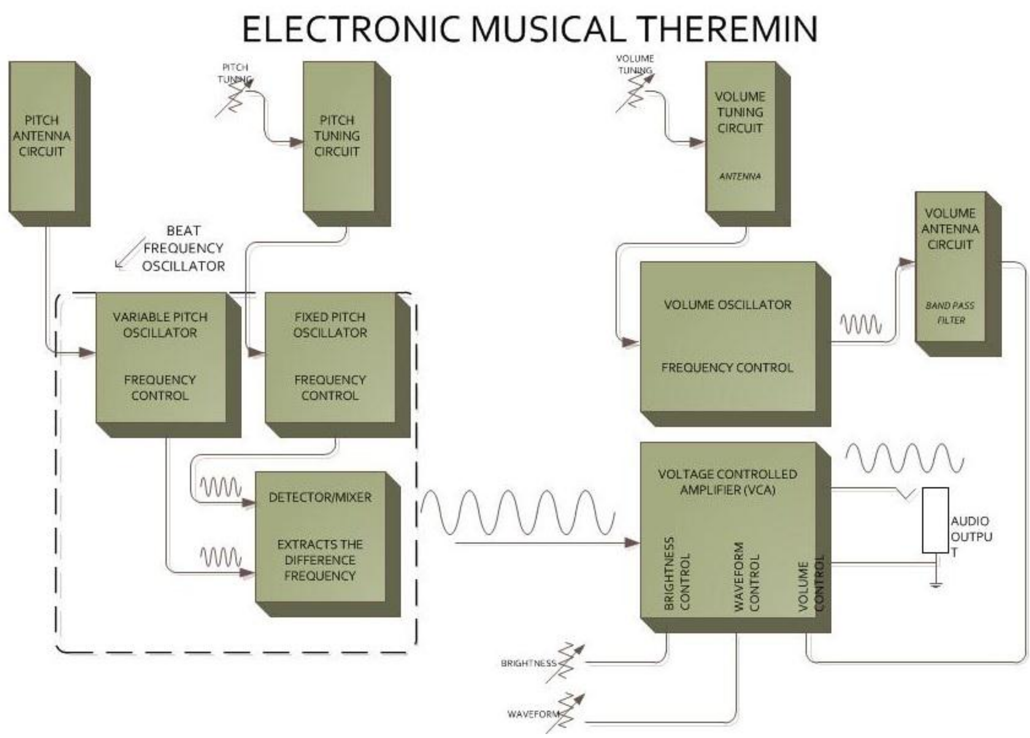

Figure 1: Block diagram of EM Theremin

5.1.1 Antennas

The uWave Theremin utilizes two antennas. Though the term "antenna" will be used, it should be noted that these aren't regular antennas but they are only one plate of a capacitor connected to a circuitry to create a capacitance sensor[6]. As mentioned in the project definition, one antenna will be used for controlling pitch and the other will be used to control volume, or the amplitude of the pitch frequency. In this section we will look at a few antennas made of different materials and see their effects. We will also investigate the pitch and volume antennas and see how their capacitance affects their respective circuits and the uWave Theremin as a whole.

5.1.1.1 Antenna Material Considerations

There are a few different materials that we can consider when designing the pitch and volume antennas; solid brass, brass tube, aluminum, solid copper and copper tube.

A 7/16 inch diameter solid brass or brass tube was noted by one experimenter that they both work equally well. It is necessary to adjust the capacitance of the variable pitch oscillator to get the best performance with each type of antenna. The brass tubing is hard to bend without the purchase and use of special bending equipment. Aluminum is a better option for an antenna because it is flexible.

The uWave Theremin antennas are not like the usual antennas which are designed based on wavelength and polarization factors. As mentioned earlier, the uWave Theremin antennas are designed to behave as one plate of a capacitor, while your hand acts as the other plate. The surface area and length are the important factors in the design of the uWave Theremin antennas (this statement will become apparent in a few moments). Also, the electrical resistance of the antenna may be an important factor since it may add an unwanted attenuation factor. For this reason, we decided to eliminate the possibility of using aluminum in our design since the resistivity in copper is greater. As the inductive reactance increases, so should the linearity of the pitch antenna.

5.1.1.2 Pitch Antenna

Before we begin, it should be noted that most of the equations that will be presented in this section also applies to the volume antenna. In theremin playing, the pitch antenna is responsible for varying the pitch or frequency of the sound produced by the device. It is operated by placing the hand near the antenna thus changing the total capacitance seen by the pitch antenna circuit. The antenna capacitance equation[7] is used to illustrate this:

Equation 1: $$C_A(\infty) = \frac{2\pi\varepsilon_0 h}{\log\left(\frac{2h}{d}\right) - k}$$

where C_A is the antenna's capacitance, h and d are height and diameter of the antenna respectively, ε_0 is the permittivity of free space (8.85×10^{-12} Fm^{-1}), and k is a constant depending on how far above the ground the antenna is mounted, which is about 0.4 for an antenna mounted almost at ground level[7]. The permittivity of free space, ε_0, gives a hint that the capacitance of the antennas will be affected by characteristics like humidity.

Equation 1 is the antenna's capacitance seen when the players hand is far away from the antenna; this equation also applies for the volume antenna. Using Equation 1 and an antenna with h = 0.45m (18 in) and d = 0.0095 m (3/8 in) and k being 0.4 we get C_A to be approximately 15.87 pF. The change in antenna capacitance when the players hand is in proximity is given by the following equation:

Equation 2: $$\Delta C_A \approx \frac{\pi \varepsilon_0 h}{10\log\left(\frac{4x}{d}\right)}$$

Where X is the players hand distance from the antenna (it is assumed that this distance is greater than d)[7]. So, according to Equation 2, if a players hand is 4 m away from the antenna, a change in capacitance of only 0.43 pF is introduced to the antenna circuitry.
As the player puts his hands close to the pitch antenna, the pitch frequency increases and as the player puts his hand away from the antenna, the pitch frequency decreases. The pitch antenna will be vertically placed on the right hand side of the uWave Theremin as this configuration is more sensitive to the players hand far away from it and less sensitive when the players hand is close by[8].

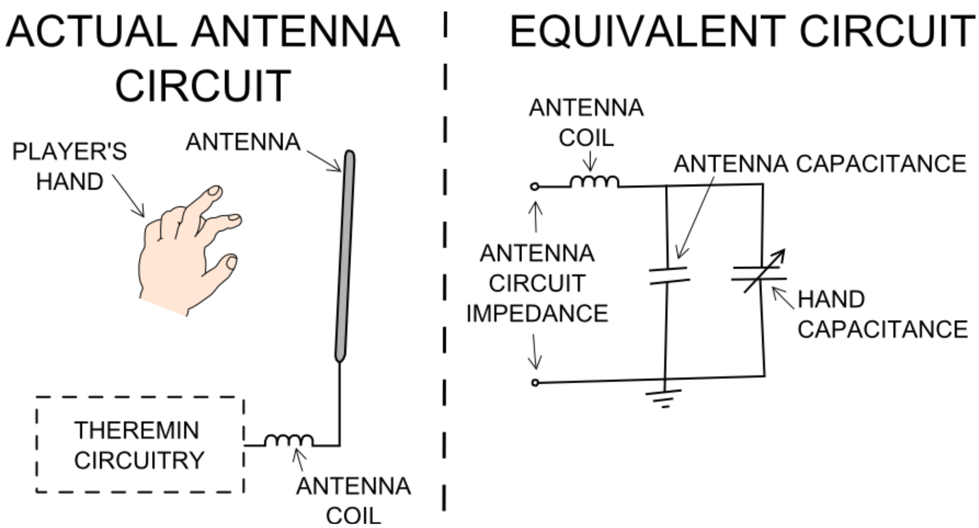

Figure 2: Equivalent representation of hand and antenna capacitance

Since the antenna is to be straight, a sufficient method for straightening it without leaving any kinks has to be considered. If the antenna isn't purchased straight, then the antenna can be easily straightened by hand or by using a tube bender on it slowly and gently. The pitch antenna of choice will be eighteen inches long and 3/8 in diameter. This dimension is chosen because it gives the uWave Theremin a compact look to it. Although the length was chosen for aesthetic reasons, the length of the antenna will have an indirect effect on the hand-to-antenna capacitance by not only affecting the antennas capacitance, but also the hand-to-antenna range as demonstrated in equation 2.[9] This will be important as it has an effect on the pitch antenna's reactance and the total impedance of the pitch circuit. Because of its relatively short length, it was cascaded with multiple inductors to form a resonant frequency of 260 kHz[8]. This frequency was chosen in order to reduce interference from other devices in the vicinity of the uWave Theremin. Since hand to antenna capacitance can vary on the order of 1pF to 15pF, the impedance of the pitch antenna circuit is drastically altered when the user brings his/her hand in proximity of the pitch antenna. The equivalent circuit of the antennas can be seen in Figure 2.

5.1.1.3 Volume Antenna

The volume antenna is quite similar to the pitch antenna. It is responsible for varying the amplification of the audio signal. It is operated in the same fashion as the pitch antenna. As seen in Figure 2, while the user moves his/her hand about the antenna, the total capacitance is altered, which in turn alters the impedance of the volume antenna circuitry. In general, when the players hand is away from the volume antenna, the volume amplification factor is increased and when the hand is close to the antenna, the volume amplification is decreased. According to the design seen in Robert Moog's paper, the volume antenna will be mounted horizontally on the left. Mounting the pitch antenna in a vertical manner on the right hand side and the volume antenna in a horizontal manner on the left makes sure that the two antennas are perpendicular to each other. This configuration is standard for most two antenna theremin devices as it minimizes the interaction between the two antennas and makes playing the theremin a little more intuitive as it is easier to grab and hold the volume antenna without having to lift your arm in order to mute the theremin[8].

Some of the theremin devices researched had both the pitch and volume antennas straight. Some didn't even use metal tubes for the antennas, instead a metal sheet was used. However, for the purpose of this project the pitch antenna of interest will be a vertical tube antenna. The volume antenna will be similar to the elongated semi-circle proportions seen in Robert Moog's paper. The reason for this is because it'll give the device a more compact look and still maintain the device's historic design.

In order to bend the volume antenna, a tube bender can be used or it can be bent by hand gently and slowly. Though the soft copper tube is easy to bend by hand, one big disadvantage is that unwanted kinks can be left after bending is complete[7]. One tube bending method performed by traditional trombone craftsmen was to pour soapy water in the tube before bending. On some theremin forums visited[10], it was mentioned that the antenna can be stiffened by putting fine sand in it prior to bending. In Moog's paper, he suggests putting stiffening resin in post-bending to reduce unwanted kinks. The total length of the bent volume antenna will be around nine inches. The volume antenna will be cascaded in series with inductors to create a resonant frequency of about 450 kHz. The effects of the players hand on the volume antenna are similar to that of the pitch in the sense that the impedance of the volume antenna circuit is altered by the presence of the hand.

5.1.2 Oscillators

Essentially, "the basic principle of the theremin is the heterodyne oscillator".[9] "An heterodyne oscillator is a radio signal technique in which new frequencies are created by combining or mixing two frequencies".[11] Therefore, in order to create the pitch or sound we can hear, the uWave Theremin will be composed of multiple oscillators set to oscillate at two distinct frequencies. These two frequencies are slightly close to the resonant frequencies of the pitch and volume antenna circuits. As seen from the block diagram above there will be three oscillators associated with the uWave Theremin. These oscillators are the fixed pitch oscillator (FPO), the variable pitch oscillator (VPO) and the voltage oscillator. The voltage oscillator will oscillate at a frequency of 432 kHz, while the fixed and variable oscillators will oscillate at a frequency slightly higher than the pitch circuit's resonant frequency of 285 kHz. We will further investigate the functions of these oscillators in this section.

5.1.2.1 Beat Frequency Oscillators

As mentioned earlier, the theremin uses a heterodyne technique to create one frequency. In heterodyne oscillation, two frequencies, f_1 and f_2, are combined to form new frequencies with components $f_1 + f_2$ and $f_1 - f_2$.[12] This is a hallmark in theremin design and it is accomplished using the beat frequency oscillator. Essentially, the beat frequency oscillator is composed of a detector and two oscillators - the VPO and the FPO. These oscillators will interact together with the detector (to be explained in a few sections) to create the audio signal. Basically, we will have two oscillators set to oscillate at a frequency slightly higher than the resonant frequency of the pitch antenna[8]. One of these oscillators will be fixed, and the other will vary in response to the pitch antenna. Their outputs will be fed to a detector in order to retrieve the component frequency $f_1 - f_2$. So for example, while the FPO is fixed at 285 kHz and the player puts his/her hand directly to the pitch antenna, the output of the VPO can be about 283 kHz. The output frequency from these two oscillators will have components $f_1 + f_2 = 568$ kHz and $f_1 - f_2 = 3$ kHz. For theremin playing we are only interested in the latter frequency therefore a filter will be used. In the case when the player takes his/her hand the maximum distance away from the pitch antenna, the oscillators will be at a “null point” and have the same frequency of about 285 kHz. Therefore, the output frequency of the two oscillators combined will have component frequencies of equal to $f_1 + f_2 = 570$ kHz and $f_1 - f_2 = 0$ Hz.

5.1.2.1.1 Fixed Pitch Oscillator (FPO)

The FPO will be used to create the reference frequency. This oscillator is always included in every theremin the group encountered. As mentioned earlier, there are many variations of the theremin, this also includes the oscillators. Some oscillators have been made using a crystal oscillator normally used for IC devices. In Moog's design, he mainly uses RLC components together with BJTs and op-amps. The advantage of using strictly just active and passive devices without a microcontroller allows us to build the uWave Theremin without the need for programing. According to Moog's design, the FPO will oscillate at a nominal frequency slightly higher than the pitch antenna resonant frequency of 260 kHz. It will also interact with a detector and a pitch tuning circuit (to be discussed later), that will act as a means by which the player can adjust the frequency of the FPO.

5.1.2.1.2 Variable Pitch Oscillator (VPO)

The VPO is very much similar to the FPO. In fact, their circuit design is almost identical. It is initially set to oscillate at the same nominal frequency of the FPO at about 285 kHz. The big difference between the FPO and the VPO, which is obvious from its name, is that the frequency it oscillates at is not constant. The frequency of the VPO is a function of hand to antenna capacitance. The variation in waveform frequency from the VPO is due to its interaction with the pitch antenna circuitry. As explained in section 5.1.1.2, the impedance of the pitch antenna circuitry will be altered by how close the players hand is to the pitch antenna. This impedance seen by the VPO circuit will alter the current it draws and the rate by which its active components stores and releases energy, hence the change in output frequency.

5.1.2.1.3 Beat Detector

As stated in the previous two sections, the uWave Theremin will use two oscillators to create the audio signal - one being a reference oscillator and the other a variable oscillator. The VPO serves to create the variation in frequencies ranging from 0 to 3 kHz (about 3½ octaves above middle c)[8]. This is more than enough range of sound for the player to experiment with as seen from the keyboard in Figure 3. The figure shows the range of sounds or notes that can be produced from the uWave Theremin demonstrated using a keyboard.

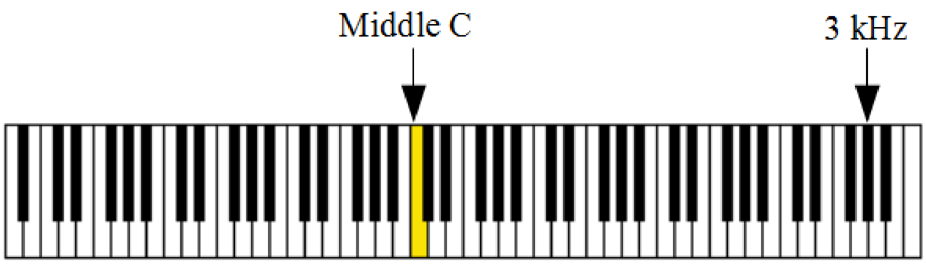

Figure 3: Range of sounds produced by uWave demonstrated using a keyboard

In order to retrieve the beat frequency, a detector or demodulator is used. The detector is mainly composed of mixer, which can be in the form of a single diode, and a low-pass filter as seen in the block diagram below. This block diagram shows what the detector is consisted of and how it interacts with the VPO and FPO.

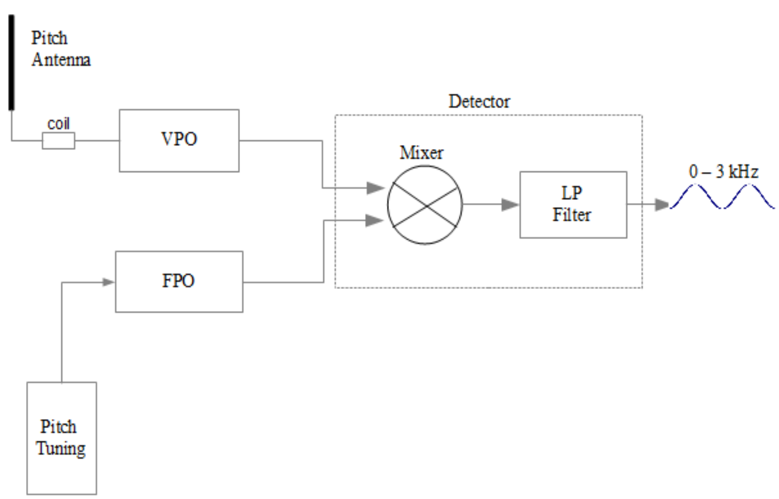

Figure 4: Block Diagram showing the function of the Detector

As seen in the figure above, the signals from both the VPO and the FPO are sent to the mixer, which will be used to create two distinct frequencies through the process known as heterodyning. The result of the mixed signals is amplitude modulated (AM) signal illustrated using the equation below.

Equation 3: $$V_{mix} = \frac{A}{2}\left[\cos(\omega 1 - \omega 2)t - \cos(\omega 1 + \omega 2)t\right]$$

As stated before, a heterodyne frequency is one that has two component frequencies of $f_1 + f_2$ and $f_1 - f_2$. Since we only require the audible frequency from the heterodyne frequency, the output of the mixer seen in equation 3 is fed to a low-pass filter that has the output given in Equation 4.

Equation 4: $$V_{out} = \frac{A}{2}\cos(\omega 1 - \omega 2)t$$

There are a few variations of mixers like the SBL-1 mixer that can be used to accomplish the heterodyne technique, but for this paper we will briefly examine the diode mixer (which will be equivalent to our implementation). The diode mixer is a passive mixer that usually produces a low output power signal. This type is mixer is an unbalanced mixer that produces the sum and difference of the original input frequencies. An advantage of this kind of mixer is that it has a very low power consumption making it a sufficient choice for our project.

From our block diagram in Figure 4 it is clear to see that after mixing has taken place, the signal will be fed to a low-pass filter in order to restrain the high frequencies ($f_1 + f_2$) not needed for our application. Only the low frequencies ranging from 0 to 3 kHz should be preserved. This small output signal contains the beat information and can be fed directly to the SenseBox module later to be discussed. As seen in Figure 4 the low signal output from the detector circuitry is fed to the voltage control amplifier (VCA) for complete audio processing (to be discussed shortly).

5.1.2.2 Volume Oscillation

Most theremin implementations use a volume oscillator circuit to control the voltage amplification of the pitch data coming from the detector. As stated in Fred Mundell's investigation of the theremin, “the most common means of implementing this function is by effectively duplicating the pitch antenna oscillator, but having these operate at a higher frequency which does not interfere (has no important overlapping harmonics) with the pitch oscillator, and then converting the frequency variation from this oscillator to a varying voltage or current”[6]. This seems to be a similar case in Robert Moog's implementation in which he states that “as the player brings a hand near the volume antenna, the resonant frequency of the volume-antenna circuit is lowered, and the DC voltage is reduced”[8]. Also, in Robert Moog's implementation he sets the volume oscillator to oscillate at 450 kHz. This is in correspondence with Fred Mundell's comments about setting the volume oscillator to a frequency greater than the pitch oscillators and thus this condition will be implemented in our design.

In actuality, the frequency at which the volume oscillator should operate at is determined by the DC voltage coming from volume oscillator and antenna circuitry. The rectified DC voltage will be sent to the VCA in order to process the amplification factor of the pitch signal coming from the detector.

5.1.3 Tuning

The ability to tune a musical instrument, be it a guitar clarinet, trumpet, or piano, is very vital in the operation and performance of a musical instrument. Most musical instruments like the ones mentioned above are usually tuned by players in reference to the *A* note, which has a frequency of 440 Hz. Tuning can also be done in order to allow players the ability to achieve different sounds on a different musical scale. The theremin is also one of those instruments that can be tuned to the players pleasing. The type of tunings done in theremin playing are usually not done to tune the device because it might be out of range of the reference *A* note, but it is done to achieve a different sound relative to the hand-to-antenna distance. This section describes the method by which the uWave Theremin is tuned and how tuning affects the internal components of the device.

5.1.3.1 Pitch Tuning

Connected to the fixed pitch oscillator is a pitch tuning circuit that will be available to the end user. "This circuit presents a variable active impedance that is used to make fine adjustments to the FPO frequency while the instrument is being played"[8]. Basically in the case of the pitch audio, the pitch tuning circuit will be used to shift the frequency range, which allows the player to keep his/her hand at relatively the same location from the pitch antenna while playing at a different sound octave. The player may choose to do this because this location might be more comfortable for the arm or that this location produces a more or less sensitive response that best fits the players' musical performance.

Most sophisticated theremin devices come with a pitch tuning option that allows the player to play a different range of pitch without too much hand extension from the pitch antenna. In previous theremin years, the pitch circuit included a large variable capacitor[8]. These capacitors make the design rather cost ineffective, therefore a cheaper and more economical method involving potentiometers was implemented in our design.

5.1.3.2 Volume Tuning

The volume tuning circuit should basically serve the same purpose of the pitch tuning circuit in that it will be used to adjust the range by which the hand and antenna affects the volume oscillator. The volume tuning circuit acts as an active impedance connected to the volume oscillator. In effect this circuit has an effect on the frequency generated by the volume oscillator circuits. As mentioned earlier, the volume oscillator works with the volume antenna to create a DC voltage that will determine the amplification factor for the pitch audio, therefore this range of DC voltage or current needed for amplification is changed due to the volume tuning circuit.

Since the volume tuning circuit closely mirrors the pitch tuning circuit, a potentiometer option rather than the extravagant variable capacitor was used as a tuning parameter for the volume tuning circuitry.

5.1.4 Volume Amplification

The pitch signal from the detector circuitry is usually a very small signal and not fit to produce an audible audio signal, therefore this signal must be amplified. This section will focus on how the pitch signal will be amplified using two different operational amplifiers, and how audio can be sent to the other modules of the system.

5.1.4.1 Voltage Controlled Amplifier (VCA)

As the name implies, a voltage controlled amplifier is a type of amplifier that varies its gain factor depending on the control voltage/current given to it. As mentioned earlier, the volume oscillator and antenna circuitry will work with the VCA to amplify the small pitch signal coming from the detector. This is done by using the DC control voltage (CV) or current (varies with hand presence on the volume antenna) from the volume antenna/oscillator as the control voltage.

In different theremin implementations, the VCA primarily performs the same function, but have different topologies and components used. Some theremin implementation the group researched can be found at thereminworld.com. The theremin implementations of interest include the Open Theremin and Robert Moog's Etherwave. The Open Theremin uses the MC34074AP operational amplifier while Robert Moog uses the LM13700 dual operational transconductance amplifier. Before we begin, it should be noted that the Open Theremin is mostly a digital theremin option while Robert Moog's implementation more closely resembles the original theremin without the use of tubes.

Factors	LM13700	MC34074AP
Slew Rate	50 V/us	10 V/us
Gain Bandwidth	2 MHz	4.5 MHz
Cost	$0.54	$0.84
CMRR	110 dB	97 dB

Table 1: Comparison of two amplifiers suitable for use as a volume controlled amplifier.

From the table above, it is noted that both the LM13700 and the MC34074AP are very similar in characteristics that fit the project need with the exception that the LM3700 is cheaper and has a smaller gain bandwidth. The LM13700 used in Robert Moog’s design would serve as a better choice for use as a VCA though it lacks in gain bandwidth when compared to the MC34074AP used in the Open Theremin design. This can be over-looked because a VCA with a gain bandwidth of 2 MHz is more than enough for our application where the maximum signal through the VCA will be less than 1 MHz.

5.1.5 Power Considerations

The uWave Theremin requires a dual power supply which provides a positive and negative 12V. It was be necessary to design a regulated power circuit to allow us to minimize the noise that we may observe in the amplifier circuits used. The zener regulation can be used to control and provide a steady output voltage of a circuit even if the input voltage is changed. We will use these techniques in our design of the power supply. The reference design[8] included a power supply circuit which used capacitors and voltage regulators to convert a stepped down AC power into positive and negative 12V DC. There are several power options which may be considered to provide power to the uWave Theremin.

uWave Theremin Power Option 1 "Wall Adapter":
Provide a power source by using a 16V AC wall adapter with a minimum of 200 milli-amperes. Connect the AC power source to a power converter circuit which will be designed to convert the voltage from AC to DC and then step down the voltage to positive and negative 12V DC. The power converter circuit will consist of two half wave rectifiers with positive and negative voltage regulators to give us ±12V. This circuit would require the following components: two 1N4001 diodes, two 2200uF polarized electrolytic capacitors, tow 0.47uF capacitors, two 0.1uF capacitors, one LM78L12 regulator for positive 12V DC, and LM79L12 regulator for negative 12V DC. When choosing a wall adapter power source, we must ensure that it is a three prong power plug to provide proper grounding to the circuit and avoid unwanted humming or noise. The capacitors in the power supply circuit provide filtering.

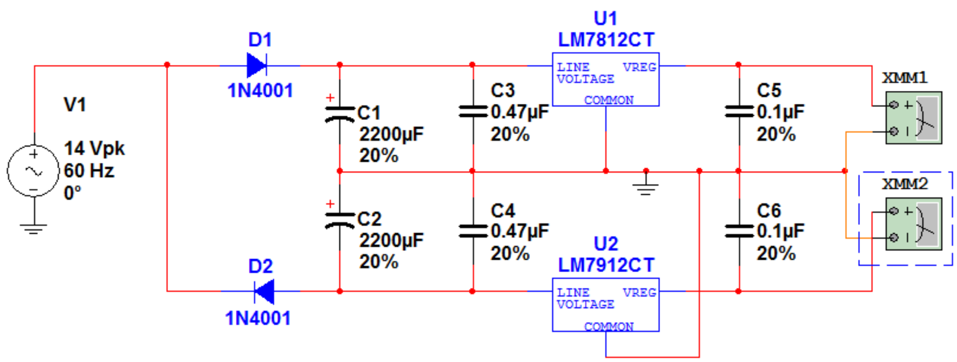

Figure 5: uWave Power Option 1

uWave Theremin Power Option 2 "Transformer":
An alternate design for a simple power supply which provides positive and negative 12 volts DC is to use a 16 volt 1A transformer to step down 110 volt AC to 16 volts DC, which may be accomplished by using a 14VDC wall adapter. The circuit uses a full wave bridge rectifier with 4 diodes, two 1000uF polarized electrolytic capacitors, two 0.1uF capacitors, and two voltage regulators as mentioned above. The transformer first performs the AC to DC conversion and then the voltage regulators are used to step down or convert the DC to 12 volts DC. When using voltage regulators to convert or step down the voltage, the voltage in must be a minimum of 3 volts greater than the desired voltage out.

The PCB mountable transformer is typically big and bulky in comparison with the electronic components and board. The design should minimize the size of the components so that ultimately the size of the printed circuit board and the size of the enclosure are as small as possible. By making the device small, it will be more portable and also affordable for this project.

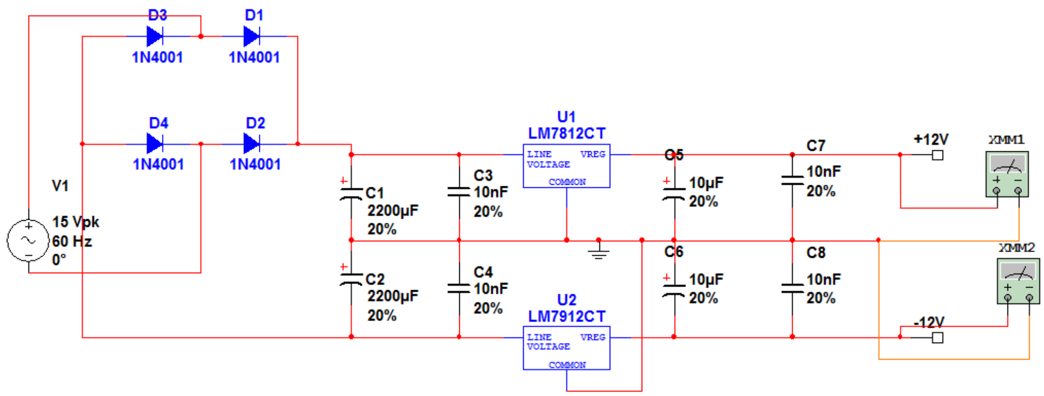

Figure 6: uWave Power Option 2

uWave Theremin Power Option 3 "Battery":

Connect two 9V batteries in series to obtain 18V DC. Connect the 18V DC power supply to a DC to DC converter circuit which will step down the DC voltage to ±12V DC. The uWave Theremin must be connected to ground. If batteries are used as the primary power source, proper grounding is required.

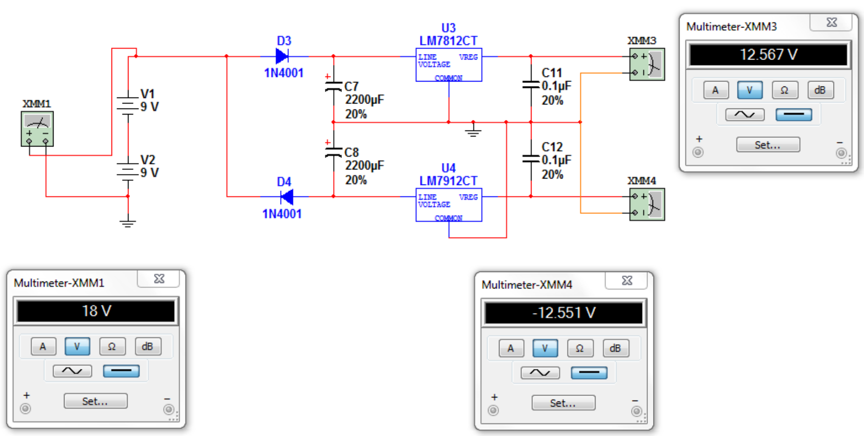

Figure 7: uWave Power Option 3

uWave Theremin Power Option 4:

The uWave may be designed with a dual power source option which uses a DC battery power source or an AC wall adaptor power source, as described in uWave Theremin Power Option 1 and uWave Theremin Power Option 3. This would require in introduction of a relay or a switch which would allow us to either automatically choose the power source or manually choose the power source.

uWave Theremin Power Remarks

When selecting a wall adapter we must ensure that it will supply at least 3 volts greater than the desired controlled voltage. Also, when selecting a wall adapter for this project the amperage rating must be sufficient for the project. For a 5V DC 100 milliamps regulated power supply, we can choose to use a 9 VDC wall wart with a 100 to 150 milliamps rating. For a 12 volt regulated power supply, we can use 14 to 16 VDC wall adapters with a minimum of 200 milliamp rating.

5.2 SenseBox

The SenseBox is the design element in this project which takes the signal from the uWave, a musical instrument which produces an analog output, and converts its output to into digital data. With this capability a musician will be able to modify the sounds of the instrument to produce a unique array of sounds and melodies.

5.2.1 Consideration of Microprocessor

Before building the components for the SenseBox, it was necessary to consider which microprocessors will be utilized to take the incoming analog signal from the theremin and convert it into digital data and simultaneously emulate it as a MIDI protocol.

In this project, the prototype model was based on an open-source microprocessor board. There are several microprocessor boards in the market that have the specification to carry the functions of this device. However, many of them are considered not intuitive for novice electronics hobbyist. For instance, Texas Instruments' Beagleboard is an open-source hardware single-board and though it has the basic functionality of a computer (for instance, it has OpenGL ES 2.0 capable 2D/3D graphics accelerator), it is costly and for the purpose of this project, we do not need any extra features that we will not use. Thus, it was decided to use an Arduino microprocessor board as a foundation of reference to for initial prototype.

Arduino is considered to be a preferred choice among hobbyist when modeling electronics projects. Its simplicity and flexibility, as well as vast myriads of resources in the internet, has earned an earnest do-it-yourself community, which helped our decision to choose this microprocessor board. Specifically, in order to have a microprocessor act as an MIDI protocol, the microprocessor must have a large enough Flash and RAM memory to store a look-up table for the MIDI protocol; Arduino's microprocessor, ATMega328p can handle the task. The ATmega328p microcontroller has 32 KB flash memory space and it is preloaded with the Arduino bootloader, which allows the chip to program directly through the Arduino language.

Furthermore, operating in low-power voltage (5V) and with a clock speed of 16MHz, ATmega328p had met our criteria for building the SenseBox (MIDI protocol operates at 32,250 bits per second—Arduino's built-in serial port can send data at that rate). A list of other specifications for the ATmega328p is given below[13]:

- 8 bit microcontroller
- 6 PWM channels
- 6 analog inputs (10-bit Analog Digital Conversion)
- 14 digital input/output pins
- 32 KB Flash
- 2 KB SRAM

Furthermore, programming the ATmega328p is natural to comprehend: the microprocessor is programmed in the Arduino environment language, which is based on the Wiring programming language. Similar to programming through assembly language and C-like language, the Arduino programming language has built in functions, variables, and mathematical operations for the programmer to control. As a result, the programmer can navigate and interface with various peripherals through the analog and digital to communicate with the microprocessor[13].

5.2.2 Audio Interface

One of the versatile things about having MIDI instruments and controllers is that it allows the user to record MIDI data through a DAW (Digital Audio Workstation). For recording musicians, this helped them to capture and enhance their creative art-work by manipulating sounds through the digital world. Thus, in actuality, when the theremin is connected to the SenseBox, the theremin acts as an MIDI controller.

However, we felt that in honor of Léon Theremin, the inventor of the theremin, we would not want to disregard the actual analog output of the theremin itself. Otherwise, it will lose a great deal of expression of playing an actual theremin.

5.2.2.1 Bit Depth & Sampling Rate

Ideally, we have to choose an audio interface that has a sampling rate of at least 16 bit depth - the higher bit depth, the better overall sound quality will produce. It is essential, however, to have a sampling rate at least 40 kHz because in order to sample a good quality audio, it must fulfill the Nyquist criterion (in terms of human physiology, the human ear can perceive frequencies from 20 to 20 kHz. Hence by Nyquist criterion, it must sample 2 times of 20 kHz to assure us that the original signal can be constructed without any loss. Otherwise, the conversion of the audio will face aliasing[14]).

Fortunately, today's market on audio interface meets our criterion. The following sections will discuss other considerations when choosing the right audio interface.

5.2.2.2 USB vs. Firewire

Today, there are wide ranges of audio interface available. In fact, many USB interfaces have almost identical specifications. If the user wishes to have a higher resolution than our criterion, the user may consider investing in a Firewire audio interface. Firewire can handle 24 bit depth/ 96 kHz sampling rate and can record multiple tracks at the same time (whereas USB can handle one or two tracks). Firewire devices also operate higher speeds than USB; Firewire is almost 70% percent faster than USB when it comes to writing data[15].

Yet, we have chosen to connect with USB for multiple reasons: for the purpose of this project, it will record one input at a time (from the theremin's monophonic output), and since most computers are equipped with USB ports rather than Firewire, our group felt this was the obvious choice. Furthermore, USB audio interfaces are considered cheaper and have wider range to choose from.

5.2.2.3 Latency

When it comes to home recording, one of the biggest issue that a novice will face is latency. In digital studio setup, latency is the amount the time it takes an incoming sound signal traveling in the audio interface, where it is converted from analog to digital, placed on an output bus, converted back to analog, amplified and then transmitted to speakers or headphones for you to hear[16].

As you can see, if latency is too high, this may cause frustrations when recording with a musician—the musician's groove should be synced with the other tracks in DAW at all time. Gratefully, Steinberg created a simple solution: ASIO driver.

ASIO, or Audio Stream Input/Output, is a computer sound card driver protocol for digital audio. Distributed by Steinberg, this protocol provides a low-latency and high reliability interaction between a software application and a computer's sound card. In essence, this allows the user to hear what the input will sound like (with or without VST plugins) before it is recorded in the track.

Audio Interfaces	Bit depth	Sample Rate	ASIO driver	Price Range
Alesis Guitarlink	16	44.1kHz	No	$35.00
Rocksmith Real Tone Cable	16	48.0kHz	Yes	$30.00
Behringer UCG 102 Guitar to USB interface	16	44.1kHz or 48.0kHz	Yes	$35.00
Hosa ¼ in USB guitar cable	16	48kHz	No	$25.00
Peavey Xport	16	48kHz	Yes	$60.00
M-audio Fast Track USB	16	48kHz	Yes	$199.00

Table 2: Different models of audio interfaces

Based from reviews in the internet, the group chose to use the Rocksmith Real Tone Cable as the audio interface because not only it is compatible with an ASIO driver and is considerably cheap, but also it is a plug and play device. The audio interface is simply a cable with 1/4 inch connector on one end and USB male jack on another as shown below:

Figure 8: Rocksmith Real Tone Cable.

5.2.3 A/D Conversion Method

Recall that MIDI is simply an electronic command. One of the challenges that we faced in this project is to figure out how to convert the analog audio input from the theremin to a MIDI data, which contains the volume and pitch information that correlates with the given audio input. The following sections provide our methods of converting an audio input to a MIDI data.

5.2.3.1 Volume Conversion

Capturing the volume data for MIDI involves integrating with an envelope function. Generally speaking, the envelope follower takes the raw audio waveform (usually in AC) and impulses a varying DC voltage, which represents the amplitude of the audio signal. Next, filtering is used to polish the output in order to filter out the low frequency. Principally, this method is used to capture solely the volume data. The signal will be placed into one of the analog inputs in the processor board.

In the MIDI protocol, the volume channel has hexadecimal value from 0 to 127, where the value 127 represents the maximum volume.

5.2.3.2 Pitch Conversion

Capturing the pitch data for MIDI involves integrating a threshold comparator. The comparator will generate a square wave and will be fed into the microprocessor. The square wave output of the comparator is used to convert the analog signal into discrete and by doing so, it will sample the incoming signal and compare with the library stored in the microprocessor. Unlike the volume conversion, we take only the pitch analog sound from the pitch preview terminal of the uWave Theremin.

Currently, U.S and many other countries calibrate and tune their acoustic instruments in the frequency of 440Hz (440Hz is the musical note A, above middle C). Although it is not universally accepted, MIDI Manufactures Association has agreed that this will be the MIDI Tuning Standard (MTS) in the protocol[17].

To understand how MIDI can translate a given analog pitch to its correspondent MIDI note number, the section that follows will clarify this concept:

5.2.3.2.1 MIDI Note

The following chart corresponds to the note frequencies in equaled-tempered scale (in this case, we are looking at only near octave range of the tuning reference note).

MIDI Octave	MIDI Note Number	Note Name	Frequency Hz
0	60	C	261.626
0	61	$C^{\#}/D^{b}$	277.184
0	62	D	293.665
0	63	$D^{\#}/E^{b}$	311.127
0	64	E	329.628
0	65	F	349.228
0	66	$F^{\#}/G^{b}$	369.994
0	67	G	391.995
0	68	$G^{\#}/A^{b}$	415.305
0	69	A	440

Table 3: MIDI notes and frequencies

In traditional music theory, note pitches are classified through the first seven letters of the Latin alphabets (A,B,C,D,E,F and G). Letter names with accidentals, denoted by # or *b*, simply means that there exists a note between two given notes. For illustrative purposes, we will assume that the chart above are the only notes available to us (realistically, the range of pitches is infinite. For more information on music theory, please visit at your local library). To convert an analog pitch to its correspondent MIDI note number, the MIDI Manufactures Association decided to use the following formula for MTS:

Equation 5: $$\boldsymbol{p = 69 + 12 \times \log_2 \frac{f}{440Hz}}$$

Where *p* indicates the MIDI note number, and *f* indicates the analog pitch frequency. As you can see from the previous table, the MIDI note number corresponds to its analog frequency, based on the formula that MTS has provided.

5.2.3.3 Noise Performance Considerations

Another challenge that we faced when is interpreting the continuous analog sound of a theremin to MIDI. Since MIDI is the discrete nature of the digital form, we must find a way for our converter to understand that there exist pitches between the MIDI note numbers. Otherwise, it will evidently loose a great deal of expression of the uWave Theremin if the issue is not addressed.

The analogy of this topic may be compared to a violinist. By nature, when the violinist changes his note, or pitch, the violinist must be able to keep his bow to the violin and move horizontally (with respect to the strings) until the desired pitch is acquired. During this course of action, there were series of notes that were produced continuously - vocal slide between the pitches. Our goal is to come close as possible to capture that characteristic, or also known as "*portamento*".

On the other hand, we may use this issue as an advantage. Recall that instruments such as piano, xylophone and harp produces pitches in discrete, or "*glissando"* characteristic. It would make sense that in order to play those instruments through a uWave Theremin, the *glissando* will be the device's default state. Hence, a feature can be added to allow the user to choose either *portamento* or *glissando* mode.

5.2.4 LCD Display

As mentioned before, playing with the theremin requires no physical contact with the user. Consequently, this means that the user is demanded to have great aural skills in order to play it proficiently. In fact, 1 in every 10,000 (1%) Americans have perfect pitch - an ability to identify the letter-name of a sounded note[18]. Although many believe that perfect pitch can only obtain through heredity, it is proven with great amount of work and hours invested, one can obtain that valuable skill. However, many of us may not have that great deal of patience to train ourselves, especially if not trained musically at a young age.

Thus, by installing an LCD (Liquid Crystal Display) in our SenseBox, we delivered a visual assistance for the user to show them where the notes are. Although this will not eliminate all the strenuous hours of practice to become a skilled thereminist, the LCD display will be a good training tool to utilize.

LCD are extremely common in electronics because they consume relatively low power and are reasonably affordable. LCD has built-in pixels that are turned on and off, based on applied electric field[19]. In the implementation of the SenseBox, an LCD will be used as a feedback tool programmed using ATmega328p processor.

5.2.4.1 Different Types

In today's market, there are several different types of LCD screen that were considered. Selections can range from minimal characters to touch screen with color display functionality. The following sections reveal some popular type of LCD that we have considered implanting in our project.

5.2.4.1.1 Segmented LCD

Segmented LCD are commonly found in digital clocks, electronic meters, and other electronic devices that are used to display alphanumeric characters; they are indicated as “segmented” because each segment of the LCD are individually turn on or off and by certain combination of each segments, they can produce a simplified representation of a character[20]. One advantage of using this LCD is that it consumes low power, but due to its less-complex design, it has very little flexibility and only limited number of characters can be displayed.

5.2.4.1.2 Character LCD

Character LCD, or also known as Dot Matrix, can display individual alphanumeric characters, using a matrix of pixels. Character LCD’s are denoted by the dimension of their screen, in terms of rows and columns.

Each character that can be displayed in the LCD is made up of a 5 x 7 pixel array. The ramification of this is that it allows the user to have higher flexibility to display custom characters through programming it.

Similar to the segmented LCD, character LCD consume low power and are relatively easy to program. However, due to its simple resolution, it cannot include images or other graphical interfaces - it is confined only to only texts.

5.2.4.1.3 Graphical LCD

Graphical LCD shares the same characteristic with character LCD, except the addition of being able to display any type of graphic or text in the screen panel. The grouping of pixels across the screen is individually accessible, which allows the user to program any area of the screen. Incorporating graphical LCDs includes flexibility in design, as well as larger interface to program with. However, because it is a larger unit than the character LCD, the disadvantage of incorporating a graphical LCD includes more power consumption, more difficulty when implementing in the controller circuits and requires more complexity in programming[19].

5.2.4.1.4 Backlighting LCD

Although LCD is great for displaying visual data representations, designers may consider installing a backlight for illumination purposes. Depending where the device is, reading of a non-backlit LCD screen may be difficult, even in adequate lighting conditions. Even though backlighting may require more power usage, it will make the device more adaptable when reading the data display is critical.

5.2.4.1.5 Color vs. Monochrome LCD

Before investing in an LCD, our group had considered if we wanted to incorporate color or monochrome LCD panels. Color LCD panels uses an active-matrix structure - they use a matrix of thin-film transistors in contact with the Liquid Crystal layer[21]. Monochrome LCD’s are typically passive-matrix—these LCD panels are typically great for only information display purposes. Both consume low power and can display graphics. However, color LCD are typically more expensive than monochrome.

5.2.4.2 LCD Driver

Typically, one way to drive an LCD screen with an Arduino is having seven or eight pin wires and few other components connecting to each other. Above all, the programmer must be aware of numerical parameters in order to program the LCD, which becomes tricky and cumbersome[22]. Thus our choice of including a driver eliminates the number of pins of a typical LCD from 14 to 3—ground, +5 volts and serial data. The driver has a small microprocessor that sits between the main board and the LCD. In addition, it eliminates the tedious programming chores since it contains a programmed PIC chip in the driver, which converts serial commands into LCD text. It also includes a potentiometer that is soldered in the driver, which enables the control of the display contrast.

5.2.5 Audio Input/Output Considerations

Recall that in order for the SenseBox to work, it needs to fetch two data from the analog input: a pitch data and volume data. Thus analog cables are essential when connecting the uWave Theremin to the SenseBox. In addition, the SenseBox will include an audio output, which can be connected directly to an audio interface or a speaker.

5.2.5.1 XLR Cables

The XLR cables are commonly used for microphone cables. Typically, XLR cables have a circular design between 3 and 7 DIN pins—they are balanced cable, which enables them to go long distance without picking up any external noise (unlike unbalanced cables). Furthermore, XLR connectors can be used for lighting control, low-voltage power supplies, and other applications[23].

One drawback for using an XLR Cable is that they are relatively more expensive than other analog cables due to its cost of durability.

5.2.5.2 TRS & TS Cables

TRS connectors (tip, ring, sleeve) are commonly found in analog application such as those seen in guitar cables. . TRS are also known as audio jack, or phone jack, because they are commonly found in headphones, computer speaker, and many other music player devices.

TRS are designed in a cylindrical shape and have three contacts: ground, and two signals (hot and cold). The three contacts enable them to be compatible with stereo devices. In addition, they are balanced cables, which allow them to go long distance, similar to XLR cables.

TS connector (tip, sleeve) on the other hand has the same specification as TRS, but since it only has two connectors, they are compatible with only mono devices. TS are unbalanced connectors. Both TRS and TS are relatively inexpensive to invest in.

In conclusion, we have installed TS connectors into the Sensebox. Since the theremin is a monophonic instrument, meaning that it only plays one note at a time, TS connectors seemed the obvious choice to have audio connections onto the Sensebox.

5.2.5.3 RCA Cables

RCA connector, or sometime called phono-connector, is a specialized cable which is designed to transmit audio while preventing noise from entering the audio stream[24]. RCA connectors are coaxial cables and it is widely used in car audio installation because its frequency can range from 0Hz up to 3GHz. Our group decided that the audio input for our converter will contain two RCA input jacks—one for pitch and one for volume.

One drawback of using RCA connectors is that they are unbalanced—they have low noise reduction when the signal is traveling through the cable at a long distance.

5.2.6 MIDI Configuration

Before understanding how to connect MIDI devices, we first need to acknowledge that in the MIDI protocol, we must differentiate which MIDI device creates the MIDI data and which receives it.

Primarily, master devices send out the data into another unit. The master device must always create MIDI information (including on/off note, MIDI note number, velocity, etc.) so that the unit receiving (called slave) can understand and carry out the information to wherever the user wants the data stream to. The slave unit may carry the information directly to a MIDI sequencer or directly to a series of other slaves.

The majority of MIDI instruments include MIDI OUT, MIDI IN and MIDI THRU. MIDI OUT is used to send out the MIDI data to a designated port, including another MIDI device (through MIDI IN port of that device) or a MIDI sequencer. By default, MIDI IN is used to receive the MIDI data from the source. The following figure demonstrates three typical scenarios connecting MIDI devices:

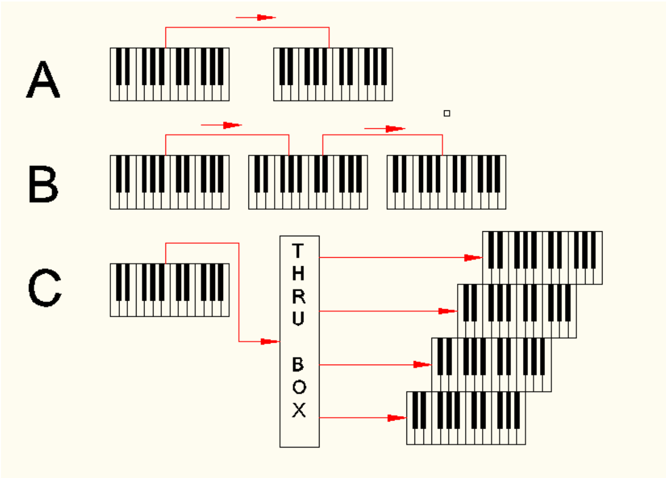

Figure 9: Reveals different type of MIDI configuration setups

In Figure 9 setup A shows that the first keyboard device is the master and the second is the slave. The MIDI cable is connected from the master's MIDI OUT to the slave's MIDI IN. The arrow represents the direction of the MIDI data flow: when a data is made from the master (pitch bend, note data, etc.) and is sent to the second keyboard, the first keyboard is controlling the second keyboard.

In setup B demonstrates one keyboard controlling two keyboards at the same time. The configuration for this setup is as so: connect the MIDI OUT from the master to the MIDI IN of slave #1. Next, connect slave #1's MIDI THRU to the MIDI IN of slave #2.

Setup C shows that even one keyboard can control many MIDI devices by simply using one port. In this example, a MIDI cable is connected from the master's MIDI OUT to the MIDI IN of the THRU box, and then the THRU box will provide several THRU ports to connect each of the slaves.

5.2.6.1 Connection PIN

MIDI's electrical and physical interface uses 5-pin DIN connectors. The following figure demonstrates what a typical 5-pin DIN looks like:

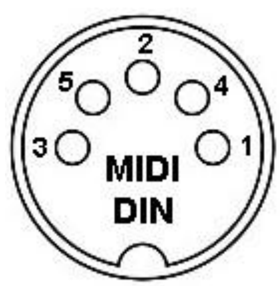

Figure 10: Standard MIDI 5-pin DIN pinout

Typically MIDI IN and OUT pin out layouts are demonstrated in the following tables:

MIDI IN Pinout	
PIN	Description
1	Not connected
2	Not connected
3	Not connected
4	Current Source
5	Current Sink

MIDI OUT Pinout	
PIN	Description
1	Not connected
2	Grounded
3	Not Connected
4	Current Sink
5	Current Source

Table 4: We used this information to design our SenseBox in the design section.

5.2.6.2 MIDI Role: Instrument vs. Controller

Over the years, MIDI has proven to be more versatile than it was originally intended because it not only it yielded a number of significant benefits to musicians, but also to recording artists, live performing artists, and hobbyist. Most often, MIDI devices can come either as a MIDI keyboard or MIDI pads as illustrated below:

Figure 11: MIDI Keyboard and MIDI PAD.

Each MIDI device can carry up to 16 different channels and each channel can be individually assigned to different tasks. In fact, MIDI does not have to be used exclusively for music purposes. One channel can be used as controlling the lighting in the room, another monitoring a car windshield while another can simply act as a video game controller. MIDI, in all essence is simply a type of electronic command. For the purpose of this project, we will be explaining two simple roles that MIDI will be used as: an instrument and a controller.

When a MIDI device is acting as an instrument, it acts as how it is noted: it will send a command to an incoming MIDI device, or a sequencer, to produce/release a note. Inside the data, it will include information such as NOTEON (tells the device to turn on a MIDI note), KEYNOTE (tells the device which notes turn on), VELOCITY (how hard the note was pressed—note sensitivity), and NOTEOFF (tells the device to turn off the current MIDI note). For instance, assume that channel 1 in the MIDI sequencer is set a saxophone preset. If the user presses and hold a note C in a MIDI keyboard, it will tell the sequencer to produce that MIDI note through channel 1, which subsequently will produce a saxophone sound, playing note C. When the user releases the note on the MIDI keyboard, it will tell the sequencer to turn off the MIDI note, which will stop producing the saxophone sound.

Although MIDI keyboards can carry the same tasks as MIDI pads, most DJs and percussionists prefer to use MIDI pads over keyboards because they offer a better navigation when controlling MIDI data. Unlike some MIDI keyboards, MIDI pads usually contain only pads, buttons, faders and knobs. One main usage of these features is that it can be assigned to control the DAW via MIDI. For instance, the user may assign knob A to controller the filter cutoff of a VST plugin, whereas fader B can be assigned to control the master volume of the mixer. Similar to a MIDI instrument, the user may assign the pads to emulate a drum kit—all of which are note sensitive. A certain pad may produce a snare of a drum kit or, if used as a MIDI controller, a pad may be used to press "record" in the DAW. As you can see, MIDI devices can interchange its role, which is all dependent on the user's usage.

5.2.7 SenseBox Power Consideration

Since our choice of the microprocessor ended up with the Atmega328p, our specified operating voltage is +5V. The SenseBox console will also include an LCD display which will also require +5V. There are several options which may be considered when designing the power supply. In conclusion, the SenseBox is powered by a 12VDC outlet, but the following sections reveals our past considerations.

SenseBox Power Option 1"Receive power from the uWave Theremin":
If the SenseBox Pitch-to-Midi converter is connected directly to the uWave Theremin, we can obtain +12V DC. We can connect the 12 volt DC power supply to a DC-to-DC converter circuit which will convert the 12 volts DC to 5 volts DC. The DC-to-DC converter circuit may be designed using a combination of one 1N5400 diode, two polarized capacitors with values of 220uF and 47uF, and a LM7805 voltage regulator to obtain the +5 volts from the +12 volts DC input.

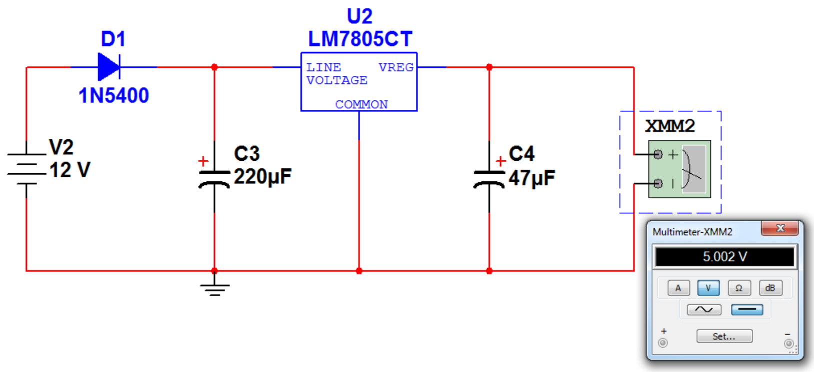

Figure 12: SenseBox Power Option 1a

Alternatively, the DC-to-DC converter circuit may also be designed using only capacitors two polarized capacitors of values 10uF and 1uF, and an LM7805 voltage regulator to obtain the +5 volts from our +12 volt DC input, as shown below.

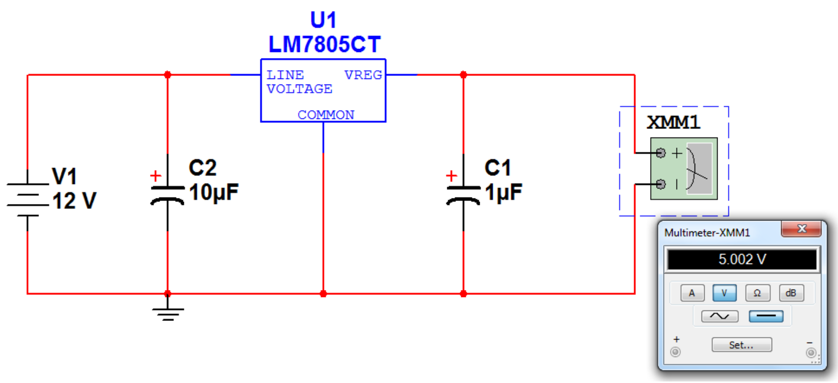

Figure 13: SenseBox Power Option 1b

Although this option to provide power from the uWave is feasible, it will require and additional cable and connection to be run between devices. For aesthetics purposes, we may want to minimize the number of cords required to connect devices together.

SenseBox Power Option 2 "Wall Adapter":
One option we may consider to obtain power for the SenseBox is to provide its own dedicated power supply. This method can be approached in two different ways, one is by wall adapter. Provide a 12 volt AC 1 ampere power adapter with a 5.5mm x 2.1mm connection. Build a AC-to-DC power supply circuit using capacitors and an LM7805 voltage regulator to convert 12 volt AC to +5 volt DC.

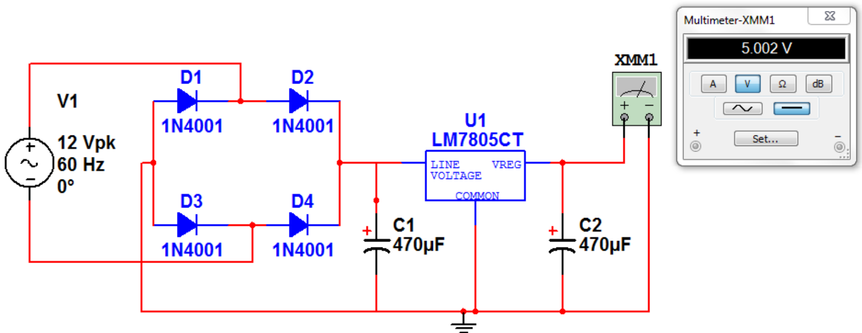

Figure 14: SenseBox Power Option 2

SenseBox Power Option 3 "Battery powered":
The second approach to providing a dedicate power source for the SenseBox is by battery power. Provide and install a 9 volt DC battery. Design and build a DC-to-DC converter circuit to step down the supply voltage of 9 volts DC to 5 volts DC. DC-to-DC converter shall use the LM7805 voltage regulator, one 470uF polarized electrolytic capacitor and one 47uF polarized electrolytic capacitor.

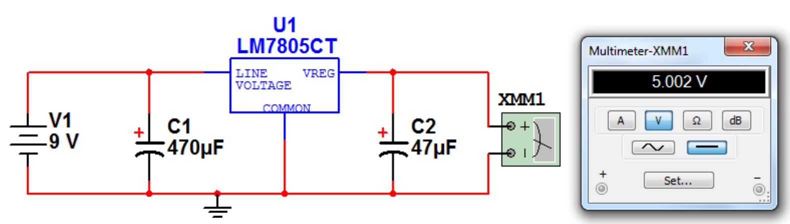

Figure 15: SenseBox Power Option 3

SenseBox Power Option 4 "USB Power":
Also, we must consider the case when we are programming or loading the microcontroller. During this process the SenseBox will be connected to the computer by USB cables. It is possible to provide a 5 volt DC supply to our electronic devices through USB connection as follows: Connect the SenseBox Pitch-to-midi converter to the computer with a USB. Use a 4-pin standard A-to-B connector. The USB has 4-pins: Pin1 Vcc(+5V), Pin2 Data-, Pin3 Data+, and Pin4 Ground. Connect Pin1 to obtain the +5 volts and Pin4 to connect to ground.

This option to provide power through USB connection may be the most logical option for this project. The SenseBox is designed and functions such that it will always be connected to a computer device to send and receive MIDI data. For this reason and the desire to minimize unnecessary cables and wires, we may find this option to be most feasible.

SenseBox Pitch-to-Midi Converter Power Option 5 "Battery and Wall Adapter":
One final option is to combine Option 3 "Battery Power" and Option 4 "USB Power". This combination of power sources could allow the SenseBox to have continuous power in case the box is disconnected from the computer. There are a couple of ways to approach this option.

Provide and install a solid-state, 3-pole relay switch to automatically switch the device from battery to wall adapter power supply. A relay is a switch that is operated electrically. There are two types of relays; electromagnetic and solid-state. The electromagnetic relays have mechanical parts which are used in the switching mechanism. The solid-state relays are able to perform switching function without the use of moving parts, but with the use of semiconductor devices. The circuit will have one common terminal which can connect to one of two poles at a time, this is commonly known as a Single Pole Double Throw (SPDT) relay switch. In this configuration there are two circuits being controlled, one normally open and one is normally-closed. When the relay is activated, the switch will automatically switch from one circuit to the other.

There are a few power management multiplexers on the market that will allow us to have two power supplies, in this case battery and USB, and when both are connected the multiplexer can transfer power seamlessly between sources. Linear Technology and Texas Instruments both manufacture power controllers which are most commonly used in PCs, digital cameras, cell phones and MP3 player where power sources switch from battery to USB when the user connects the device for charging or programming. The TPS2110A Autoswitching Power MUX by Texas Instrument can receive two DC input voltages from 2.8V-5.5V and one output. The battery source of 9V would have to first pass through a voltage regulator circuit to step down the voltage, then connect to the TPS2110A, while the USB (5V) can be connected directly.

This option is viable if it is necessary to have continuous or uninterruptable power on the SenseBox.

5.2.8 LCD Power

The SenseBox will have an LCD screen panel that is used to display the notes being played, which are gathered from the pitch and the volume levels of the analog output signal coming from the uWave Theremin. The typical LCD display requires +5V DC and a connection to ground. Because the microcontroller also requires 5V DC, the design can easily obtain power from the PCB.

The SenseBox PCB design should include an additional 5V pin so that the LCD display module will obtain its power directly from this power pin located on the PCB. The printed circuit board layout for the SenseBox Pitch-to-MIDI converter design may also include a 5 volt bus bar or any additional pin/header to supply +5 volts and ground to the LCD display module and any other peripheral devices.

5.3 Expandome

5.3.1 Hardware

Many innovative and alternative interfaces were explored by the group towards achieving our instrumental goals. The most simplistic and adaptive device was decided upon, both because of its easy-grid physique, and also the design philosophy that surrounds it. After all, the Arduinome is itself an iterative design upon the Monome. The following section covers the researched hardware to create the Expandome easy-grid device. As an iterative design process, there are many components that were initially selected for prospective parts based on the Arduinome community's experience and knowledge. After much consideration of fulfilling our goals, additional parts were selected that to enhance that design towards the adaptable interface we seek.

5.3.1.1 Button Related

5.3.1.1.1 Button Pad and Button Pad PCB

The Expandome device's most visible feature is its backlit LED grid. There is already a large community resource built around the open-source Monome project, and Flipmu's Arduinome and Chronome. SparkFun Electronics, aware of the high demand for Monome-type buttonpads, fulfilled this niche with a 4x4 buttonpad and buttonpad PCB[25] that can host typical LEDs or RGB LEDs. The 4x4 button pads made by SparkFun are *'made from translucent silicone rubber and [are] LED compatible. Each button has a conductive circle backing so that a switch can be created with exposed PCB traces. Button force is between 190 and 210 grams activation force.'*[26]

Our initial prototypes will implement standard LEDs, but as each buttonpad PCB is also compatible for both standard LED and RGB LEDs this allows for adding this RGB capability at a later time. These also allow for connecting any number of these boards together to form a *n* x *m* matrix of any 4x4 multiple dimension.

5.3.1.1.2 IN4148 Diodes

These are used in conjunction with the LEDs on the buttonpad PCBs to isolate the button-switches for easier decoding. These diodes are also commonly used at high frequencies for low-current applications and have a reverse recovery time of no more than 4ns[27]. These diodes are recommended by the Arduinome group and community and fulfill our need for the LED push-button grid.

5.3.1.2 Integrated Circuits

5.3.1.2.1 Atmel Atmega328 and 2560

The reference design that was studied from the Flipmu website used the ArduinoMEGA microcontroller board which is based on the ATmega2560 chip. The ATmega 2560 has 54 digital input/output pins, 16 analog inputs and several other features. This chip has

100 pins and is not produced in the dual in-line package, also known as the DIP. The DIP form is most desirable because this form has two rows and n number of pins and is easy to mount on a breadboard or printed circuit board for testing.

For the 8x8 grid device, the additional 256k of RAM is desirable to host the additional default programs that the primary Expandome will utilize. For the initial prototyping, because the 2560 is an SMD component, we will implement a breakout board and component set modeled on the Mega microcontroller.

The Atmel ATmega328 microcontroller has only 14 digital input/output pins and operates at 5V with 2 kB of SRAM. This microcontroller comes in the DIP form with 28 pins. The Arduino manufacturer makes a microcontroller board called the Aruduino Uno which uses this chip; this would be ideal for initial prototyping and testing. The ATmega328 has a 32 kB of flash memory to allow the capability to store programs and has 1 kB of EEPROM which can is used to store parameters.

The 328 has enough memory for all grid sizes, but not enough to host additional default programs. For the initial prototyping the ATmega328 will be used for the 4x4 and 4x8 grid devices.

Microcontroller	**ATmega328**	**ATmega2560**
Operating Voltage	5V	5V
Input Voltage	7-12V	7-12V
Input Voltage	6-20V	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)	54 (of which 15 provide PWM output)
Analog Input Pins	6	16
DCA per I/O Pin	40 mA	40mA
DCA for 3.3V Pin	50 mA	50mA
Flash Memory	32 kB (of which 0.5 KB used by bootloader)	256 kB (of which 8 kB used by bootloader)
SRAM	2 kB	8 kB
EEPROM	1 kB	4 kB

Table 5: Comparison between the two microcontroller chips that will be implemented.

Many features for the Atmel chipset functions will be modeled after the Arduino board, such as implementing an external 16MHz quartz crystal for clock speed.

5.3.1.2.2 Shift Registers and LED Driver

The 74HC164 and 74HC165 are 8-bit serial-in/parallel-out and parallel-in/serial-out shift registers, respectively. These low-to-high edge-triggered, highspeed, low-Schottky shift registers came under consideration as a recommended part by the Flipmu researchers and because they are commonly used for multiplexed grid-logic applications when used in concert.

The planned dimensions for each of the Expandome grid devices have a maximum dimension of eight on each axis, since each of the shift registers can hold a max set of 8-bits of data. Any higher dimension would require additional components. For the smaller dimensioned devices, this extra memory is of course not a hindrance.

5.3.1.2.3 LED Driver

The use of an LED display driver is necessary to interface the microprocessor with the LEDs used on the Expandome device. The driver device allows easier control over the LED grid via the shift registers and aids in analog and brightness control. It will greatly facilitate the control of the multiplexed LED grid. Each Expandome device will require at least one LED driver. There are two LED drivers which we may consider.

LED Driver Option 1 MAX7219CGB:
The MAX7219CBG, manufactured by Maxim Integrated Products, is an integrated circuit which has an 8-digit LED display driver that has the capability to provide interface between the microprocessor and up to 64 individual LEDs. This integrated circuit also gives the capability to control brightness through analog and digital control. This driver uses a SPI compatible slave interface which will allow the capability to control using only 3 of the digital output pins from our microcontroller. The MAX7219 requires +5V. For a single LED matrix, it is possible to use the +5V power supply from the microcontroller circuit, but if there are more than one LED matrixes than an external power-supply may be required.

LED Driver Option 2 TLC5940:
The multiplexed TLC5940 is a 16 channel PWM constant current LED driver. This LED driver provides constant current set by one resistor which minimizes the amount of resisters required.

There is a reference design called the *Mini Monome*[28] which used RGB LEDs. The LED driver which they used was the TLC5940 by Texas Instruments. This Driver is capable of controlling 16 LEDs at the same time, and in reference project the driver is used to control 64 LEDs on an 8x8 board. The LEDs were broken down into groups of 16 (4x4 squares) therefor using a total of four TLC5940 LED Drivers to control all 64 LEDs.

Comparatively the MAX7219 driver was more appealing, because it was tailored to fit in with the multiplexing of the grid switches and only one component was required instead of four for an 8x8 device.

5.3.1.3 Passive Components

5.3.1.3.1 LEDs and Rset Resistor

With needing to go forward with prototyping we set some initial limiting factors based on the community's wiki on the Monome site[29] that proposed 3mm, 500mcd - 2000mcd, about 50° or better viewing angle, below 4.5V forward voltage and below 20mA current draw.

The MAX7219 LED Driver was chosen, as it was a proven precedent chosen by the Arduinome developers[4] and is capable of driving a fully lit 8x8 LED matrix. Garnered from the Arduino website, and the LED and LED Driver datasheets' *DC Forward Current* and *Forward Voltage*, *Rset* was chosen.

When calculating the maximum current and voltage draw of the device, only one row of the LED matrix needs to be calculated in the total. This is because the MAX7219 multiplexes the rows of the LED matrix at about 800 times a second[30]. This is fast enough to give the illusion that all rows are lit at the same time. The maximum current is easy to calculate.

PeakCurrent = (8 x LedCurrent) + MAX7219-Supply
PeakCurrent = (8 x 20mA) + 10mA = 170mA

The 8x8 Expandome grid device will be largest consumer of power. The primary device connected to the computer will draw its power a maxim of 170mA. It is an estimate[30] that the maximum supplied current a typical USB port gives on a computer is approximately 500mA. If we allow an additional 40mA for the ATmega and 50mA for each shift register, then a total current draw could be as high as 310mA, although this is an estimate that will have to be tested when the prototype device has been constructed.

Additionally, our choice of supplying the smaller grid devices with power with a battery power allots for the estimate from the same source that 32 out of 64 LEDs are lit, consume a static current of only 78mA and allowed the matrix with an Arduino to be powered for 55 minutes.

It is a consideration to minimize the current draw if necessary by setting the *Rset* Resistor to help lower the total cascading instrument draw for 3 Expandome to below the estimated 500mA threshold, an observed[30] typical computer USB output. Lowering the current draw can be done with a physical or logical switch that changes the *Rset* to a higher resistance when the device is under battery power. This switch may be integrated into the power switching circuitry.

Rset is chosen based on the LEDs' *DC Forward Current* ('the maximum current that is allowed to go through the LED without damaging it in the long run').[30,31] This limits the total current draw of all LEDs in the *n* x *m* grid. For grids with *n* less than *m*, the lesser *n* row is chosen to be lit due to multiplexing, which results in half the current draw due to LEDs. For example, in a 4x8 grid, if the row of 4 LEDs is only lit, this is of course half the draw that 8 LEDs at a time. Multiplexing allows for the rapid lighting of only a row at a time to give the illusion that the entire grid is lit[5]. This works because the human eye

[5] Often called 'persistence of vision'.

can only pick up 10 to 12 separate frames per second[32], this is much less than the LED driver can perform at, which is approximately 800 times a second.

5.3.1.4 Input and Output Components

5.3.1.4.1 MIDI and USB

The primary connector component on the Arduinome is a USB type B female connector that supplies the device connection to the computer and power. One of the most apparent improvements to the device is that the Expandome will have two pair of MIDI connectors, on opposite sides of the enclosure. This will allow multiple devices to be connected in cascade. A 5-pin DIN MIDI gives us an extra 2 pins on the MIDI connector, which we will allow us to pass power and maintain ground.

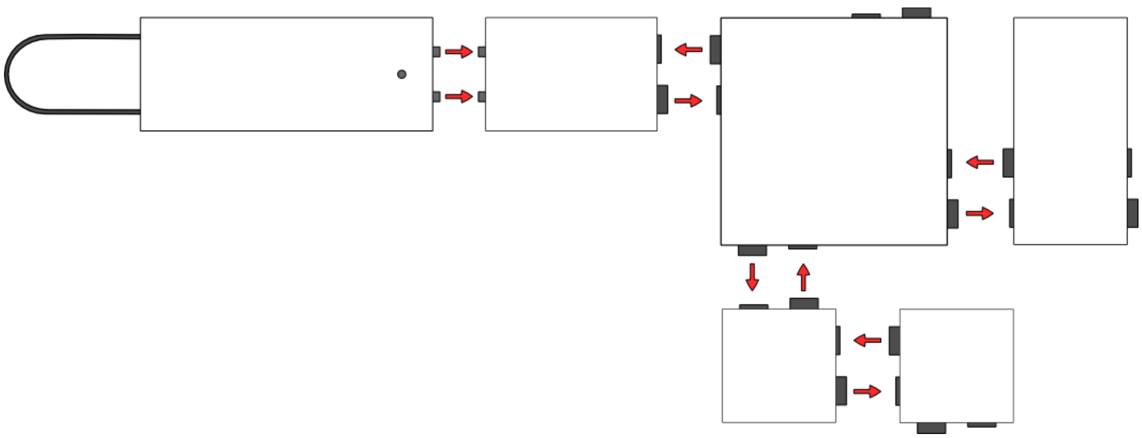

Figure 16: Expandome Device depicting interconnects

The 8x8 grid will have one USB type-B with two pair MIDI (In/Out per pair). All smaller grids will have only two pair of MIDI, no USB. All initial prototypes however have a USB port.

5.3.1.4.2 Power Connections

Considerations for power connectors include USB, MIDI, 2.1mm jack and other common connections. A primary consideration is to minimalize cabling and utilize the unused pins available on the USB and MIDI cables for simplicity.

Power for the 8x8 grid devices can be passed via the USB connector, if the primary connected device to a computer. If an 8x8 is in cascade, then its power can be passed via the extra MIDI lines.

For the smaller grid devices, 4x8 and 4x4, power may be supplied by battery if in a standalone implementation, or will be passed via the extra MIDI lines if in a cascade setup.

The Arduino Uno microcontroller board comes standard with a USB Female Type B connector which allows the board to be powered by a USB Cable A to B Male/Male type peripheral cable connection which can provide +5V. The plug at each end has 4 pins. Pin

1 is for Vcc(+5V), Pin 2 and Pin 3 are for Data – and Data +, and Pin 4 is for ground. We may use this same connection in the Expandome design, as was with the Arduinome; this will allow us to program the microcontroller and provide power through USB.

The passing of power from one Expandome to another will be done through the two additional lines in the Expandome's MIDI-in and out cables. It is important to have a common ground between devices. When considering these additional MIDI pins, each connector comes in a pair, one in, one out. This gives a total of 5 free pins, of which three will be utilized as data, power, and ground.

5.3.1.4.3 SD Card

The need to implement an SD card as additional storage stems from the wish to provide a quickly exportable data architecture that will store beat sound files (for standalone playing), can capture performance in the form of MIDI data messages (for later playback or editing), and allow for default programs for computer interaction that would otherwise not be possible due to memory constraints.

The Expandome modules will be designed to have default programming installed which will allow the user to be able to plug in the module and play without external computer programs. A Secure Digital memory card is most commonly known as the SD Card. The SD card memory is non-volatile, which means that it can retain information even without power. We may use this memory storage type to implement the default or additional programming.

There are three different form factors and card families which we may consider for this project. The form factor refers to the size of the card, while the card family refers to the capacity type of the card. The card family choices are: Standard-Capacity (SDSC) 1MB to 2GB, the High-Capacity (SDHC) 4GB to 32GB, the eXtended-Capacity (SDXC) greater than or equal to 32GB to 2TB, and the SDIO, which combines input/output functions with data storage. The three form factor sizes are standard, mini, and micro. To save space on our printed circuit board and in the enclosure we may choose to use the smallest of the three forms "micro". Also, the data files that we will create for the default programs and settings will not require more than 2GB of memory therefore we may select the Standard Capacity SDSC card.

The SD cards typically use 3.3 volts, with the exception to the High Capacity and Extended Capacity cards which may be switched to operating at 1.8 volts if programmed. The power requirements for the microSD cards range from 66-330mW with 100mA at 3.3 volts supply. The SD cards are all block addressable storage devices which can be used for read or write functions.

5.3.2 Software

As participants in a community of design we gravitate towards open source software. This is because it allows maximum participation in innovative discussion and developments to all interested. All software developed and implemented in this project shall be open source. See License.

5.3.2.1 Data Transfer Considerations

5.3.2.1.1 Between Devices

The Expandome devices will have two pair of MIDI connectors. Of each pair of 5-pin DINs, there will be a total of 5 extra pins available to utilize. At the time of writing, 3 pins are required for power, data, and ground. The data pins will be implemented to pass the non-MIDI data messages between the devices. An I^2C chip will be utilized to manage this inter-device communication.

The Expandome will have a unique design feature which allows us to expand the base 8x8 musical array device by adding other Expandome devices such as a 4x4 or 4x8. For this feature to function properly, it is necessary for each device to communicate with each other over the MIDI interconnects. Research on how to properly implement this function shows that we may use either an Inter-Integrated Circuit (I^2C) or a Serial Peripheral Interface (SPI) microcontroller.

According to Arduino Learning website[13], it is possible to use the Inter-Integrated Circuit (I^2C) to communicate between two Atmel microcontrollers. The I^2C allows the devices to perform in a Master/Slave type configuration. The Arduino "Wire Library"[33] is available for use with the I^2C.

There are only two wired connections required to communicate between the two devices. One shall be connected to the serial clock pin and one to the serial data pin. It is important to assign each device a unique address so that communication with multiple devices is possible.

According to the Arduino literature on connecting multiple microcontrollers, it is also important to make sure that the devices share a common ground. The intended implementation is going to be over MIDI 5-pin DIN cables. Since the proposed design has two MIDI connectors per side, one input, one output; and since the total extra pins are three and two per pair, the ground component will be amongst the three (five total available) pins assigned to power, data, and ground.

The Serial Peripheral Interface (SPI) microcontroller may also be used as a Master and Slave Interface which allow communication between two or more microcontrollers. When using the devices that have an SPI microcontroller, the SPI library will be used and is provided by the Arduino Environment site[13]. SPI is a synchronous serial data protocol. The communication between multiple SPI devices requires three lines of communication: Master In Slave Out (MISO), Master Out Slave In (MOSI), and Serial Clock (SCK). When developing the code for the use of an SPI device, it is necessary to identify many specifics, such as how the data is being shifted (MSB or LSB), data clock status when high or low, the speed that the SPI is running at and many other specific settings.

According to the Arduino Learning website, the SPI device can be implemented in many different ways which indicates that it is not standardized and can be more difficult to use.

Between the I^2C and SPI transfer considerations, the I2C appears to more fit for this project's needs and will be tested in further prototyping.

5.3.2.1.2 Compatibility, Device Recognition and Firmware

A new serial number needs to be flashed onto the FTDI USB-to-serial chip for Monome-type applications to speak to the device. This is because an OSC serial protocol is implemented, such as SerialOSC or ArduinomeSerial[34], that automatically detects devices connected and handles the device's data flow over the network. These protocols only handle dataflow for devices with a specific naming convention. Any variation of 'a40h-xxx', or 'chr-xxx' set as a static serial number, will do. Future work will include additional handling of devices with the 'exp-xxx' enumeration.

The primary Expandome routine will be flashed onto the ATmega microcontroller via FTDI or by flashing the routine onto a chip with an Arduino and moving it to the DIP socket on the board. All Expandome grid devices will have the same main routine, except that the 8x8 grid devices will have additional SD card and USB functions. Additional programs for the uWave and SenseBox devices will be hosted on the SD card.

5.3.2.2 Data Structure

The non-MIDI data passed between the instruments will use the extra pins available on the MIDI connectors. These will be managed by the I^2C chips. The primary routine flashed onto each ATmega chip for each Expandome will be developed to be general enough between instruments, to manage data transfer whether any grid size.

The data structure implemented will utilize multiple environments that are tailored towards differing themes of function. The required functionalities include storing, organizing and parsing data to-and-from devices and computer alike. These will involve the collection, storage, and retrieval of data between these multiple source; uWave, SenseBox and multiple Expandome devices. This project will include a logically assigned data structure to aid in accurate messages that are simplistic and efficient, yet informative. Prefixes and other acronyms will aid in this parsing, but first we must consider the environmental components that will be utilized.

5.3.2.3 Code (sorted by environment)

The Expandome, as well as the SenseBox, will require the use of a development environment, and support software which will allow the user to program the microcontroller in each device. A unique feature which will be developed in conjunction with each device is the user interface modules. Each device will have its own interactive window which allows control, yet there is a lot going on in the background.

5.3.2.3.1 Arduino Environment

Firstly, all primary routines are hosted on the ATmega chips. These will use the Arduino environment[35]. It is an open-source software which means that it is free and the source code is included and may be modified by anyone. The Arduino environment was used in the reference design in conjunction with an Arduino Board. The Expandome design will include the Atmel microprocessor chip which as demonstrated in the reference design, can be programmed using the Arduino environment. The boot loaded Arduino environment has the capability to program and run the Expandome as a stand-alone musical device or it has the capability to communicate directly with software on the computer. The language that will be used is Arduino, which is based primarily on the C/C++ language. The development environment that will be used is the Arduino environment.

There are several resources which provide tutorials and sample programs to allow the user to explore the Arduino environment and capabilities and familiarize one with the data structures and Arduino language. These resources will be utilized prior to implementing the software portion of this project in order to have a good understanding of the software and code. In the community of design and hobbyists, the Arduino Environment is widely used to implement programming logic on the microcontrollers for various applications. It is easy to use and there is a plethora of resources available.

The initial reference design used for the Expandome includes a reference code named "ArduinomeFirmware3 3a," revised 01/21/2012 by Jordan Hochenbaum. This code is listed as free software and may be modified under the terms of the GNU General Public License as published by the Free Software Foundation. This source code will be referenced as a basis for the Expandome code structure but through the course of development has drastically deviated to include the unique features of the Expandome such as different array sizes, pre-programmable stand-alone feature, capability of expanding the array by connecting multiple devices is series, and the capability to interact with the uWave Theremin and SenseBox. Future work will explore other microcontrollers such as the PIC18F family, which CNMAT has tested and developed the uOSC firmware to directly implement the OSC protocol[36].

5.3.2.3.2 Löve Lua

The Löve Lua environment is a 2D game library written for Lua[37]. This environment was selected for its rich visual capabilities that can be implemented with ease and without overdue processing. The Lua language is extremely flexible and is an appealing candidate to integrate with both OSC protocols, Pure Data (PD), and Arduino environment. At the time of writing, there is little on this extensive of integration between all of these environments and this adds further appeal as to something innovative to work on. The Löve Lua environment is not as 'direct' as say using Processing to create a visual interface, and is not as easily flashed onto an ATmega chip, but this is okay, because its real advantage comes in the way it can be used to handle file structures. Löve Lua is not a compiled language, but in fact runs from zipped file folders that are renamed from .zip to .love. The EMI Browser will be hosted on the external SD card which will be implemented for quickloading of beat files, as well as very quick programming edits. By not needing to compile a program before running, time is saved and this adds an enticing feature to the project by being both practical and novel.

5.3.2.3.3 Processing

Processing is a graphical programming environment that is both minimal and very visually capable. It is a promising candidate for future work for flashing directly onto ATmega chips. It was the original candidate to create the visual functions of the Expandome browser (in fact we are still pursuing development of the Expandome Browser in Processing), but is trumped by Löve Lua in that it must be compiled before running. Additionally, if hosted on an ATmega, it would not be as readily edited and re-flashed as a Löve Lua interface swapped onto an SD card. Additionally explored, was VVVV[38], which is similar, but capable of more robust media and physics. Interfaces utilizing VVVV will be explored in future work.

5.3.2.3.4 Pure Data

Pure Data is a programming environment that we will be highly utilizing for MIDI communication between the devices and the interface. It is largely implemented as a visual way of programming sound synthesizer type of programs. We will be using it as a middleman between the visual languages and the data languages, as well as utilizing it as an interface by its own. It is also an appealing choice because of its minimalist memory footprint. On one additional note for the future, it will allow many features to be added such as wireless communication to the devices, because it is very capable of mitigating between ports and protocols on the computer for any use. In its most minimal usage, it will implement a router to handle OSC and MIDI transfer between both our physical SenseBox and Expandome devices and emulators and apps, such as Androidome[39], TouchOSC[40], and Control[41].

5.3.2.3.5 OSC

Open Sound Control (OSC)[54] was initially described as a "new protocol for communication among computers, sound synthesizers, and other multimedia devices that is optimized for modern networking technology. Entities within a system are addressed individually by an open-ended URL-style symbolic naming scheme that includes a powerful pattern matching language to specify multiple recipients of a single message. We provide high resolution time tags and a mechanism for specifying groups of messages whose effects are to occur simultaneously[42,43]". This strongly timed protocol is an ideal candidate for device recognition and already is tailored towards being used towards modular devices. It implements already existing networks to deliver data, and in conjunction with Pure Data, this information can be translated to MIDI, or MIDI-to-OSC with ease.

5.3.2.4 Digital Audio Workstation (DAW)

Digital Workstation Audio, or DAW, is a computer-based electronic system designed for recording, editing and playing back digital audio. In essence, DAW is used as a virtual recording studio. Contemporary DAWs are now commonly found in recording software that interfaces with audio and MIDI hardware. Depending on what kind of DAW the user is using, DAW can support editing audio tracks and MIDI tracks, as well video editing. However, keep in mind that the performance of the DAW exclusively depends on the computer specification—the higher CPU and RAM memory it has, the more functional the DAW will execute. As an illustration, with higher CPU the DAW will be more capable to record multiple tracks at the same time, as well as using more plugin effects.

In today's market, there are wide ranges of DAW software to choose from—free software's like Garageband or Audacity are primarily used by novice users, whereas high-end software like Steinberg's Cubase or Logic Pro are primarily used by professional producers. What differentiates from other DAW software is based on the features that a DAW has. It is important to note that simply having more features in the DAW does not automatically mean that the DAW is superior from others. When choosing the right DAW, the user should consider only the key features that they will primarily utilize. The lists below are some important considerations:

- Does it have an audio sequencer?
- Does it have a MIDI sequencer?
- Does it support VST plugins?
- Retail Price?

5.3.2.4.1 Audio Sequencer

The audio sequencer deals with mixing and editing audio tracks. In a typical audio sequencer, it includes a virtual mixer that allows the user to edit and master audio tracks. One of its functionalities involves routing—it allows the user to route the signals to internal buses and add effects such as reverb, delay, or any other built-in audio effects.

Note that the quality of the sound is not solely determined by DAW, but is also dependent on the quality of the audio source.

5.3.2.4.2 MIDI Sequencer

MIDI Sequencer is a software application that handles MIDI events along a timing grid. Typically, the MIDI sequencer is the core of manipulating and coordinating MIDI - it can record, playback, and edit MIDI information. Furthermore, the sequencer ultimately decides what exactly the MIDI information can be used for. Since MIDI is simply an electronic command, it can be used to assign a play note from a sound module (either hardware or software) or assign short-cut commands in the software.

5.3.2.4.3 Virtual Studio Technology (VST)

Virtual Studio Technology (VST), licensed and freely distributed by Steinberg, is a software sound module that interfaces with audio and MIDI tracks in the DAW. Unlike using a hardware sound module, VST offers plug-ins called VSTi (Virtual Studio Technology Instruments), which widely opens limitless parameters and sound sources for use. For instance, if the user assigns a MIDI event to a violin VSTi, the MIDI sequencer will tell the VSTi to play accordingly to the MIDI event and consequently, it will simulate a traditional violin recording. For audio applications, VST allows the user to manipulate the output of the sound with effects such as reverb, delay, or other sound modules. It does not permanently affect the input of the audio tracks.

5.3.2.4.4 Retail Price

To reiterate, there are wide ranges of DAW to choose from. However, to choose the right DAW solely depends on the purpose of using it. Is it going to be used for amateur recording? Or is it going to be used to record a live band in your mother's garage? Or is it going to be used in a professional studio? The retail price of a DAW reflects on these considerations.

The table below shows some DAW brands that we have considered using for our project. It lists the capabilities the software has, as well today's retail price - shaded area indicates the feature is included.

Brand	**Audio Sequencer**	**MIDI Sequencer**	**VST plugin**	**Retail Price**
Steinberg's Cubase 6	(shaded)	(shaded)	(shaded)	$500
Audacity	(shaded)			Free
Reaper	(shaded)	(shaded)	(shaded)	$60
Propellerheads's Reason Essential	(shaded)	(shaded)	(shaded)	$100
Logic Pro	(shaded)	(shaded)	(shaded)	$200
Kristal	(shaded)			Free

Table 6: Digital Audio Workstation brands

For our design, we have chosen Reaper as our DAW to test with because it includes the features that we will utilize. In addition, the cost of Reaper made a valuable candidate as our DAW.

5.3.3 Enclosure

Our goals for the enclosure were to select materials that were lightweight, cheap, easily cut/shaped, and won't be EMF-obtrusive to the theremin device. The materials we chose that fit these requirements were acrylic and wood. For the initial prototypes acrylic was chosen to be the sole enclosure material, except for the smaller gridded Expandomes which use wood.

Finger (or comb) joints were selected because they were easily cut and are often used in wood joinery for making boxes. Dovetail joints have greater strength; after the initial prototype we may consider this if we have access to a laser cutter.

All devices shall have the same height. The Expandome's width and length are determined by the tightest fit that the buttonpad PCBs allow. Dimensions, drawings, and applicable calculations are in the Expandome's design section below.

5.3.4 Power Supply Considerations

The Expandome requires the use pf a microprocessor which typically requires +5 volts DC and connection to ground. The Expandome will have from 16 LEDs on the 4x4 device, 32 LEDs on the 8x4 device and 64 LEDs on the 8x8 device which will also require power.

The Arduino Boards were the most popular microprocessor devices used when researching the various designs and iterations of the Arduinome and Chronome. Much like the reference designs, the Expandome design will utilize the Atmel ATmega microcontroller, similar to the ones used found on the Arduino Boards. The ATmega microprocessor selected for this project requires +5 volts DC and connection to ground. The Expandome will also have from 16 LEDs on the 4x4 device, 32 LEDs on the 8x4 device and 64 LEDs on the 8x8 device which will also require power. The power design must consider all power requirements and the possibility of connecting multiple devices and the possibility of passing power from one device to another. The design shall consider the following possible power designs:

Expandome Power Option 1 "USB":
Each Expandome device must be programmed or uploaded via USB connection to the computer, as previously discussed, the USB can be used to provide a +5 volt DC supply to our electronic devices. This can be accomplished as follows:

Connect the Expandome to the computer with a USB cable. Use a 4-pin USB cable with standard A-to-B connectors. The USB has 4-pins: Pin-1 Vcc (+5V), Pin-2 Data-, Pin-3 Data+, and Pin-4 Ground. Connect Pin-1 to obtain the +5 volts and Pin-4 to connect to ground. The Expandome printed circuit board shall have a B type female USB connection. This will allow us to power the LEDs and the microprocessor from the USB port on the computer. There are different types of USB hubs which have varying limits of how much current can be drawn. There are two types of USB hubs that we may consider.

The root hub
The root hub is located on the computer and can provide up to 500mA when the computer is plugged into an AC power supply. If the computer is running off battery power, such as a laptop, it will limit the current to 100mA[30].

External self-powered hub
The external self-powered hub has its own power supply. This device will provide 500mA by USB. This device can sometimes provide power to the device even when the computer is off.

In this option we need to make sure that we are providing at least 500mA to the main Expandome so that there is adequate power for the microcontroller, 64 LEDs and any additional devices that may be connected and designed to receive power from the main device.

Expandome Power Option 2 "Battery Powered":

The smaller Expandome devices should be capable of being portable if powered by a 9V battery. This type of power would be an ideal use for the 4x4 and 4x8 Expandome which will be connected to the primary 8x8 Expandome board. The 9V battery can be connected by using a snap connector and a 2.1mm power plug. Design and build a DC-to-DC converter circuit to step down the supply voltage of 9 volts DC to 5 volts DC. The DC-to-DC converter shall use the LM7805 voltage regulator, one 470uF polarized electrolytic capacitor and one 47uF polarized electrolytic capacitor.

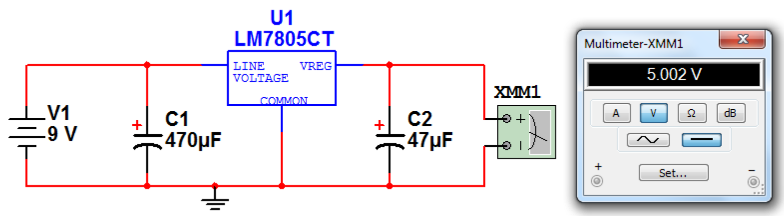

Figure 17: Expandome Power Option 2

When considering using battery power for 64 LEDs, it is necessary to estimate how long the battery will last. It is difficult to estimate an exact time so when considering battery life we shall take into account in our programming and estimates that not every LED will be lit at one time, however for simplicity in calculation we will assume maximum consumption. We may accomplish reduced current draw by using a logical or physical switch, which reduces the brightness of the LEDs to a minimum. This is done by having two *Rset* values that can be set such that the display is not as bright, therefore reducing the amount of current going through the LEDs when under battery power.

Expandome Power Option 3 "Wall adapter with AC-to-DC converter":
As discussed previously, it may be desirable to provide the main 8x8 Expandome with a dedicated power source so that it may have enough power for its own functions and enough left to power additional devices when multiple devices are connected. This can be accomplished as follows: Provide a 12 volt AC 1 ampere power adapter with a 5.5mm x 2.1mm connection. Build an AC-to-DC power supply circuit using capacitors and an LM7805 voltage regulator to convert 9 to 12 volt AC to +5 volt DC.

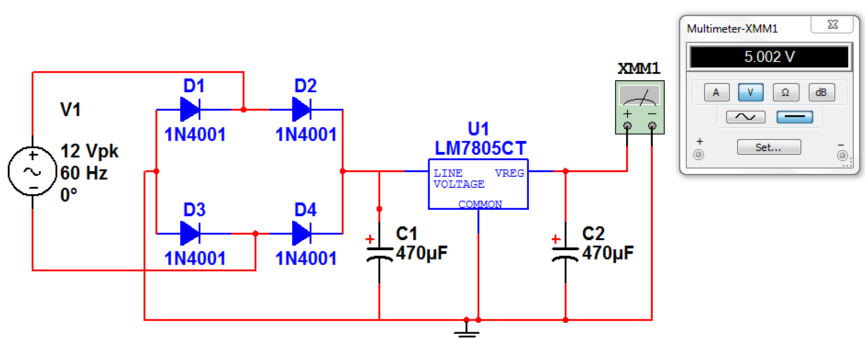

Figure 18: Expandome Power Option 3

A 9VDC, 1A wall adapter with a 5.5x2.1mm barrel jack with center-positive may be used to supply a regulated 9VDC. If used, it will only be necessary to connect this to a DC-to-DC converter circuit to drop the voltage to 5 volts.

Expandome Power Option 4 "USB and Battery Power with automatic selector":
Similar to Option 5 for the SenseBox, the option to have 2 power sources with an automatic selector may be used. This design would combine Expandome Power Options 1 and 2 into the AutoSwitching Power Mux TPS2110A. This option would be ideal for the smaller Expandome devices which will be connected to the computer by USB when programmed and can run on battery when connected with other devices.

Expandome Power Option 5 "Power through MIDI":
Our final power option to be considered is transferring power through MIDI connection. The MIDI data connection uses a 5-pin DIN connector. Only three of the five pins are used to transfer data. The remaining two pins may be used to supply power; +5V and ground. To minimize the number of cords, we may choose to provide power to the 4x4 and 4x8 Expandome through the MIDI connection. This option may also be used as the second power source using batter or USB in conjunction with the TPS2110A Power Mux, as mentioned in Option 4.

5.3.4.1 USB and MIDI Power

The Arduino prototype board comes standard with a USB Female Type B connector which allows the board to be powered by a USB Cable A to B Male/Male type peripheral cable connection which can provide +5V. The plug at each end has 4 pins. Pin 1 is for Vcc(+5V), Pin 2 and Pin 3 are for Data(–) and Data(+), and Pin 4 is for ground. When designing the power and data supply and transfer scheme for the Expandome, we will base our power use on the same scheme as used with the Arduino prototype board, because there is a large resource in the community of hobbyists on this topic.

This primary USB power is intended for when the instrument is directly connected to a USB power source. If connected in cascade to another Expandome, its power shall be drawn through the extra MIDI lines in the MIDI-in or MIDI-out cables.

SECTION 6.0 DESIGN

6.1 uWave Theremin

For this project, the group decided to use the uWave Theremin implementation given by Robert Moog in his DIY paper. It was noted that there are two versions of his design: the first, which was created in 1996, and a more recent version, which was updated in 2003 will be considered. These two versions are very similar with only a few component differences in the power supply design and in the pitch tuning implementation. Also, the 2003 paper has more details that will be of use to our design of the uWave Theremin, therefore the group will implement the 1996 version and use the extra details given in the 2003 paper to improve on the design.

6.1.1 Meeting Specifications and Requirements

According to the requirements given for the uWave Theremin, the uWave Theremin was designed to utilize two antennas, a pitch and volume antenna, and it is operated with a positive and negative 12 volt supply. Its pitch antenna and volume antenna together with their respective circuitry form a resonant frequency of 260 kHz and 450 kHz, respectively. Its audio output has frequencies in the range of 0 – 3 kHz. The uWave Theremin also provides a means by which the player can adjust the tone to fit the needs of the performance. As mentioned in the previous section, Robert Moog's design closely matches our requirements thus making it a good candidate as a reference design.

6.1.2 Antennas Circuits

The antenna type that was used for both the pitch and volume is a soft copper tube. In this section we will discuss the design of the pitch and volume antenna circuits and how they will be used to meet specifications, which states that they are to create a resonant frequency of 260 kHz and 450 kHz.

6.1.2.1 Connections Considerations

The pitch antenna is to be connected vertically to the side of the enclosure, while the volume antenna was mounted horizontally on the side of the enclosure. In order to do this, a suitable way to connect the antennas was investigated. The group decided to use a connector that allows the antennas to be easily removed and replaced. The connector of choice can be seen below in Figure 19.

Figure 19: 3/8’’ connector used to connect antennas to theremin

6.1.2.2 Antenna Dimensions

According to Robert Moog's design, in order to create the antenna circuitry that will form the resonant frequency given in the project specification, the pitch antenna should be about 18 – 19 inches long and the volume antenna should have a total length of 9 inches after bending has taken place. A hacksaw was used to cut the antennas to the correct length as these dimensions are critical in creating the right resonant frequency as illustrated by Equation 1 and Equation 2 given in the uWave Theremin research section.

Since the type of antenna the group used is a soft copper antenna, bending of the volume antenna was done with care. It has been proven that bending by hand slowly is a good technique to use because it reduces the risk of leaving unwanted kinks on the antenna therefore, this is the method of choice used to bend the volume antenna. Other researched methods for bending the volume antenna include putting sand in the tube before bending.

6.1.2.3 Pitch Antenna Circuit

The pitch antenna circuit was designed in such a way that it produced an approximate resonant frequency of 260 kHz. Knowing that the antennas basically act as a capacitor, a series connection with inductor coils was used to achieve the resonant frequency. According to the resonance equation, $f_o = \frac{1}{\sqrt{LC}}$, in order to get a resonant frequency of 260 kHz using inductors of value equal to 40 mH, a capacitance of 9.4 pF is needed (theoretically). According to the calculation done in the research section using Equation 1 and the dimension of the pitch antenna, the capacitance of the pitch antenna is equal to about 15 pF, which is a true estimate for antenna capacitance. Obviously this value does not give the desired resonant frequency. This is because Moog has taken into account the stray capacitance that affects the total capacitance when choosing the antenna length. The pitch antenna circuit design is shown below.

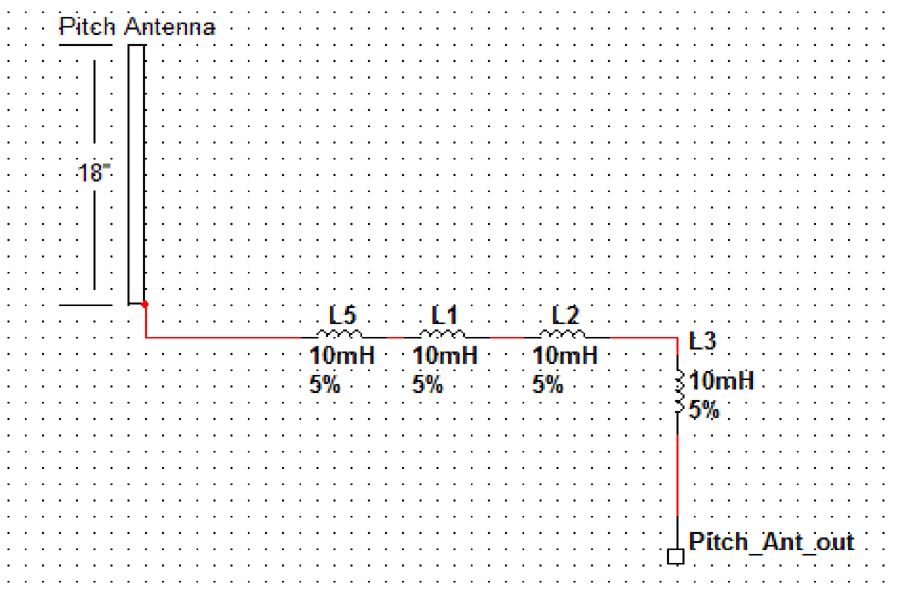

Figure 20: Pitch antenna circuit design

As mentioned earlier in the research portion, the antenna was cascaded with inductor coils to account for its relatively short length of 18 inches. We chose to use four 10 mH RIF choke inductor coils in series to create the 40 mH inductance need for the resonant frequency. Four 10mH inductors as opposed to one 40mH were used in order to increase linearity in the antenna behavior.

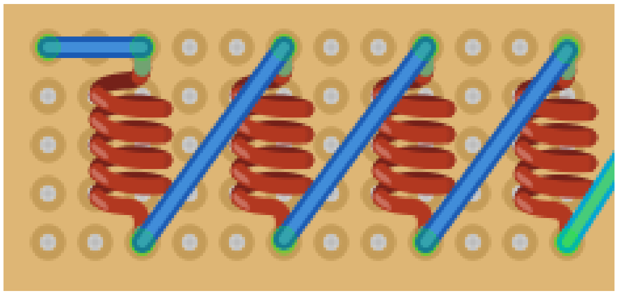

Figure 21: The pitch antenna inductors was connected in this fashion, one inch from each other

According to Moog's DIY paper, the antenna with its inductors was constructed on a separate board and mounted close to the other circuits. Robert Moog and other theremin forum members at thereminworld.com advise that these inductors should be mounted on boards “with little or no copper circuit pattern” in order to reduce unwanted capacitance seen at the antenna. Before mounting the pitch antenna, we made sure the antenna was as straight as possible by using one of the techniques mentioned in the research portion of this paper.

The pitch antenna was mounted vertically at a 90^{o} angle on the right hand side of the enclosure. The inductors are positioned “so they are parallel to one another and about one inch apart, center to center”[8]. The orientation of the inductors can be seen in Figure 21. The pitch antenna board was positioned close to the socket which will enable us to connect the pitch antenna to it using the connector in Figure 19.

6.1.2.4 Volume Antenna Circuit

Deriving the circuit to achieve the resonant frequency of 450 kHz is much similar to that of the pitch antenna though the volume antenna has a higher length. We also used four inductors in series with the volume antenna to achieve the resonant frequency as seen below in Figure 22.

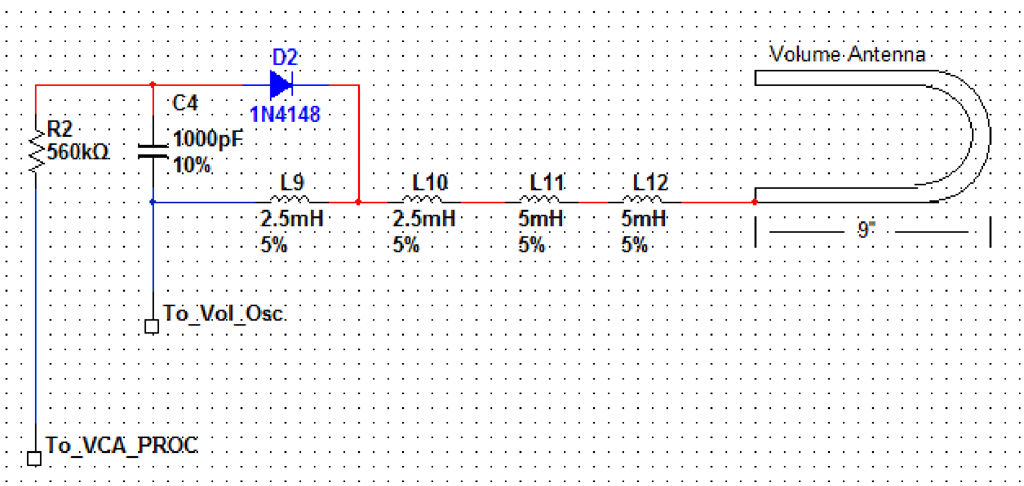

Figure 22: Volume antenna circuit design

As noted from Figure 22 the volume antenna circuitry is quite different from the pitch antenna circuit. Earlier in the research section it was mentioned that the volume antenna shall produce a control DC voltage/current responsible for determining the amplification of the pitch signal coming from the detector. The diode, resistor and capacitance addition form the half-wave rectifier/filter combo needed to convert the oscillating signal from the volume oscillator into said DC voltage/current. This DC voltage was used to determine the frequency of the volume oscillator by tuning it (VO) to give a voltage that ranges from 11V to -11V.

We used the combination of hand bending and a tube bender to bend the volume antenna. The volume antenna was bent so that "the ends will be separated by 3¼ inches, center to center". The volume antenna has a total length of nine inches[8].

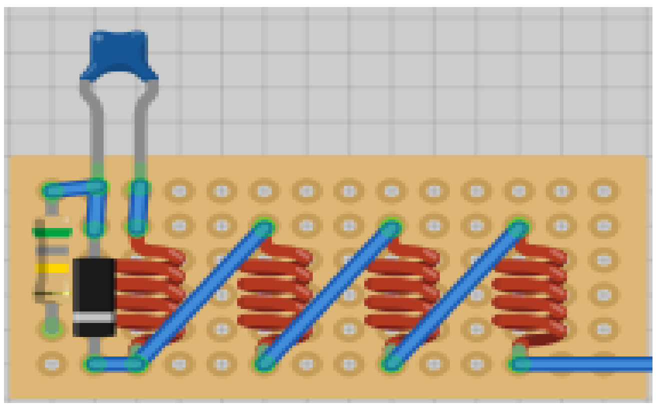

Figure 23: Example of how the inductors in the volume antenna circuit are connected

As far as inductors go, the same antenna topology applies for the volume antenna circuitry, but this time we used two 5 mH inductors and two 2.5 mH inductors in series with each other to form a total inductance of 20 mH. It is advised that this circuit be mounted on a separate board with little or no copper tracing. The inductors should be mounted one inch apart from each other as seen above in Figure 23.

6.1.3 Oscillator Circuits

The uWave Theremin uses the heterodyning technique to create sound. Meaning it uses two oscillators, one fixed and the other a varying kind, and subtracts the difference in the two oscillator frequency to create a pitch sound with frequencies ranging from 0 – 3 kHz. These oscillators are called the fixed pitch oscillator (FPO) and the variable pitch oscillator (VPO). The uWave Theremin also utilizes an additional oscillator to create the variation in volume or amplification. This oscillator is known simply as the volume oscillator. The design of these oscillators will be the focus of this section.

6.1.3.1 Fixed Pitch Oscillator (FPO)

In our research, it was noted that most theremin oscillators are initially set to oscillate at a frequency slightly higher than the resonant frequency of the antenna circuits. Our pitch antenna circuit was designed to create a resonant frequency of 260 kHz, therefore we will design the FPO to oscillate slightly higher frequency of 285kHz. The circuit is to be comprised of normal components like resistors, capacitors, transistors and a variable inductor, which can be easily found at Digi-Key.

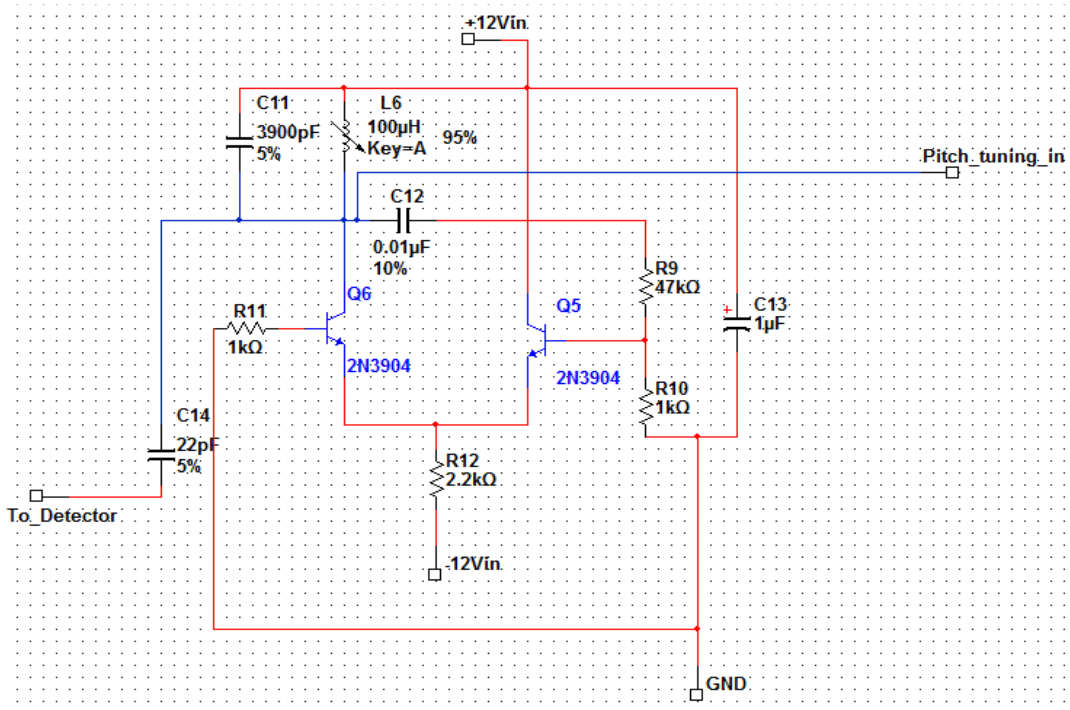

Figure 24: Schematic design of the Fixed Pitch Oscillators

The schematic seen in Figure 24 shows the basic design for the FPO. After construction, the FPO was set to oscillate at a frequency of about 285 kHz by turning the adjustable inductor labeled L6 in Figure 24.

Since the documentation of Robert Moog's theremin design was done in 1996, a few component were hard to find in the market as they have become obsolete, therefore we had to find substitutes. The adjustable inductor, L6 is one of those components. In Moog's DIY paper is requires the inductor to be a 100 uH, hi-Q variable inductor with model number RWRS-T1015Z. In our research for a variable inductor that would be a good substitute we came across the A7BRS-T1040Z which has to following properties

Nominal Inductance (uH)	100
Q	80 @ 796 kHz
Dimensions (mm)	7.5 X 7.5
Height (mm)	12
Cost (digi-key)	$1.00

Table 7: Key properties of substitute variable inductor for the VPO & FPO A7BRS-T1040Z.

It should be noted that the 2003 version of the design slightly remedies this issue by splitting the 100 uH variable inductor into two inductors, one 47 uH inductor and a variable 47 uH inductor. The choice we made in finding a substitute should suffice for the project.

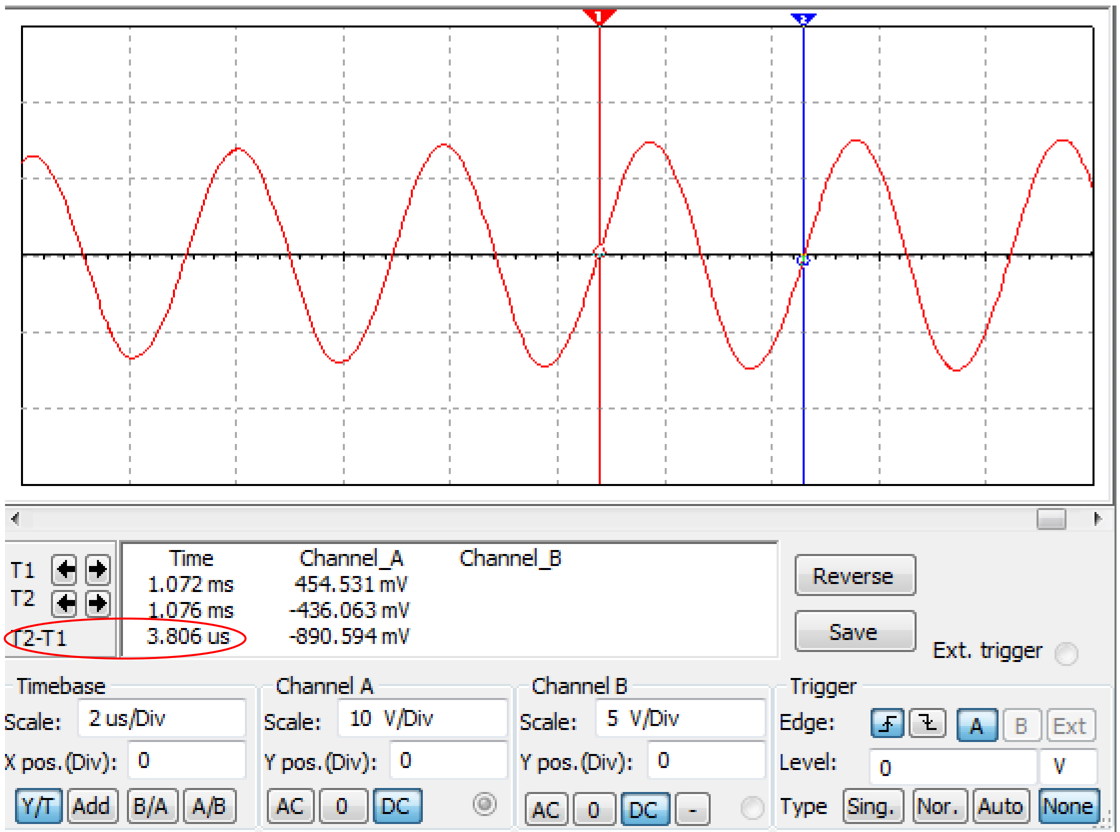

Figure 25: Output of the FPO at label "To_Detector" captured using Multisim software

In Figure 25, we can see what to expect from the FPO. We expect to see a signal with a peak voltage around 15 V. Also, note from Figure 25 that T2 – T1 = 3.806 us which translates to a signal with a frequency of about 262.7 kHz. As mentioned earlier, this was done by tweaking L6 in the Multisim simulator to be about 93% of the nominal value of the 100uH variable inductor. Our final prototype is tuned to 285kHz using the same method mentioned.

In Figure 24 a pin label "Pitch_tuning_in" can be noticed to be connected to the FPO. The use of the pitch tuning is mentioned in the research section of this paper. Since it was a requirement for our uWave Theremin to have some kind of way by which the player can adjust the sound range during performance, a pitch tuning circuit must be designed. This circuit interacts strictly with the FPO to adjust its output signal and frequency.

Essentially, the pitch tuning circuit acts as active impedance. Meaning it adjusts the current drawn by the FPO. To achieve this we used a RC transistor circuit. A detailed look at the circuit can be seen in the figure below.

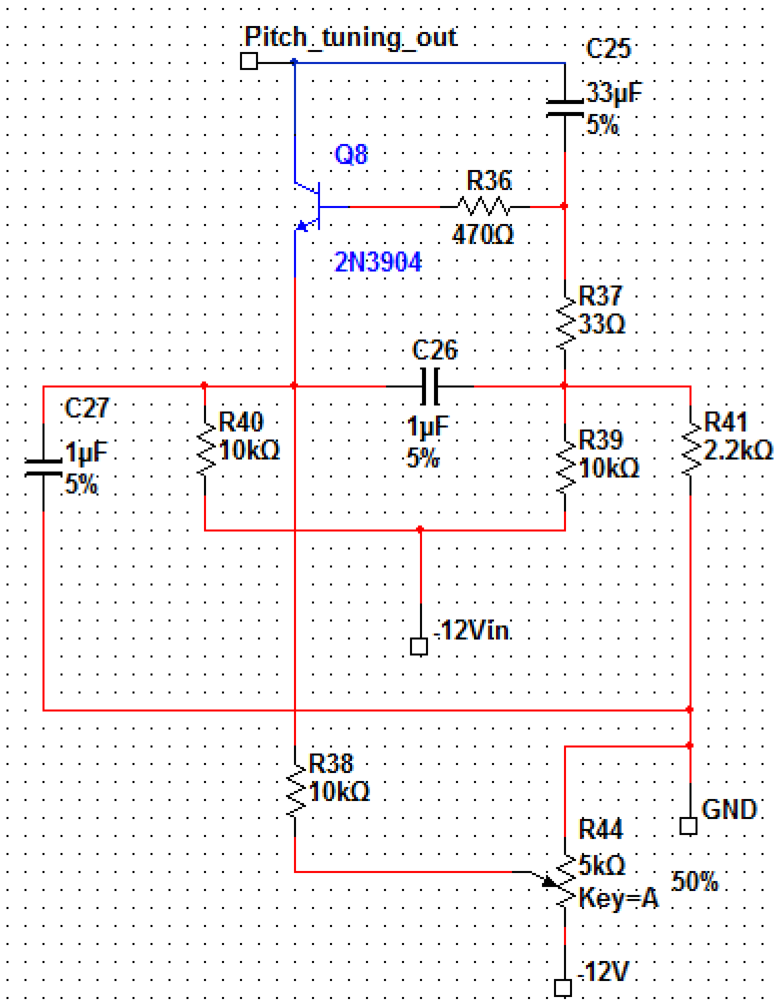

Figure 26: Design of the pitch tuning circuit

We connected the pin labeled "Pitch_tuning_out" to the appropriate location seen in Figure 24. The 5 KΩ potentiometer labeled R44 is mounted on the uWave Theremins front panel and will be used to adjust the current through the transistor labeled Q8, thus creating an active impedance seen by the FPO circuit.

6.1.3.2 Variable Pitch Oscillator (VPO)

The VPO is where the magic of the uWave Theremin is created. Essentially, this circuit is virtually identical to the FPO with the exception that its output signal might not necessarily have a constant frequency, but instead its frequency will be allowed to vary as its name implies. Much like the FPO, we constructed it separately and more importantly, we tweak it to oscillate at a frequency very close to that of the FPO. The reason for this importance is because the difference between the two oscillators, FPO & VPO, will be used to create the audio signal. If the hand isn't present on the pitch antenna, we want the two frequencies to be virtually the same or the frequency difference between the two oscillators to be almost zero. Like the FPO, the VPO uses a tuning inductor, L4 in Figure 27, which allows us to set the initial frequency to something close to the FPO. More on how we will achieve this will be discussed later.

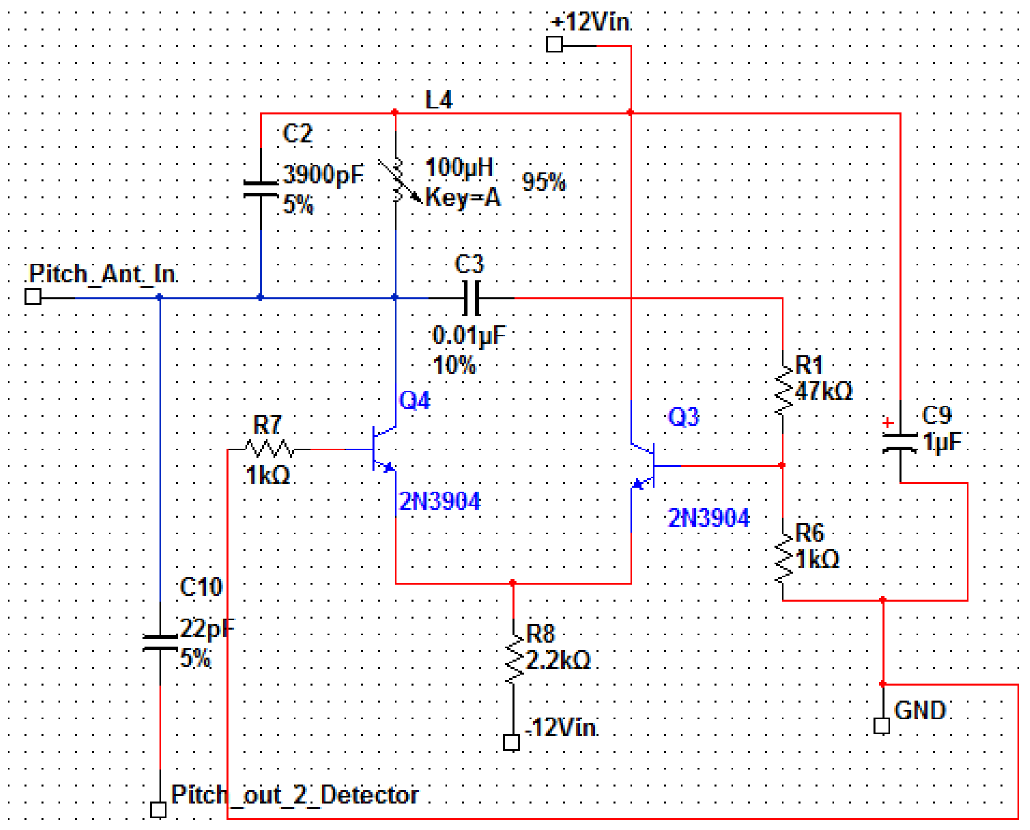

Figure 27: Schematic design of the Variable Pitch Oscillator

Earlier, it was mentioned that the FPO and VPO are similar. Another aspect in which the two oscillators are similar is that their output frequency can be altered. The FPO's frequency is altered by the pitch tuning circuit while the VPO's frequency is altered by the pitch antenna. As seen in Figure 27, the VPO is connected to the pitch antenna circuitry at the pin labeled "Pitch_Ant_in". Since the VPO circuit is identical to the FPO, Figure 26 can be referred to for its output. The outputs of the FPO and the VPO will serve as inputs to the detector circuit which will be discussed later in the design section.

6.1.3.3 Volume Oscillator

According to our requirements and information gathered during our research, the volume oscillator is designed in such a way that it will oscillate at a frequency slightly close to the resonant frequency of the volume antenna circuit of 450 kHz. This oscillator is almost similar to the other oscillators in the sense that its output varies with respect to external active impedance – two active impedances to be exact, the players hand and a tuning circuit. This oscillator is composed of RC components, transistors and also a variable inductor. Like the other oscillators, a variable inductor is used to tweak the volume oscillator to a frequency slightly in range with the frequency of the resonant antenna circuit.

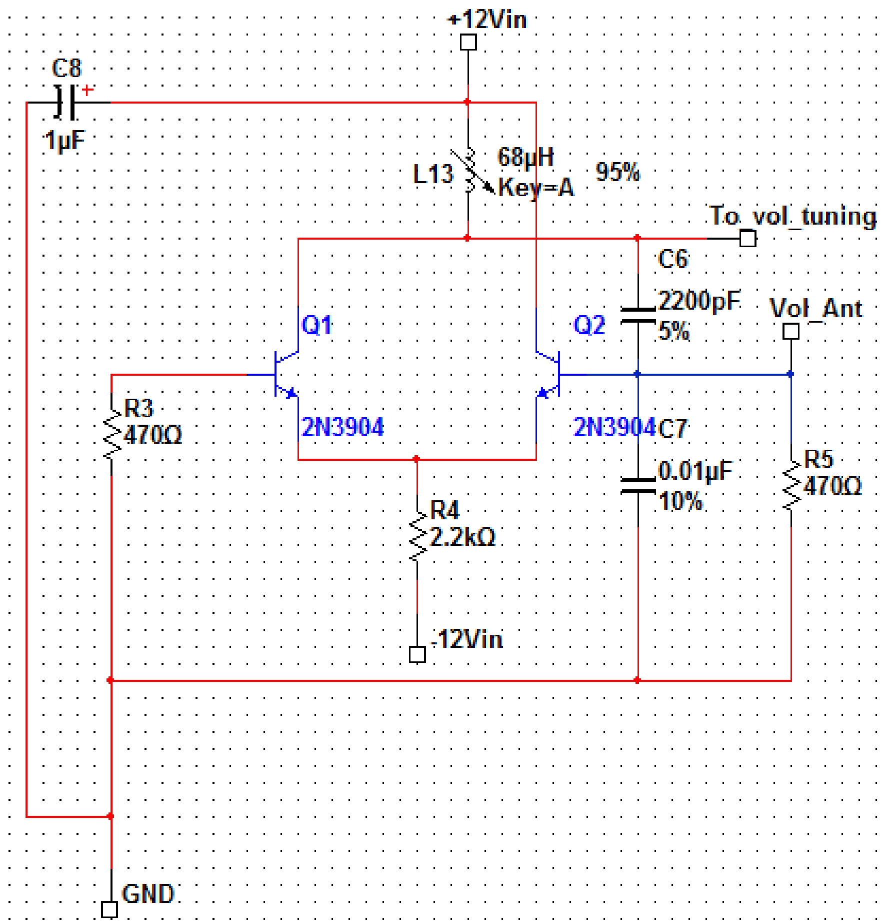

Figure 28: Schematic design of the Volume Oscillator

As was the case for the VPO and FPO, the variable inductor used in Robert Moog's design has become obsolete as he uses the 154ANS-T1019Z which is a 68 uH variable inductor. During our research, the group came across Coilcraft's SLOT TEN-5-11 tunable inductor with a nominal inductance of 62.8 uH.

Nominal Inductance (uH)	62.8
Inductance Range (uH)	36 - 90
Q	30 @ 2.5 MHz
Dimensions (mm)	10 X 10
Height (mm)	13
Cost (Coilcraft)	$3.38

Table 8: Key properties of the substitute variable inductor for the Volume Oscillator, SLOT TEN-5-11

From the table above, the SLOT TEN-5-11 is suitable for use in the volume oscillator circuit. Though its Q-factor seems low, we will not be operating at a frequency close to that range.

The 62.8 uH variable inductor used to tweak the volume oscillator to a frequency of 432 kHz is labeled L13 in Figure 28. The output from our simulation shown below shows us what to expect from the oscillator.

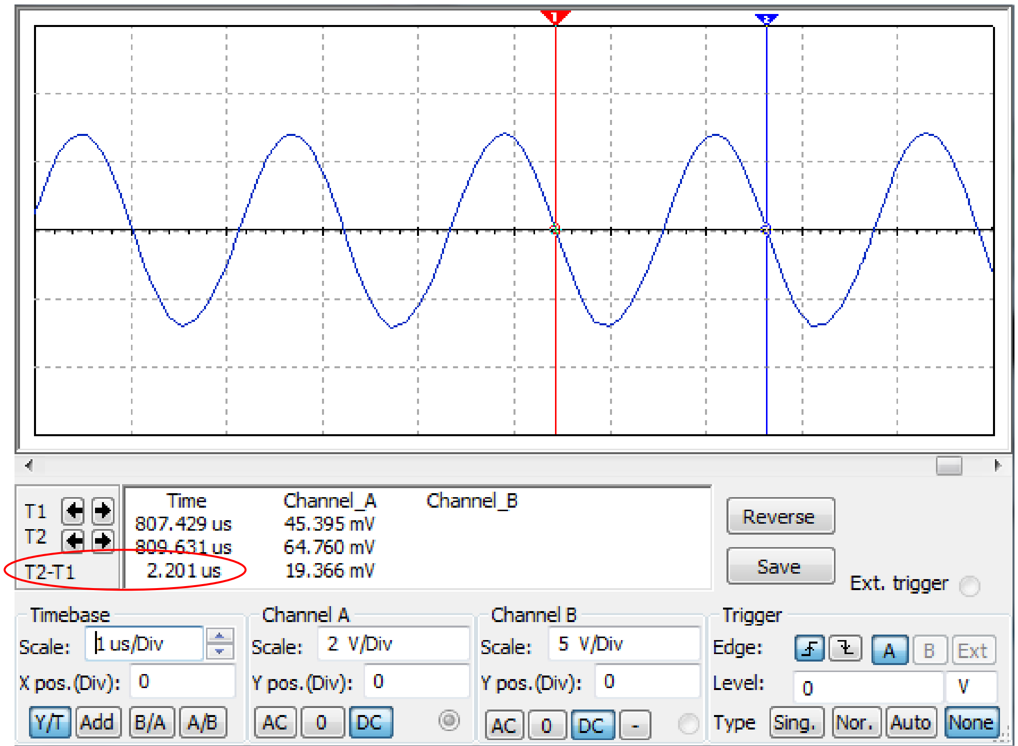

Figure 29: Output seen at R18 of the Volume Oscillator captured using Multisim software.

With the circuit in Figure 28 tuned using the variable inductor, L13, the output seen is an oscillating signal with peak voltage about 3 V and a frequency of 454.3 kHz (frequency used for simulation purposes). One thing to note was that the variable inductor had to be adjusted to 98% of its nominal value in order to achieve that frequency. We suspect this to not be the case during actual testing and expect more room for tuning with the inductors if need be.

The volume oscillator interacts with a tuning circuit, which allows the player "to make fine adjustments to the volume oscillator frequency during performance"[8]. This tuning circuit is very much identical to the tuning circuit used to adjust the FPO. The volume tuning circuit basically performs the same function the pitch tuning circuit does. It creates active impedance that alters the current draw of the volume oscillator thus changing its frequency. The circuit design can be seen below.

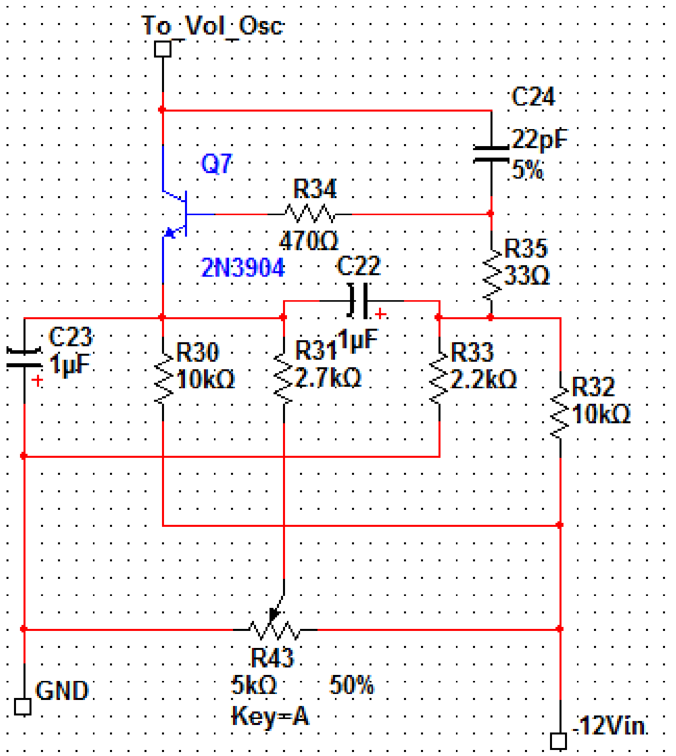

Figure 30: Design of the volume tuning circuit

The pin labeled "To_Vol_Osc" will be connected to the appropriate location on the volume oscillator. The 5 Kohm potentiometer R43 is mounted on the front panel of the uWave Theremin and will be used to control the current through the transistor Q7, thus creating an active impedance circuit seen by the volume oscillator circuit.

6.1.4 Beat Frequency Detector

According to block diagram seen in Figure 4, the signals from both the FPO and the VPO are connected to a detector in order to retrieve the beat signal (pitch preview) shown in Equation 4 of our research section. For this requirement we used an envelope or "cat-whisker" detector similar to Moog's design. The schematic representation can be seen below.

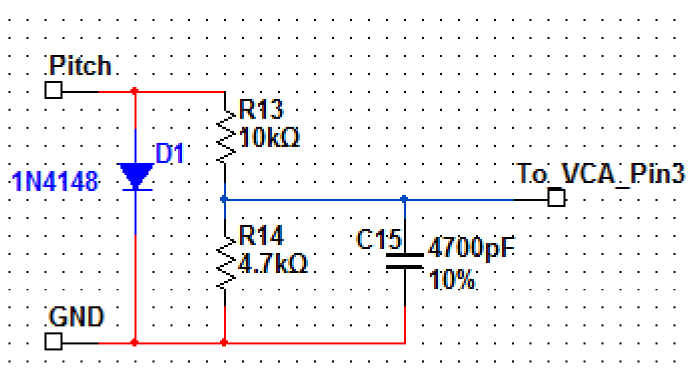

Figure 31: Simple envelope detector circuit using a diode mixer and a low-pass filter

The envelope detector uses a single diode as a mixer and a low-pass filter with a cutoff frequency of about 10 kHz as seen in the bode diagram below.

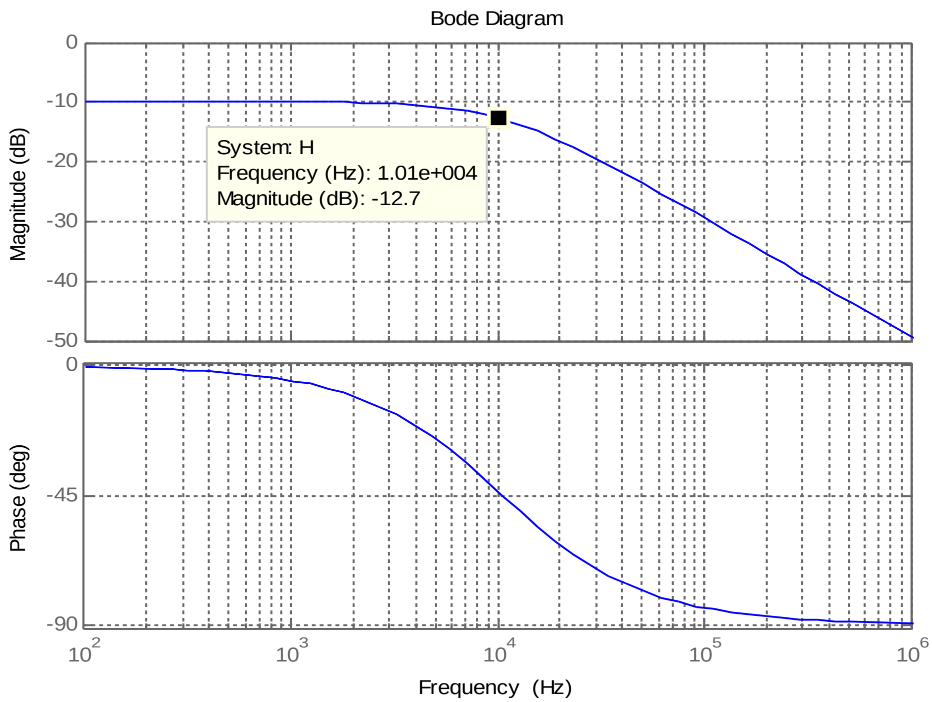

Figure 32: Bode plot of the detector circuit showing a cutoff frequency of 10 kHz.

From the bode plot alone, it is noticed that frequencies from 0 – 10 kHz will produce a very low output with magnitude on the order of -10 dB. According to the work done in our research, the signal coming from the detector should be a small signal that requires amplification in order for it to be used with a speaker or headphone. Later on we will discuss how this small signal will interact with the SenseBox in order to achieve midi data.

The output of the mixer is an amplitude modulated signal as seen in the figure below. As noticed in our simulation seen in Figure 33 the envelope detector should do a pretty good job of following the amplitude variations of the AM signal. To achieve the output in the figure above, the VPO is set to about 261 kHz (using inductor L4) and the FPO set to about 263 kHz. For lack of an efficient way to simulate an antenna in Multisim, frequency variation between the two oscillators was accomplished using their respective variable inductors L4 and L6 respectively.

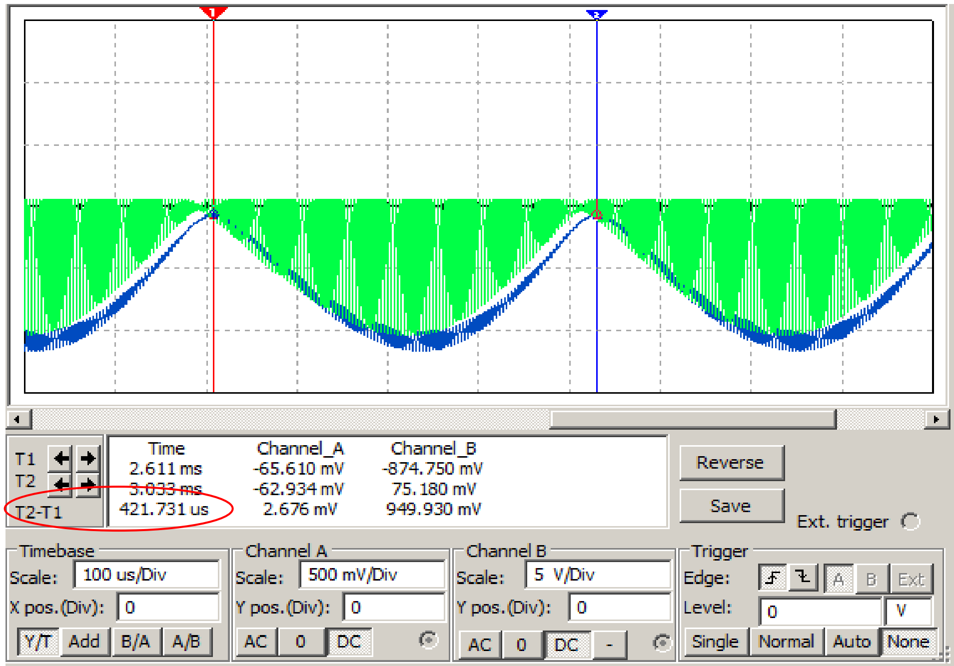

Figure 33: Mixed signals from FPO & VPO (green) and output of detector (blue)

The envelop signal has a frequency that equates to the desired audible frequency of f_1-f_2, which should be in the range of 0-3kHz during performance. This can be seen in the circled area in the figure above. T1 – T2 = 421.731us which roughly equates to 2 kHz. Also another thing to notice is the output signal from the envelope detector is a relatively small signal with peak at about -65 mV. This is to be expected from our research and specifications for this device. This signal is not powerful enough for most speakers therefore it must be amplified. This will be the discussion in the next section.

6.1.5 Pitch Amplification

As mentioned in the previous section, the output of the detector contains the pitch/frequency data required to play a musical note. Since the pitch signal is a small signal, an amplifier was used. According to work done in research, the uWave Theremin will increase volume relative to the hand distance from the volume antenna. Therefore, a voltage controlled amplifier (VCA) is used to amplifier the pitch signal and this is the topic of this section.

6.1.5.1 Voltage Controlled Amplifier (VCA)

In Robert Moogs design, he uses the LM13600 dual operational transconductance amplifier to convert the volume oscillator behavior into a current that will be used by the VCA to amplifier the pitch signal. These amplifiers have become obsolete as an update has been introduced to the market from manufacturer, National Semiconductor. The newer amplifier, LM13700 was used as a substitute for our design. The major improvement of the LM13700 over the LM13600 is that it has a much improved output buffer. For our project, the group decided to use the LM13700 amplifier for both the VCA and the VCA processor.

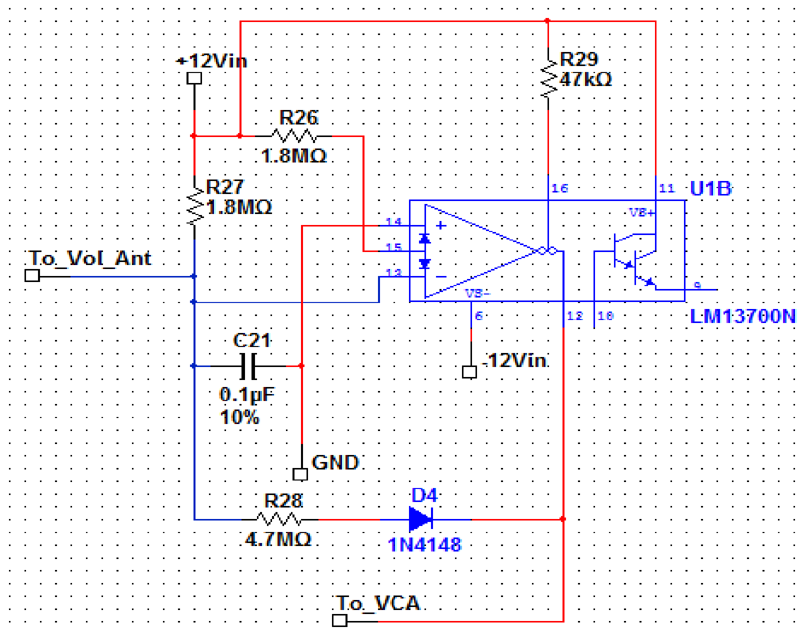

Figure 34: Design of the VCA processor

The voltage/current produced from the volume oscillator and volume antenna is sent to the VCA processor circuit seen in Figure 34 at pin labeled “To_Vol_Ant”. This circuit is responsible for level-shifting the relatively low voltage coming from R2 of the volume antenna circuit Figure 22. The output of the VCA processor is sent to the VCA. This is the amplification factor for the pitch signal at the VCA.

6.1.5.2 Voltage Controlled Amplifier (VCA)

The VCA was designed using the LM13700 amplifier to amplify the pitch signal received from the envelop detector. The design follows that of Moog's, which can be seen below. As its name implies, the VCA will amplifier an input signal by a gain factor relative to the voltage from the VCA processor. The circuit in Figure 35 is the VCA design referenced in Moog's theremin design. From the figure, it is noticed that the VCA has one output, the “Audio_out”, and it will take in three inputs. It takes in the demodulated signal from the detector and amplifies it. It also takes in two inputs from the player, which are external potentiometers connected to the front of the uWave Theremin to make adjustments to the output audio signal.

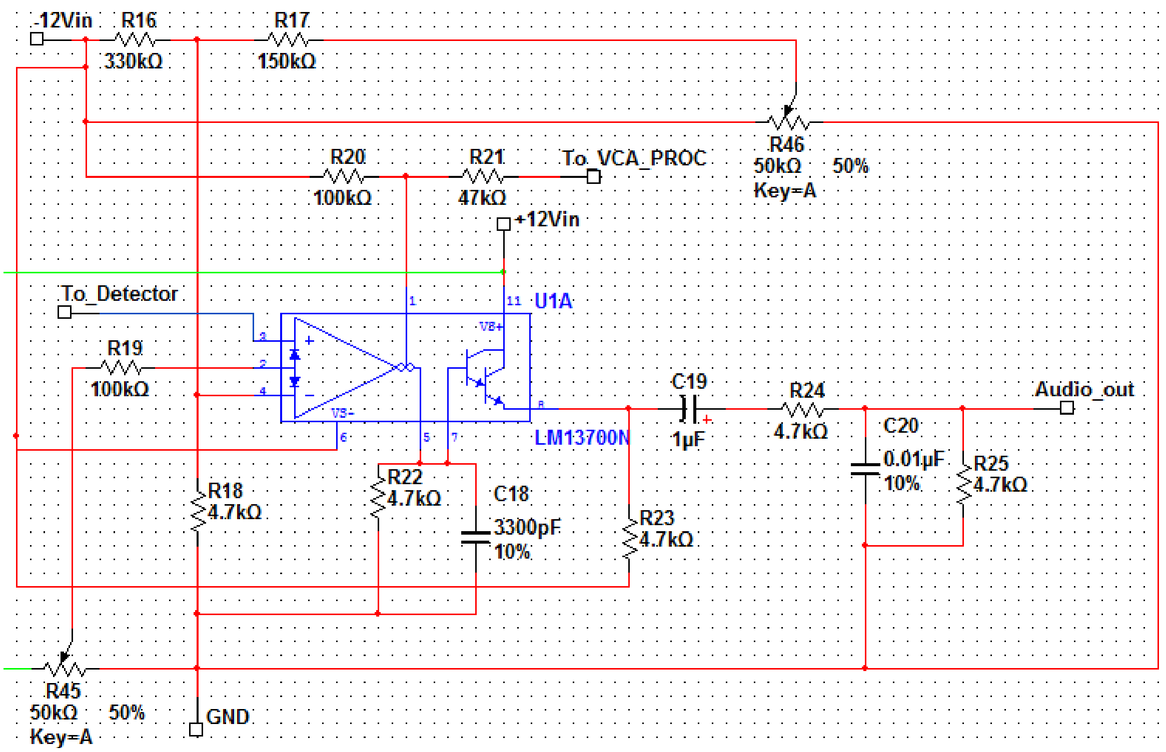

Figure 35: Design of the VCA

The VCA is intentionally designed to distort the audio waveform in order to add harmonic contents. The 50 KΩ "Brightness" potentiometer, R45, is used to control how much the audio signal waveform gets distorted, hence controlling the amount of harmonic content. Another 50 KΩ potentiometer is connected and it is known as the "Waveform" potentiometer. It is used to determine which harmonics will be strong and which will be weak[44].

6.2 SenseBox Pitch-to-MIDI Converter

While investigating a way to convert an analog signal to a MIDI data, we stumbled upon Stephen Hobley's blog page. Stephen Hobley is a software engineer who works in a major software company in Indianapolis, Indiana, but in his spare time he has a passion of re-building and re-modeling electronics. In his blog, he reveals in great amount of detail about how to setup and build a Pitch-to-MIDI converter. Our SenseBox design is based on his specifications with some additional features. We have asked Hobley for his permission to use his design for our scholar usage.

6.2.1 Meeting Specifications and Requirements

According to the requirements given for the SenseBox, the converter shall translate the analog signal from the theremin and produce a MIDI note that correlates with the given analog signal. The SenseBox have two inputs from the theremin: volume and pitch preview. Furthermore, the converter have MIDI IN and OUT connectors, as well as an audio output. The converter should output a range of 8 octaves of MIDI notes and it will only send out one MIDI note at a time (the converter will act as a monophonic interpreter). The SenseBox can receive MIDI data from other devices, which enable a feature that allows the user to play in ARPEGGIO mode. The converter includes a LCD screen panel, as well as series of LED's for visual aid demonstrations. The SenseBox operates with only 5 voltage supply.

6.2.2 Hardware

The following sections will explain how to configure the hardware components for the SenseBox, specifically with the ATmega328p chip.

6.2.2.1 Microprocessor

For our SenseBox, we used the ATmega328 microprocessor chip (pre-loaded with the Arduino bootloader). The ATmega328 microprocessor meets our criteria of achieving our goals, in terms of hardware and software: the microcontroller is in charge of handling the analog to digital conversion into MIDI data.

With this intention, when building the PCB board for the SenseBox, a few components was added to make the ATmega328p into an Arduino microcontroller: a 16MHz crystal oscillator, 5V regulators, and a reset button. Since Arduino is notably known for being an open source hardware, integrating into the PCB design can be easily obtained[13].

6.2.2.2 Audio In & Out

Recall that we are designing our input for the converter with TS connectors. When connecting the SenseBox with uWave Theremin, we have decided that both devices will be in vicinity to each other (roughly 5 inches apart) The SenseBox will have female TS plug mounted onto its side so only an TS cable is needed to connect both devices. The tip of the TS connector is used for sending or returning the signals and sleeve is used for ground. The SenseBox have two female TS connectors: volume and pitch preview signals.

6.2.2.3 Electrical Configuration

The following diagram demonstrates the configuration of the different components for connecting the SenseBox into the pins in the microcontroller:

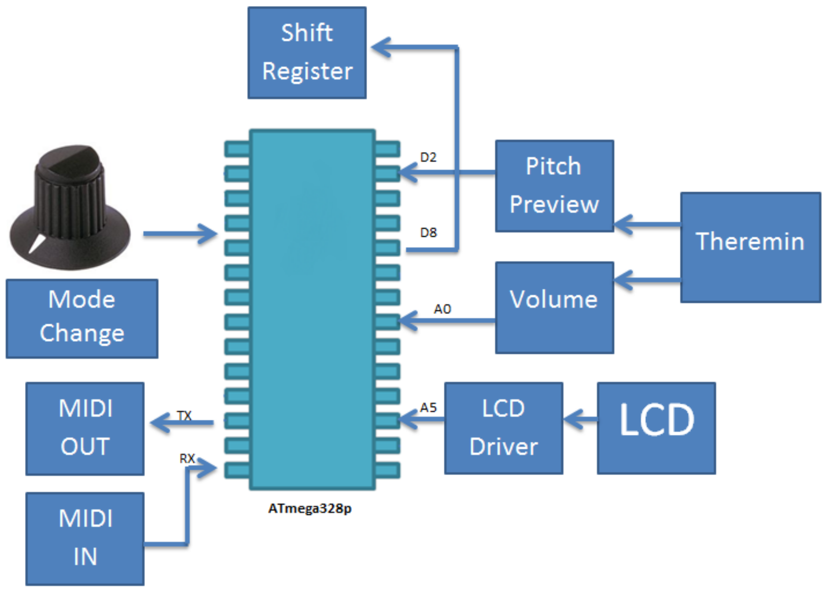

Figure 36: Block diagram of hardware connections with ATmega328p

The configuration above will be explained thoroughly for all its components in the following sections.

6.2.2.3.1 Volume

In order to capture the volume data from the uWave Theremin, we needed to consider how to incorporate it with our microprocessor. One of the specifications that the ATmega328p has for its analog-to-digital converter is that it can only have an analog input from 0 to 5 V. Furthermore, its analog to digital converter specification is a 10-bit converter, which means that its input voltages will convert into integer values between 0 and 1023 (10 bit equals to $2^{10} = 1024$).

We have chosen LM324N for our operational amplifier because we want our design to operate just with a 5V supply and ground—the LM324N can be directly operated with a standard 5V supply.

Our basic audio system is as shown in the following block diagram to capture the volume data from the uWave Theremin's output.

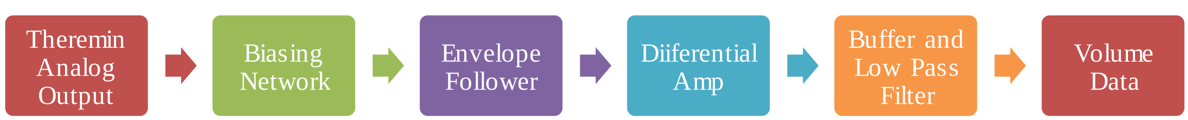

Figure 37: Volume data capture from uWave analog output

According to the design of the uWave theremin, the audio output has a maximum level of about 0 dBm (0.8V RMS). For this reason, we need to modify our analog output in a way that the Atmega328p can capture the details of the output's amplitude

6.2.2.3.1.1 Biasing Network

Having an AC signal with for roughly 1V, we need to bring the AC signal up in the range where it can be active with the op-amp. As shown in the schematic diagram, this demonstrates how to bring the incoming AC signal from the uWave Theremin and give an offset voltage. In the simulation, as shown in Figure 39, the input that is fed to the "Volumein" pin in the schematic below is 5 kHz at peak-to-peak 1V. The red signal indicates the output and the blue signal indicates the output of the biasing network.

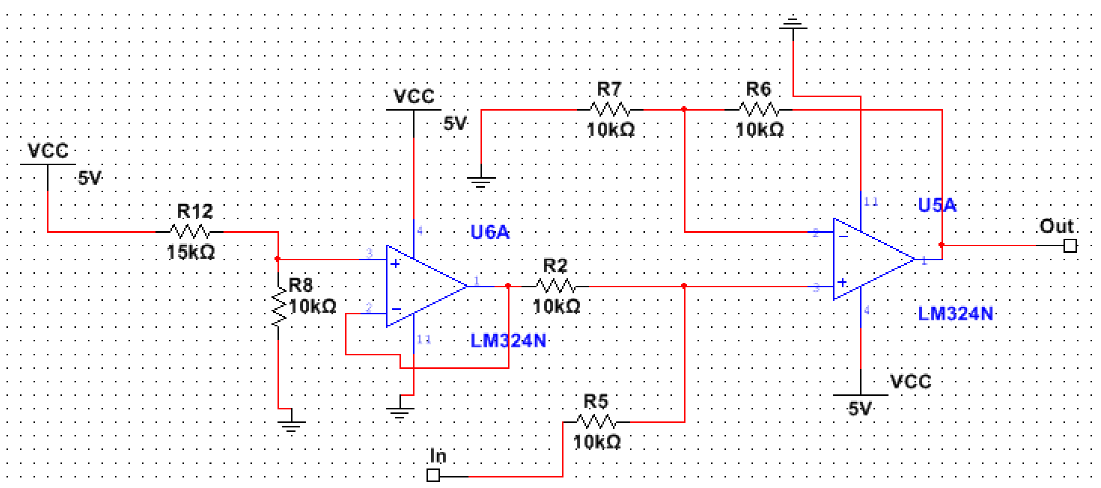

Figure 38: Biasing Network Schematic

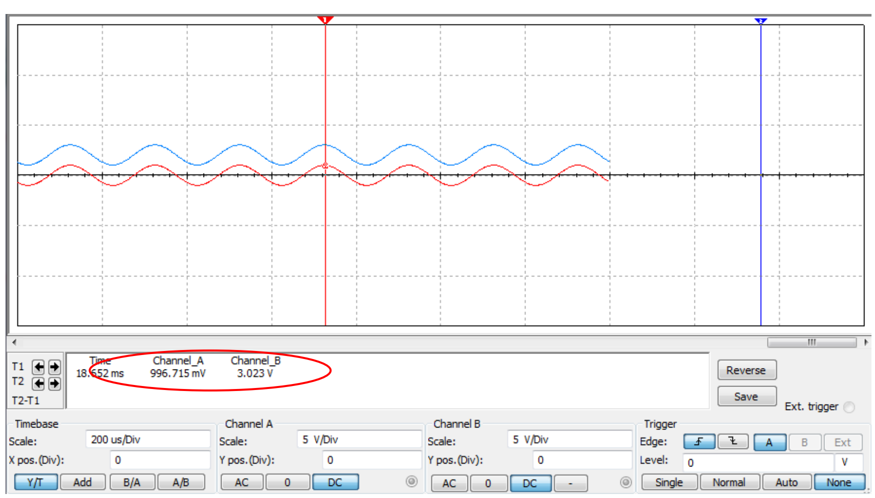

Figure 39: Scope Output of Biasing Network

The first stage of the biasing network provides the DC offset of 2V using a voltage follower, which will feed into the incoming AC signal from the theremin. The second stage provides the combination signals of both the DC offset and AC signal from the theremin.

6.2.2.3.1.2 Envelope Follower

The schematic diagram below demonstrates our envelope follower:

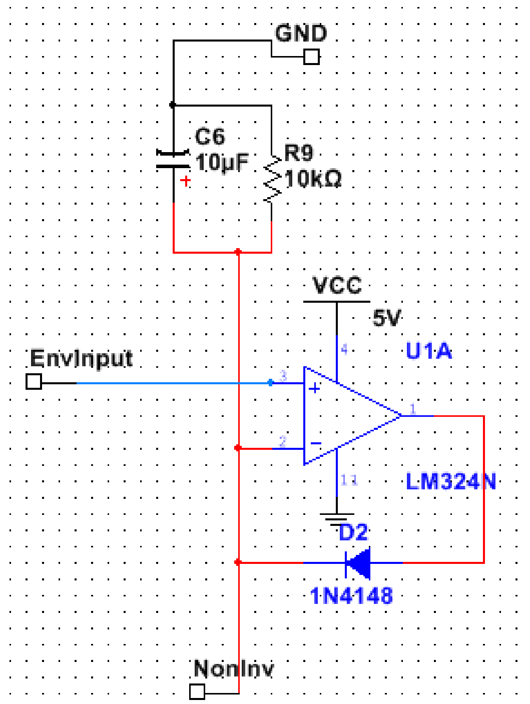

Figure 40: Envelope follower schematic

In essence, the envelope follower is responsible for converting the incoming AC signal and pulse it into a varying DC signal, based on the AC signal's amplitude. The capacitor charges on the rising edge and when the signal starts to slowly fall, it will release onto the resistor, which will then feed onto the diode, and produces half-wave rectification. Figure 41 illustrates that behavior (note that the blue signal indicates the input and the red signal indicates the output of the envelope follower).

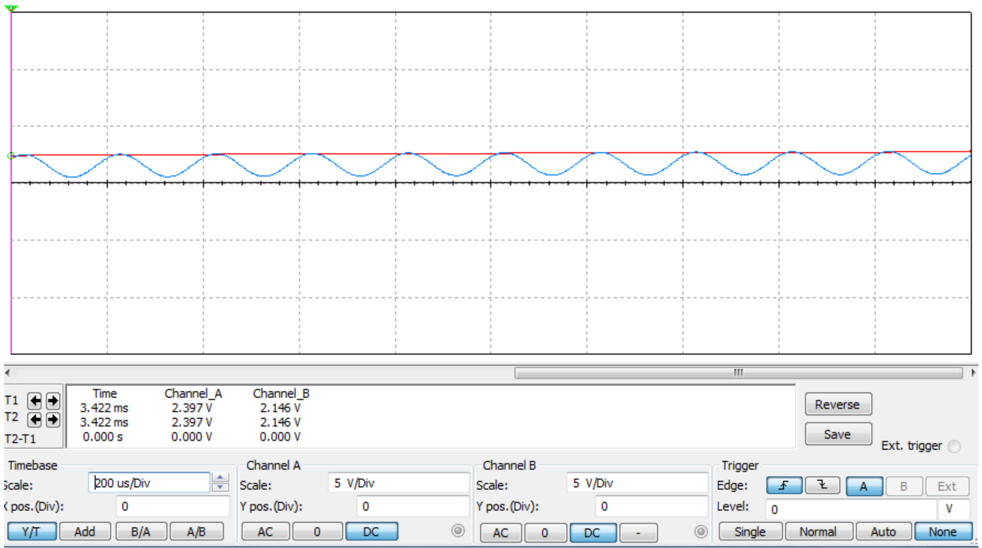

Figure 41: Output of Envelope Follower

6.2.2.3.1.3 Differential Op-amp

Once the output from the envelope follower is obtained, we seed the signal onto the differential amp. As shown in Figure 42, the "NonInv" pin is obtained from the output of the envelope follower. "Inv" pin is obtained from an external voltage supply that we've designed to scale down the "biased" signal back down to the original signal output. Figure 43 reveals the behavior of the Differential op-amp. The blue signal represents "NonInv" pin, whereas the red signal represents the "Inv" pin and the output of the Differential op-amp represents in pink.

Recall that the Atmega328 microprocessor can only take a DC voltage ranging from 1-5V to its analog pin. We have chosen the according resistor to have a gain where it's reachable within that range.

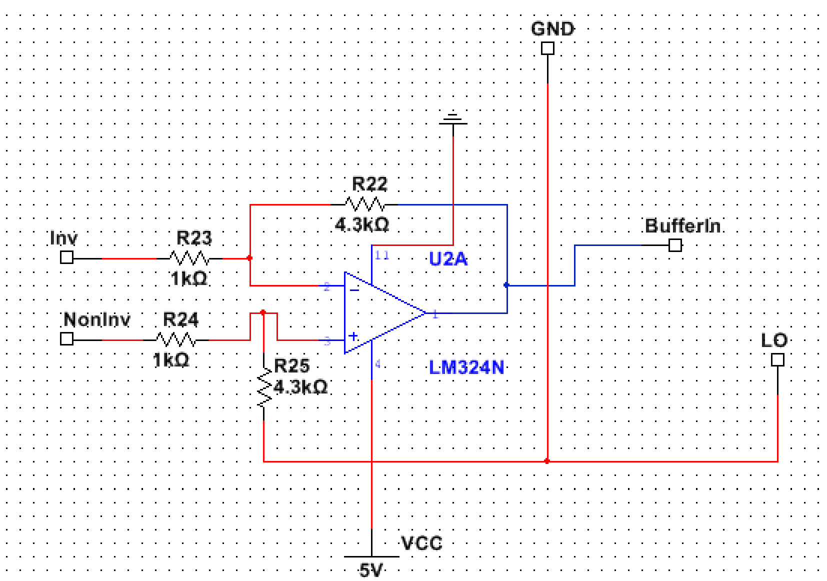

Figure 42: Differential op-amp circuit implemented

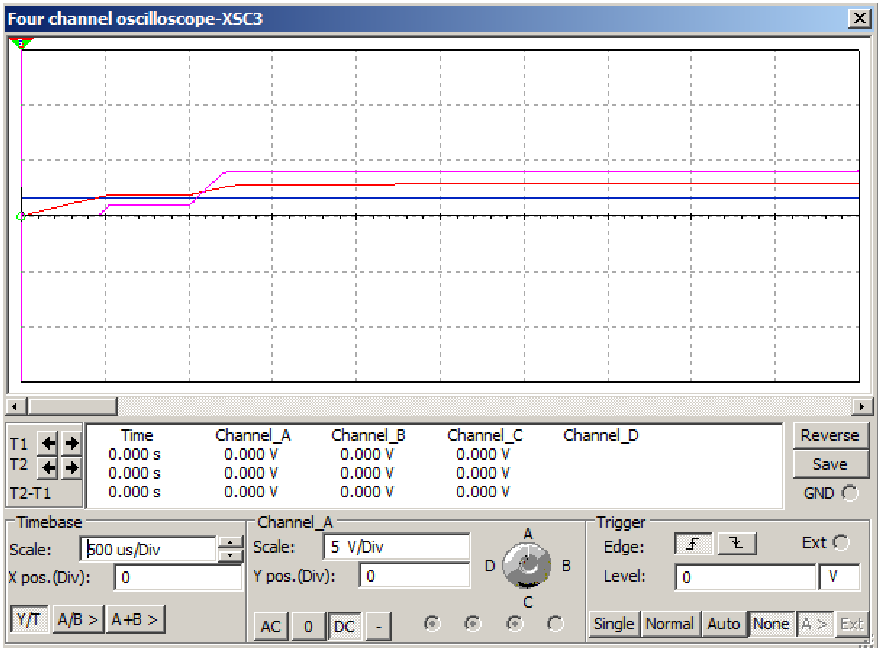

Figure 43: Scope Output of Differential op-amp

6.2.2.3.1.4 Buffer and Low-pass Filter

In the last stage, the signal will go through a standard buffer, which is used to filter out unwanted low frequency signal from the capacitor as shown below:

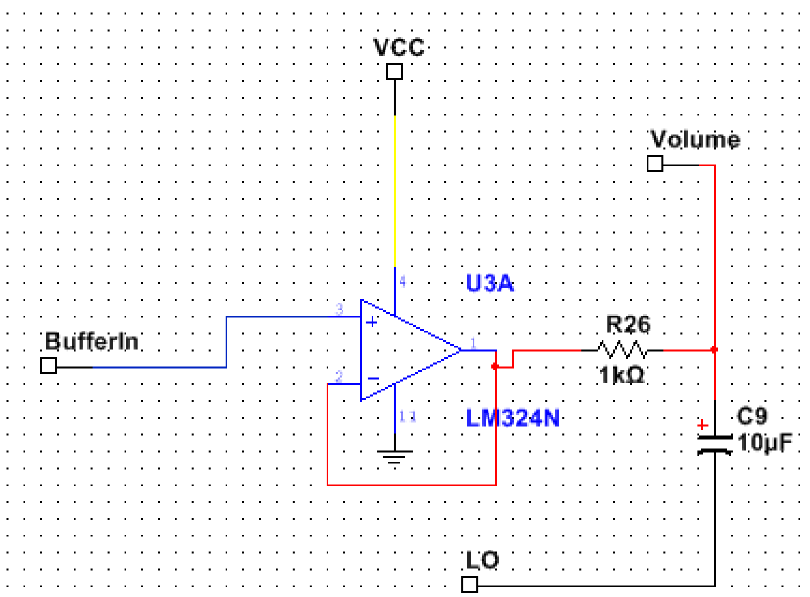

Figure 44: Buffer and Low Pass Filter Schematic

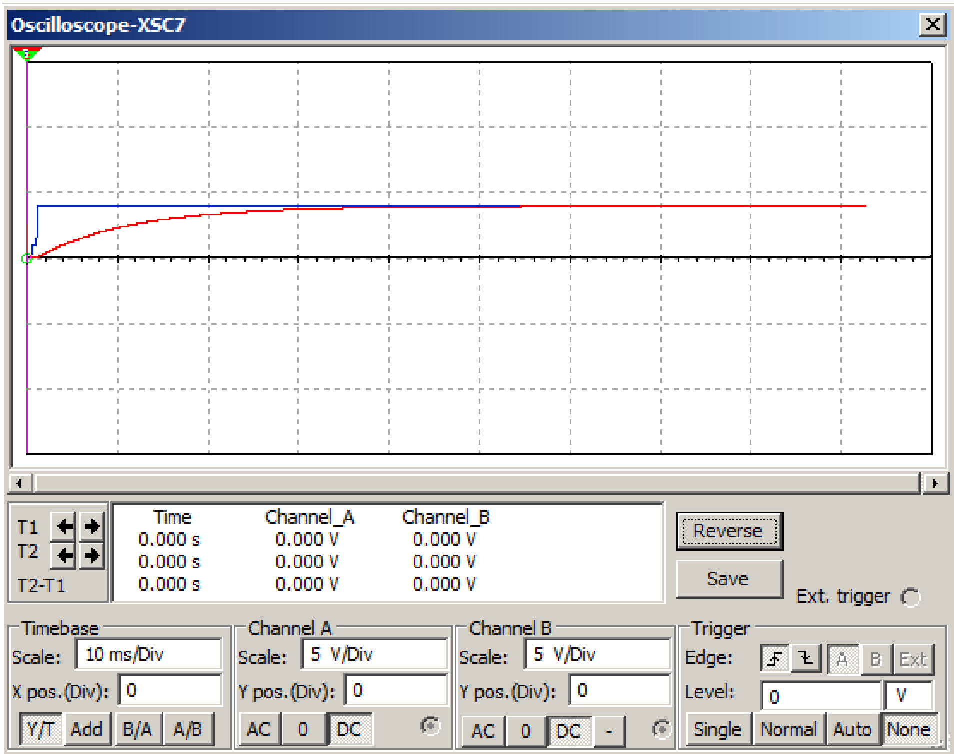

Figure 45: Buffer and Low Pass Filter Schematic

In Figure 45 the red signal represents the output of the Differential op-amp and the blue signal represents the output of the buffer. The output of this will feed into one of the analog input of the ATmega328p chip to obtain the volume data for our SenseBox. For more information on the method of extracting the data, refer to the software section.

6.2.2.3.2 Pitch

Unlike extracting the volume data, we need to connect the uWave Theremin's pitch preview output (output at the detector), instead of the actual audio from the uWave Theremin simply because we are only interested in what frequency it is emitting. The following diagram demonstrates our method of capturing the pitch data:

Figure 46: Pitch data flow

Typically, the voltage range from the pitch preview is in the magnitude of milli-volts but since we are only concerned about the pitch, we are only interested in obtaining the frequency. The range of the frequency is around 0 to 3 kHz.

6.2.2.3.2.1 Threshold Comparator

The following schematic diagram demonstrates the threshold comparator which was used in this design:

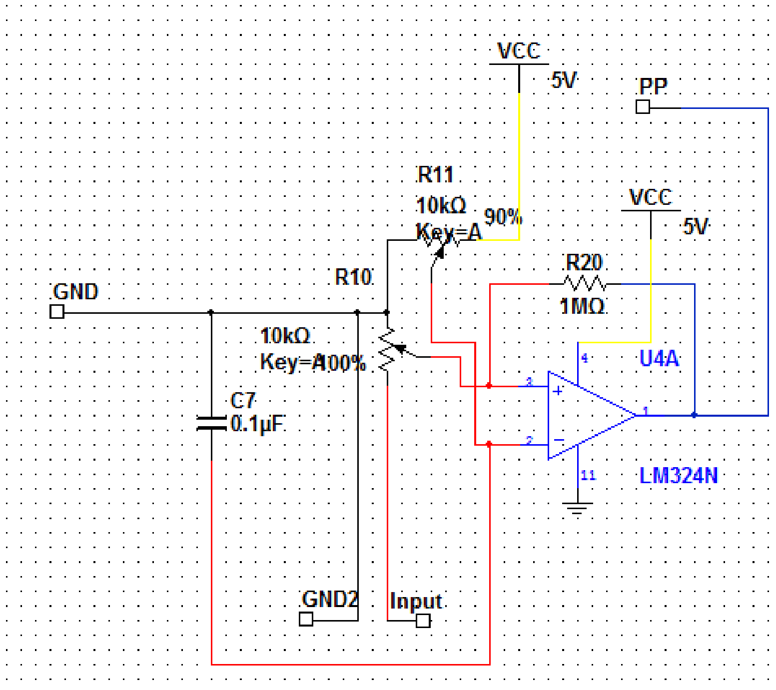

Figure 47: Threshold Comparator Schematic

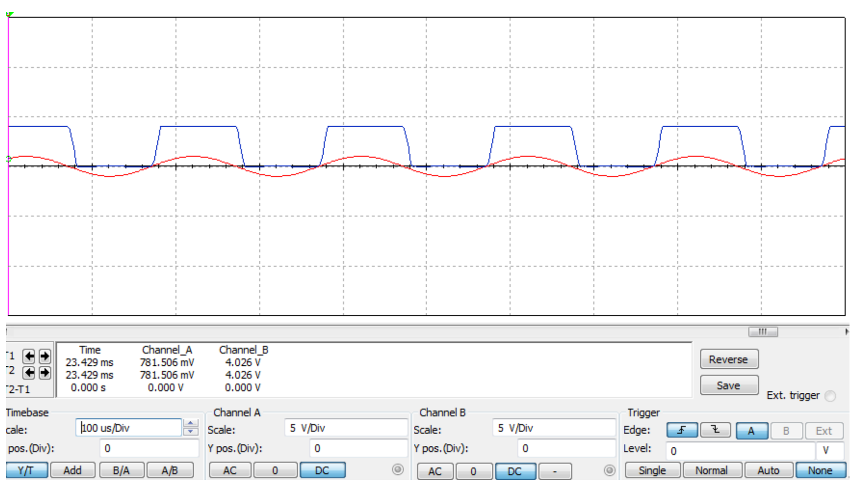

Figure 48: Scope of Threshold Comparator

The concept behind a threshold comparator is that it will convert the pitch preview signal from the uWave Theremin into a square wave output. The reason we must do this is because since we are feeding the pitch preview data onto the digital pin of the microprocessor, we need to have a binary signal - the HIGH output of our design will produce a voltage of 4 and LOW output will be a voltage of 0.

For more information on the method of extracting the data, refer to the software section.

6.2.2.3.3 MIDI Connection

Recall the role of MIDI OUT is to send out MIDI data to another MIDI device while MIDI IN is to receive the MIDI data. Hence, this means that if the SenseBox is connected to another MIDI device, there must be a serial communication. Thus, the configuration for both MIDI OUT and IN is interfaced using TTL serial communication from the ATmega328P; the microprocessor only has one set of UART serial ports.

In the Atmega328p microprocessor, digital pins 0 (RX) and 1 (TX) are used to receive and transfer TTL serial data. RX pin and TX pin is the digital input and digital output, respectively. In our design, RX pin is utilized in the MIDI IN because it will receive MIDI data as digital input from a MIDI device and TX is utilized in the MIDI OUT because it will send out MIDI data as a digital output to another MIDI device.

6.2.2.3.3.1 MIDI IN

MIDI IN only uses two of the five pins: PIN4 and PIN5 as current source and current sink, respectively. The following schematic diagram exposes a standard electrical configuration for a MIDI IN connection.

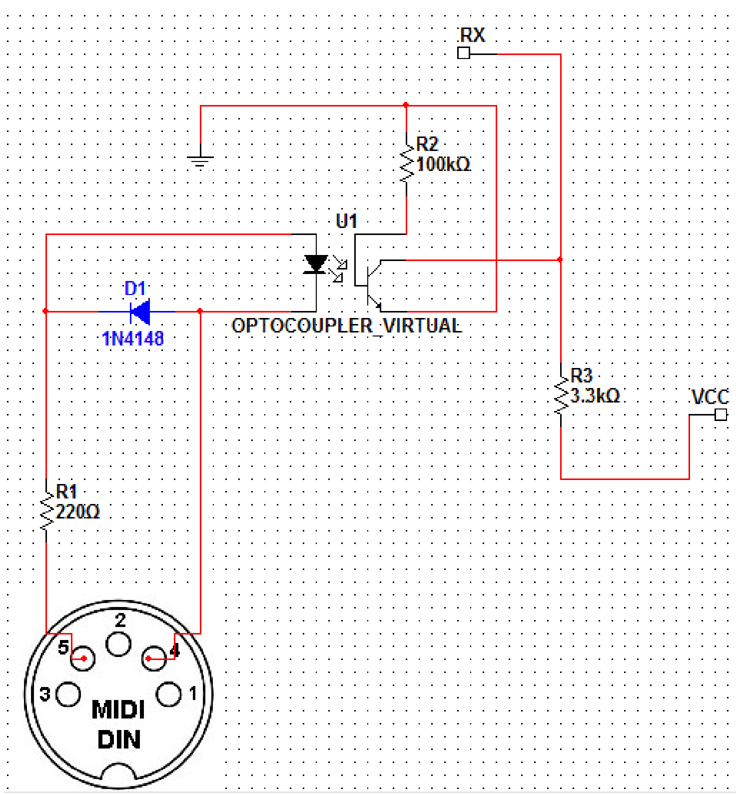

Figure 49: MIDI-IN Connection

The Opto-Isolator in this schematic diagram was used to provide electrical isolation between the incoming MIDI data and ATmega328p. In other words, it helps prevent any undesirable changes in voltage and current that might occur if the two parts were connected electrically[14]. In principal, the two circuits only connect momentarily when there is an electrical current from the source, which will convert to a light signal from the LED. After capturing the light from the LED with a phototransistor, an electrical current is initiated and fed to the RX pin of the microprocessor.

6.2.2.3.3.2 MIDI OUT

MIDI OUT involves only three of the five pins: PIN2, PIN4 and PIN5 are used as ground, current sink and current source, respectively. The schematic diagram is a standard configuration of building a MIDI OUT:

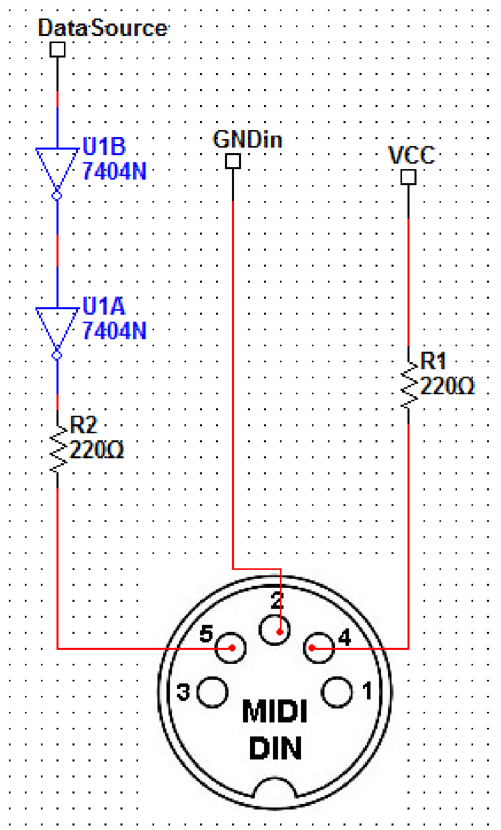

Figure 50: MIDI-OUT Connection

The two inverter gates in the above schematic diagram simply acts as buffer. The signal coming from the TX pin may have a weak signal and by using the buffer, the signal is boosted enough to drive a load resistance if needed; the logic function stays the same.

6.2.2.3.4 LCD

When considering which LCD panel should be installed in our design, we had to reflect on some constraints. One of them is the usage of power. For obvious reasons, we want to include components in our design that uses little power as possible. Thus, our group decided that we will install a character LCD because unlike using a graphical LCD, character LCD consume less than 100mA. In addition, adding a graphical LCD will require complexity in our programming. Thus installing a character LCD is enough to serve its purpose: to deliver a visual assistance for the user to show them where the notes are.

Based on these reflections, a 20x4 character LCD was selected for the design. The LCD panel should provide sufficient space for displaying necessary information for the user to know where the notes are. In addition, the 20x4 has an adjustable backlighting feature, which allows the LCD screen to be read with ease. The display contains white LED back light against a blue screen. The actual LCD screen has a dimension of 98.0 x 60.0 mm.

6.2.2.3.4.1 LCD Driver

Our group used the serial/LCD117 driver for our project. While there are other comparable LCD driver that we could have picked , we chose LCD 117 because it has a 9600 baud chip that is recommended for the Atmega328p microprocessor (it is capable of 9600 bps serial transmission). The LCD117 driver reduces the number of pins from 14 to 3 - ground, +5 voltage and serial data. In other words, the communication between the LCD and the ATmega328p is now serial, as oppose to parallel.

The LCD117 is compatible to almost any industry standardized LCD character display (it is recommended that this driver works well when connecting to a LCD that is based on Hitachi HD44780 LCD controller chip)[45].

Typically, serial LCD screens can be easily interfaced with the ATmega328p by using one of many standard software libraries that are available in the Arduino language. "SoftwareSerial" library has a list of commands that enables the user to communicate to the LCD through the microprocessor. Table 9 shows some example code which the user defines the baud rate, as well as the read and write values[46].

Library: SoftwareSerial.h		
Function	**Code**	**Description**
Initialize	Software.Serial()	Create new serial port
	serial.begin()	Set the baud rate
Read value	serial.read()	Reads in a value
Print to screen	serial.print()	Prints character to the screen

Table 9: SoftwareSerial.h Program Declarations

In addition to the built-in commands from SoftwareSerial library, the LCD117 driver further assist the programmer on navigating the LCD panel; it has a small microprocessor that virtually allows the programmer to program the LCD at ease. For instance, the driver has built-in functions that allows the programmer to adjust the backlight setting, set cursor style (none, blinking, or underline), and many other commands though serial commands. Table 10 shows some codes that were utilized in the project (for a complete command list, refer to the appendix).

LCD117 Driver Commands		
Command	**Code**	**Description**
Clear Cursor Line	serial.print("?l")	Clear the row where the cursor line is located
Clear Screen	serial.print("?f")	Clears the LCD screen
CRLF	serial.print("?n)	carriage return & line feed in the LCD screen

Table 10: Example list of commands that can be implemented with the LCD driver

6.2.2.3.5 Shift Registers

In addition to installing an LCD panel for visual aid, an 8 bit-shift register was included in the project. Our choice of shift register was a serial-in/parallel-out 74HC595 shift register. The 74HC595 is connected with 8 different LEDs for each bit register that it is assigned. The 8 bits in the shift register was grouped in two: three of the LEDs (which consist of three orange LEDs) will represent the intensity of the volume coming from the uWave Theremin, and the remaining five LEDs (which consist of two reds, two yellow, and one green LED's) will represent how accurate the pitch is in-tuned from the user's hand with respect to the pitch antenna. The figure below demonstrates how the 74HC595 is connected:

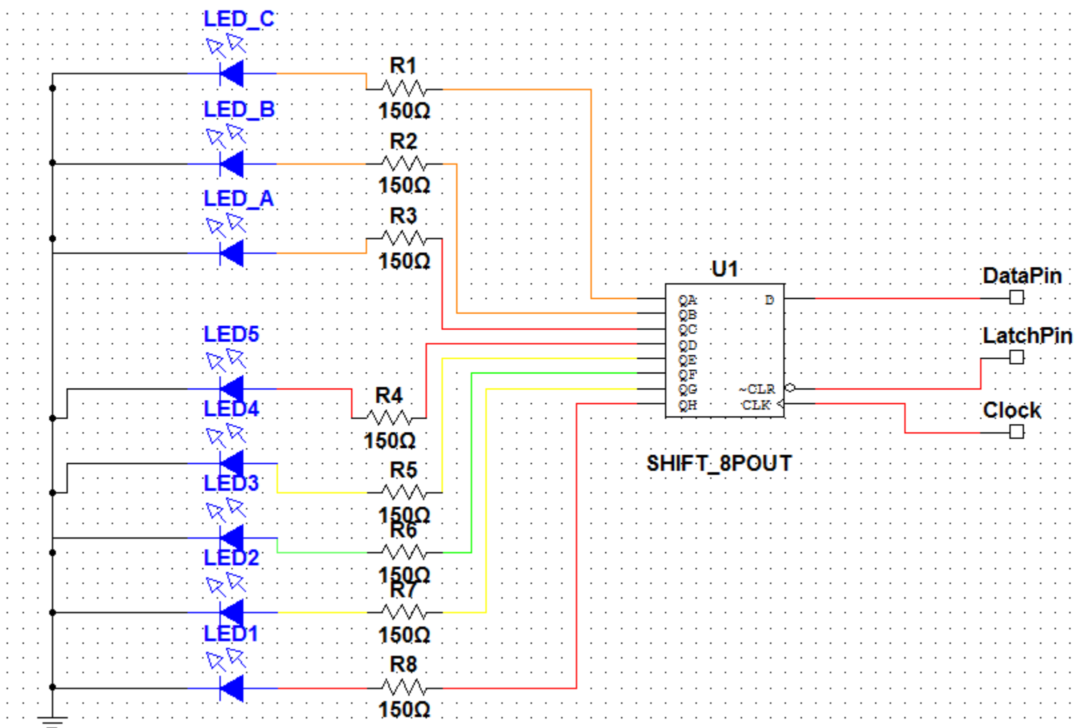

Figure 51: 8-bit register circuit

6.2.2.3.5.1 Volume LED's

The volume LED display was programmed as so: the ATmega328p microprocessor was set to have three thresholds for the incoming volume signal from the uWave Theremin. When the first threshold is met, 74HC595 will assign a HIGH on QC, which will turn on

LED_A. When the second threshold is met, both LED_A and LED_B will turn on since it will assign a HIGH on QC and QB. Lastly, when the third threshold is met, QC, QB and QA is assigned HIGH, which consequently will turn on LED_A, LED_B and LED_C. As you can see, if all the LEDs in the volume display are on, it specifies that the uWave Theremin is producing a considerably large volume (this presumably happens when the hand is not in vicinity of the volume antenna). If all the LEDs are off, it specifies uWave Theremin is not producing any volume, or sound.

6.2.2.3.5.2 Pitch LED's

The pitch LED display is programmed as so: LED1-5 was used to depict the accuracy of the current pitch; these series of LED's will turn on one at a time. The picture below describes the setup:

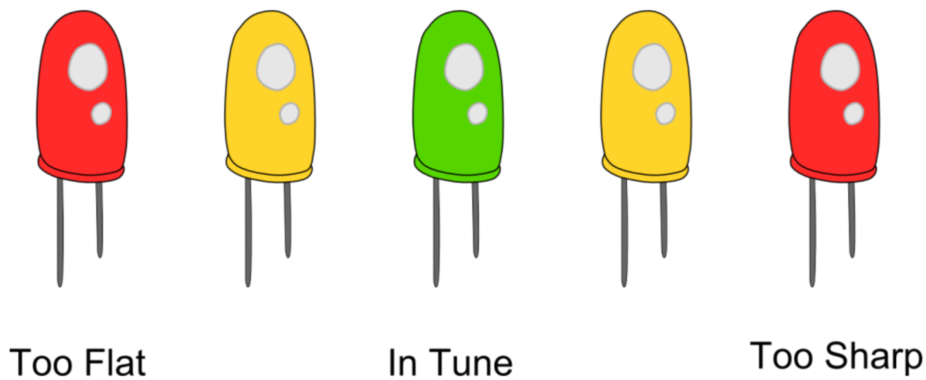

Figure 52: Pitch LED setup

When the SenseBox is receiving pitch data from the theremin, it will tell the user if the current pitch is either in tune, too flat or too sharp from the desired pitch. Markings on the readout drift left if the current note is too flat and right if the note too is sharp. If the current pitch is matched to the desired pitch frequency, the green LED in the middle will turn on. The reference pitch for our tuning system is based on the frequency of 440Hz (440Hz is the musical note A, above middle C).

6.2.2.3.6 Switches

The SenseBox includes a rotary switch. The rotary switch will allow the user to choose four modes of operation: PITCH, CONTROL, ARP1 and ARP2.

In PITCH mode, the SenseBox will track the frequency and volume from the uWave Theremin and display the current note to the LCD panel and output the corresponding MIDI data .CONTROL mode will carry the same task as PITCH MODE, but instead of sending a MIDI note and volume , it will send a MIDI data control. ARP1 will read an input from a MIDI keyboard and arpeggiate the chord (Major,Minor, Dominant7, Minor7) from the given root note it is receiving. ARP2 will read an input from a MIDI keyboard and arpeggio any chord combination held on the keyboard. For more information about the modes of operation, refer to the software section.

Switch 1	Switch 2	Function
OFF	OFF	ARP2
OFF	ON	ARP1
ON	OFF	CONTROL
ON	ON	PITCH

Table 11: Choosing Mode Operation in SenseBox

6.2.3 Algorithm Implementation

The following sections will explain how the ATmega328p captures the analog signal from the theremin and convert to a MIDI data through our algorithm implementation.

6.2.3.1 Arduino Language

Our primary reason why we have chosen ATmega328P as our microprocessor for SenseBox is because it can not only handle the non-linearity of the system (when deriving the values to track the analog pitch, which takes into account any discrepancies in timing), but also it can store a considerably large lookup table, specifically for our MIDI library. The MIDI library is used to display important texts, such as the current note name and its correspondent MIDI note to the LCD screen panel for the user. It is often convenient when working with large amounts of text such as this to store in the program memory of the ATmega328p[13]. The program memory has 16KB available. In the Arduino language, we used a built-in function called "strcpy P" to call up an array from the MIDI library, which will then store it into the SRAM memory (volatile memory) of the ATmega328p. This function is used when tracking the frequency from the theremin and outputting a MIDI data.

6.2.3.2 SenseBox Processflow

When turning on the converter, it will check to see if there is a MIDI device connecting to the SenseBox's MIDI IN port. If so, it will store any incoming MIDI data into the ATmega328p (this feature will be exclusively used in ARP1 and ARP2 mode. For more details, refer to the Mode Operation section). Recall that each MIDI device has 16 channels of MIDI stream to choose from; we selected channel 1 to be the default channel. This means that our assigned presets will be entirely on channel 1 (for future goals, we may add more channels to add different presets). Lastly, the SenseBox will choose which mode of operation to perform based on the toggle switch configuration it is currently in.

6.2.3.2.1 Mode Operation

Currently, we have four different modes of operations in our SenseBox: PITCH, CONTROL, ARP1 and ARP2. The following sections will explain the methods behind each mode operation.

6.2.3.2.1.1 PITCH Mode

As mentioned before, when the SenseBox is in PITCH mode, it will track the frequency and volume from the uWave Theremin and display and output the corresponding MIDI data. Our method of performing this task is as so: before operating this mode, the converter must sample the incoming signal from the uWave Theremin, which is obtained from the uWave Theremin's pitch preview output. When there are sufficient samples of the signal available, the frequency is calculated in average over the taken samples. This frequency is compared with the lowest threshold frequency in the library in order to identify which MIDI note number it corresponds to. Lastly, the SenseBox will receive analog signal from the uWave Theremin's output and convert it to MIDI volume data (the volume data can only take a value from 0 to 127, 127 being the max volume). When accomplished, it will combine both the MIDI note number and volume data and send it out in channel 1.

6.2.3.2.1.2 CONTROL Mode

Unlike PITCH mode, which sends a MIDI note and volume data, CONTROL mode will send, as it implies, a control data. Before operating in this mode, note that we have decided to assign port #35 in channel 1 as the default control data for SenseBox. In other words, if the user wants to control a specific parameter in the DAW through the uWave Theremin , the user simply must assign port #35 to whatever the user wants to control. In MIDI protocol, port#22-35 are undefined in most MIDI devices.

When the connection is met with the port and its destination, it will take a sample from the incoming frequency and that sample will be the magnitude of the controller data—it will emit a value from 0 to 127. As a result, controlling the magnitude of the control data is purely based on the hand's position with respect to the pitch antenna of the theremin.

6.2.3.2.1.3 ARP1 & 2 Mode

In music theory, an arpeggio is a musical technique where notes are played or sung based on a chord. Rather than ringing out simultaneously, notes are played sequentially, giving a harp-like effect. In ARP1 and 2 mode, these will allow the user to have that technique employed in the SenseBox. Note that these modes can only be used if there is a MIDI keyboard connecting with the SenseBox; the MIDI keyboard will dictate which note and chord type to play.

In ARP1 mode, it will read an input from a MIDI keyboard and arpeggiate the chord based on how many keys is pressed; the first input will be the root note and the following inputs (which has to be above the root note) will dictate which type of chord it will arpeggiate. The diagram below demonstrates the operation process of ARP1 mode:

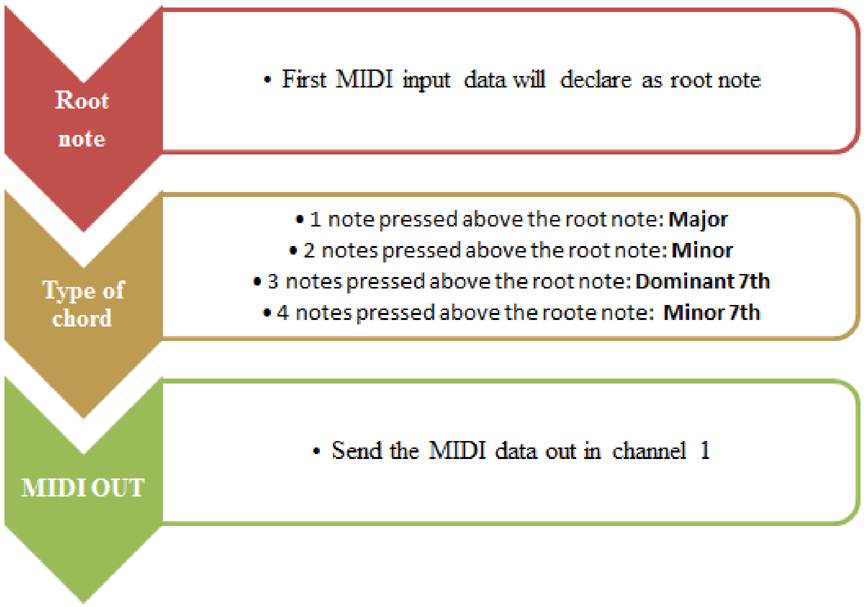

Figure 53: ARP1 Step by Step

For example, if the user presses and holds the note C as the first input, then the SenseBox will recognize that as the root note. Next, if the user presses and holds one note above C while still holding the C note, the converter will now appregiate C Major Chord, which consist of notes C, E and G. As soon as the user moves his hands around the pitch antenna, the SenseBox will only output those notes in channel 1. Alternatively, if the user presses three notes above C while still holding the C note, then the SenseBox will arpeggiate C Dominant 7^{th}, which consist of notes C, E, G and Bb.

In ARP2 mode, it will read inputs from a MIDI keyboard and construct an arpeggio around any collection of notes that are held on the keyboard. Unlike ARP1, ARP2 does not need to recognize a root note and it offers more control for the user to virtually arpeggiate any chords he or she desires. For instance, if the user holds the notes A, D, and E on the keyboard, the SenseBox will only output those notes; ARP2 only appregiates up to 12 notes at a time.

6.3 Expandome

6.3.1 Meeting Specifications and Requirements

The Expandome was designed to meet the criteria from the Specifications and Requirements section above. The design is capable of exhibiting the following features; at least three devices will be made, one 4x4, one 4x8, and one 8x8 grid device. These will have an *n* x *m* backlit push-button grid. The 8x8 device will be the primary Expandome that will connect directly to the computer, while the smaller devices can be added to expand the grid to allow the user to add beats and custom functions. In order to achieve these requirements set for our project, the Arduinome design, which was based on the Monome design, has been used as our reference design. The Arduinome's design was almost to the point of modularity, with the exception of the USB port that must be moved out of the way, and that rotating the devices was allowable, but had to be accounted for in the application or protocol. For example, in both MonomeSerial and ArduiomeSerial, the cable-orientation is selectable. This mostly-modular grid design allowed for multiple Monome-like devices to be cascaded programmically, but does not allow absolute flexibility in orientation, and plays a role in our decision to choose it as our reference design. The Arduinomes have a back light functionality that indicates button presses and that state the device is currently in. This gives the user the feedback needed during performance. The device runs on an Arduino microprocessor that allows for the inclusion of an SD card reading functionality and recognition by other devices like other Arduinomes and a computer. With this feature, files can be saved and manipulated to fit performance using the Arduinomes.

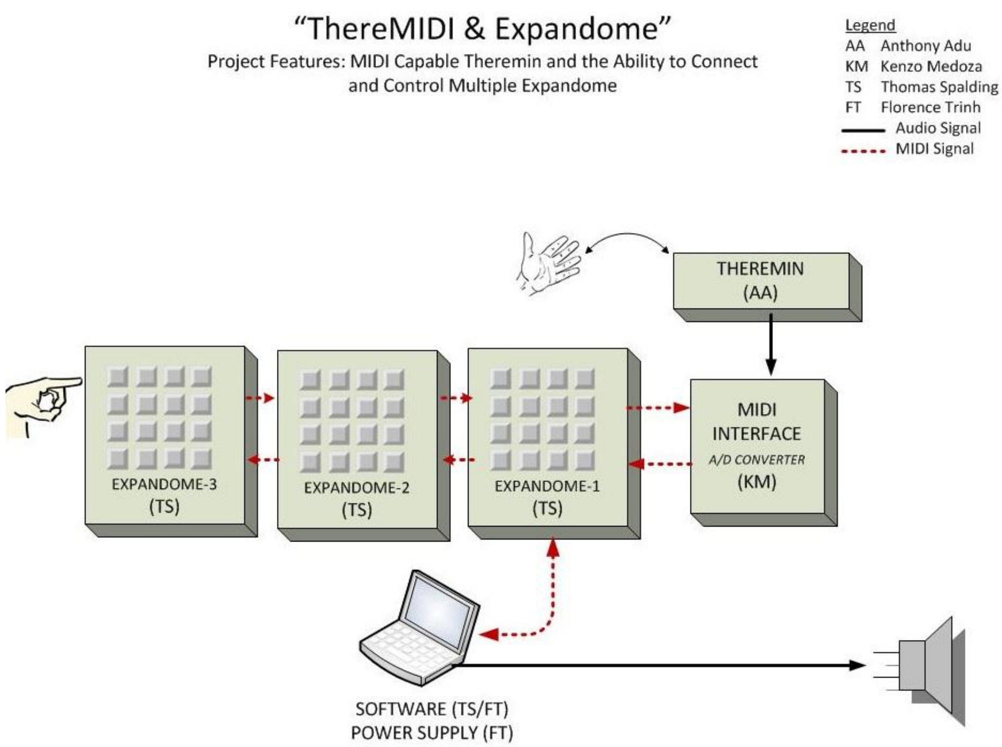

Figure 54: Design that was established based on specifications and requirements.

6.3.2 Hardware

6.3.2.1 Button Related

The device buttons and LEDs are an important feature of the project. Unlike the standard button pad selection, the LEDs can be chosen in a multitude of colors and brightness to customize each device. We discuss the selection in this section.

6.3.2.1.1 Button Pads and Buttonpad PCBs

The 4x4 button pads are made by SparkFun[26]. *'Each 4x4 button pad is made from translucent silicone rubber and is LED compatible. Each button has a conductive circle backing so that a switch can be created with exposed PCB traces. Button force is between 190 and 210 grams activation force.'* These can be combined to create any grid size for our *n* x *m* devices. The button pads are also PCBs made by SparkFun. They are compatible for both standard LEDs and RBG LEDs.

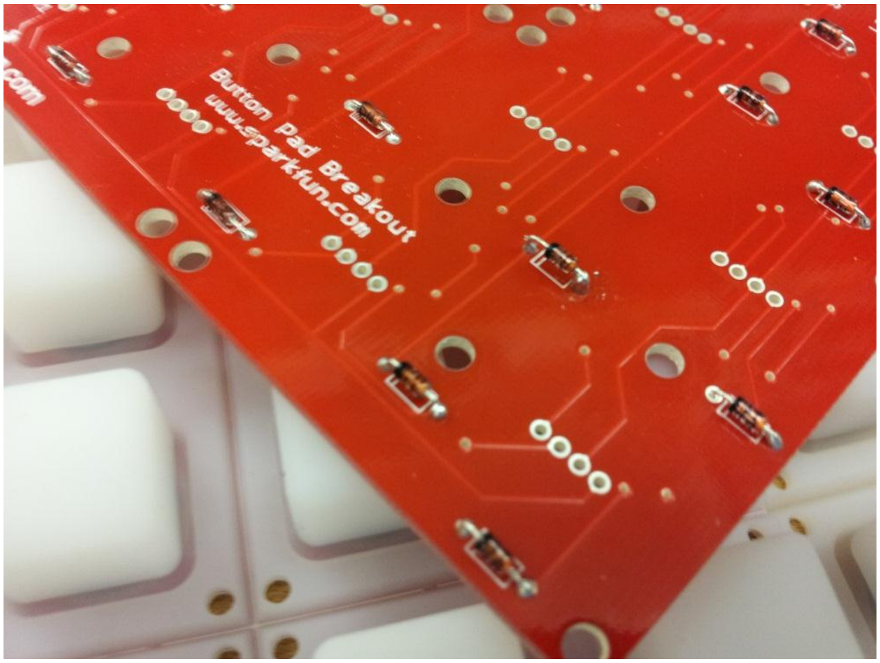

Figure 55: Sparkfun PCB after soldering on the diodes.

6.3.2.2 Microprocessor

The 8x8 Expandome device was designated as the master device when connected to multiple Expandome devices. This expandable feature requires additional input and output pins to reside on the Master device to allow communication to other devices. In addition to the need for more I/O connections, the Master device requires additional memory. For this reason, the 8x8 device may use the Atmega2560 which has 54 digital input/output pins, 16 analog inputs and 256k of flash memory.

On the other hand, the smaller devices such as the 4x4 and the 4x8 device does not require excess I/O ports or memory, therefor the design for the smaller devices will include the Atmega328 which has 14 digital input/output pins and has 32kB of flash memory. Both microprocessors have been incorporated into the PCB design circuit and require 5V DC power.

6.3.2.3 LED Related

Each Expandome device is comprised of an array of LEDs which is controlled by the microprocessor in conjunction with an LED Drivers and shift registers. The LEDs are the visual part of the project that allows the board to illuminate and indicate that a button has been pressed and the associated beat is played. In the reference designs it was discussed that there are numerous choices of LEDs that exist and are to be considered for power and desired illumination and color. The design can be customized to use multiple colors to make each device unique. The LEDs selected are discussed in this section.

6.3.2.3.1 LED Selection

Initial limitations set at the time of research allowed for 3mm, 500mcd - 2000mcd, about 50° or better viewing angle, below 4.5V forward voltage and below 20mA current draw. Further design considerations were only whether we preferred the color blue[47] or yellow[48] more. The blue LED is used for the initial prototyping, which had a 20mA draw and a 45° viewing angle.

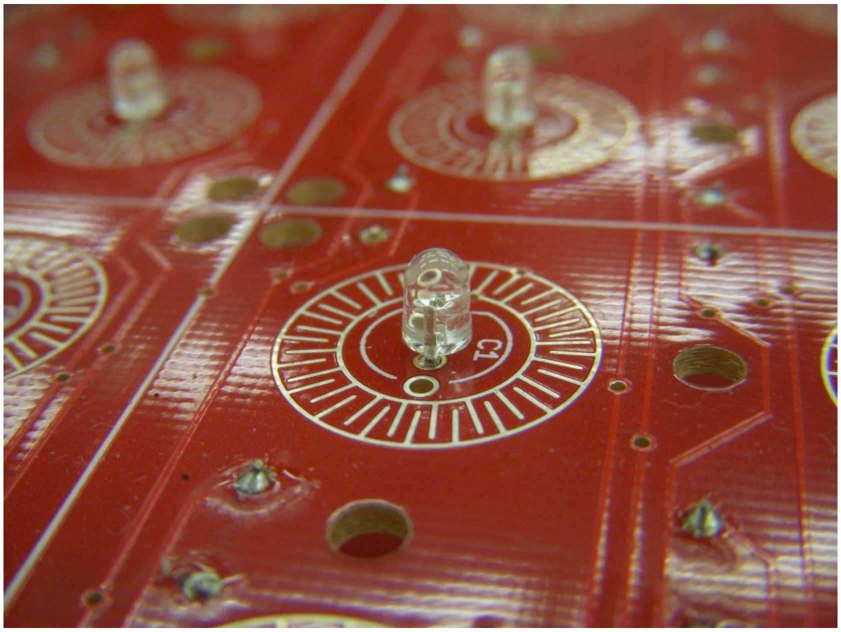

Figure 56: The blue LEDs soldered onto the button pad PCB

6.3.2.3.2 Rset

For the initial prototyping, the case of the two *Rset* values was not included; one high consumption, and one low to save power. This design is for slightly below maximum brightness of LEDs. Lower resistance corresponds to brighter LEDs, and higher resistance means lower luminosity and less power drain.

6.3.2.3.3 Shift Registers and LED Driver

The 74HC164 and 74HC165 shift registers are used to provide 8-bit serial-in/parallel-out and parallel-in/serial-out shift registers, respectively.

The use of an LED display driver is necessary to interface the microprocessor with the LEDs used on the Expandome module. The MAX7219CBG, manufactured by Maxim Integrated Products, is an integrated circuit which has an 8-digit LED display driver that has the capability to provide interface between the microprocessor and up to 64 individual LEDs. This integrated circuit also gives the capability to control brightness through analog and digital control. The MAX7219 requires +5V. For a single LED matrix, it is possible to use the +5V power supply from the microcontroller circuit, but more than three Expandome devices will require additional units to have secondary power over USB if an 8x8 device, or have batteries inserted if a 4x4 or 4x8 device.

The MAX7219 LED Driver is used in this design, as it has a proven precedent chosen by the Arduinome developers[4]. Garnered from the Arduino website, and the LED and LED Driver datasheet's *DC Forward Current* and *Forward Voltage*, *Rset* was chosen.

According to MAX7219 link, it gives estimates on maximum current and voltage out for USB ports, and LED power estimates for a 9V battery. We will implement a 9V battery for the smaller 4x4 and 4x8 devices, as they may not always be connected via USB to a computer.

Rset is chosen based on the LEDs' *DC Forward Current* ('the maximum current that is allowed to go through the LED without damaging it in the long run')[30]. This limits the total current draw of all LEDs in the *n* x *m* grid. For grids with *n* less than *m*, the lesser *n* row is chosen to be lit due to multiplexing, which results in half the current draw due to LEDs. Multiplexing allows for the rapid lighting of only a row at a time to give the illusion that the entire grid is lit.

I_{SEG} (mA)	V_{LED} (V)				
	1.5	**2.0**	**2.5**	**3.0**	**3.5**
40	12.2	11.8	11.0	10.0	9.69
30	17.8	17.1	15.8	15.0	14.0
20	29.8	28.0	25.9	24.5	22.6
10	66.7	63.7	59.3	55.4	51.2

Table 12: RSET vs Segment Current and LED Forward Voltage. (Referenced from Table 11 of MAX7219 datasheet.)

Due to multiplexing, only one row is lit at a time. The max current is calculated for an 8x8 device as 8x(~23.5mA current set with Rset) + ~10mA for MAX7219 chip itself, which gives approximately 200mA for the 8x8 Expandome device. Accordingly, the USB-hub on the computer will deliver up to 500mA on every USB-port, and notebooks might limit the max current to 100mA when running on batteries. Also, it is estimated that another 40mA will be taken for the microcontroller itself. It also estimates a 9V battery can supply all multiplexed LEDs for 55 minutes. Calculations for the estimated maximum current draw are shown in the below table.

Dimension	DC Forward Current (mA)	MAX7219 (mA)	ATmega (mA)	Total (mA)
4x4	20*4	10	40	140
4x8	20*4	10	40	140
8x8	20*8	10	40	210

Table 13: Estimated max current draw for the three Expandome devices.

As seen in the above table, each device will be more than covered by a typical computer USB port for power. In fact it is estimated that up to three full 8x8 gridded devices can be used in cascade from a computer. However, for this project have built one 4x4, one 4x8, and one 8x8 device for initial prototyping and this tally gives a total of 490mA out of a possible 500mA available. Not every LED shall be lit at the same time, and each smaller device has a battery pack that it is possible to switch to if not able to draw enough power.

6.3.2.4 Power Related

The power has been broken up into two different sections for the Expandome. The 8x8 master devices will receive its power directly from the USB connection to the computer. The USB port will supply 500mA to the device. Based on the LED calculations above and the known power requirements of the microcontroller, the USB power supply should be sufficient for power to the master device and any peripheral devices which may be attached by MIDI connection.

The second power design is for the smaller 4x4 and 4x8 devices. The power for the small devices will have battery capability and automatic transfer to the DC voltage supplied through the MIDI connectors, when connected to the master device.

6.3.2.4.1 Power Supply

8x8 Expandome Power by "USB":
Connect the Expandome to the computer with a USB cable. Use a 4-pin USB cable with standard A-to-B connectors. The USB has 4-pins: Pin-1 Vcc (+5V), Pin-2 Data-, Pin-3 Data+, and Pin-4 Ground. Connect Pin-1 to obtain the +5 volts and Pin-4 to connect to ground. The Expandome printed circuit board shall have a B type female USB connection. This allows us to power the LEDs and the microprocessor from the USB port on the computer. Ensure that the USB hub connection supplies a minimum of 500mA.

4x8 and 8x8 Expandome Power "MIDI and Battery Power with automatic selector":
Provide 2 power sources with an automatic selector. Install the AutoSwitching Power Mux TPS2110A. Connect one voltage input from a 9V power source and voltage regulator circuit for an input of 2.2V to 5V. Connect the second input voltage to the MIDI power pins. When the device is plugged into the Master device by MIDI connections, the power mux will automatically switch from battery to MIDI power. This selector may also be configured to switch from batter to USB power.

6.3.2.5 In/Out Components

Each Expandome prototype requires a USB connection to allow for programming the microcontroller and for the master device, power. Prototypes shall provide a B type USB connection on each Expandome printed circuit board.

The Master device communicates to the peripheral devices via MIDI connection. Install two female and two male 5-pin DIN MIDI connectors on each device. This allows for connection between devices. For aesthetics, flush mount connectors so that when devices are connected, they fit tightly together to appear as one unit.

The following connectors are used on the Expandome devices to transfer data and power:

- 5 Pin Din Female Panel Mount Jack – Locking: (Vetco Electronics PN: CAL-30-483) This 5 Pin Din Female Panel Mount Jack has a threaded ring that mates to our locking din plugs for a secure connection. Solid all metal construction. Overall Dimensions: .52" Tall x .75" Wide x 1.12" Long. Mounts in a 0.68" hole. This connector has solder cups 18-24 AWG wire.
- 5 Pin Din Male Inline Plug – Locking: (Vetco Electronics PN: CAL-30-458) This 5 Pin Din Male Inline Plug has a locking ring that mates to our panel mount locking din jacks for a secure connection. Solid all metal construction with a rubber strain relief. Dimensions: 2.28" Long x .76" Diameter. This connector disassembles to reveal solder cups 18-24 AWG wire.
- The small devices will have a 9V batter holder with 5.5x2.1mm, center-positive barrel jack. This 9V battery holder allows your battery to snap in tight and holds it in place, which is great in situations where you don't want the battery just hanging. It also has three mounting holes so you can attach it securely to your enclosure. It is terminated with a standard 5.5x2.1mm, center-positive barrel jack.

6.3.3 Software and Embedded Systems

6.3.3.1 Embedded Programs

6.3.3.1.1 Primary Expandome Routine

The primary Expandome routine was first based off of Owen Vallis' and Jordan Hochenbaum's firmware for the Arduinome[49] and Chronome[50]. The first iterations of our primary Expandome routine was first developed to include the expandable protocols for transferring data over the additional MIDI pins available on the MIDI connectors as well as read/write capabilities of the SD card and implementing the additional default programs. At a minimum the code implemented includes the following functions, declarations and capabilities:

Define the data ports and pin assignments of the following components:

- Shift Registers
- LED Driver

Define variables for the following:

- Hold incoming data.
- Store flags to indicate if a message has been received and the corresponding value.
- Temporary storage of incoming data from the button pad.
- Button array (8x8, 4x4, 4x8)
- Registers/instruction addresses on the LED Driver.

Declare functions:

- Send information to the LED Driver.
- Clear or zero out all button states.
- Check the current state of a button.
- Send LED status information to the LED driver

We used the following libraries during development:

- Wire Library - This library is used in conjunction with the I^2C chip to allow communication between two or more Expandome devices. Two analog or digital input pins can be used. One Expandome will be the Master Writer and the other will be the Slave Receiver.
- SD Library - This library is used to implement the feature of the Expandome that allows a user to read beats from a file and write capabilities to record a sequence of beats. The communication between the SD card and the ATmega chip uses SPI.
- SPI Library - This library is for communication between SPI devices such as the SD card.
- LED Control Library (for the MAX7219) - This library is used to allow the user to control of up to 64 LEDs on the Expandome devices. This may also be used to write commands to implement a power saving mode.

6.3.3.1.2 Default Programs and User Interface

The 8x8 Expandome devices implement an Atmel ATmega 2560 microprocessing chip, which has 256k of flash memory, and can also have an expandable SD card slot for additional memory. The SD card allows additional programs to be added to the main Expandome routine and interactive debugging through the Ether CMD, an embedded command line console. The 8x8 device may have additional Expandome oriented default programs on its microprocessor. The SD card will hold the additional device programs oriented towards the Theremin and SenseBox, as well as additional computer oriented features. The 4x4 and 4x8 devices do not have any default programs other than the primary Expandome routine.

The data messages that are passed through the devices are done on the two unused MIDI pins (three of five are reserved for just MIDI data messages). This accounts for all data messages sent between SenseBox-to-Expandome, Expandome-to-Expandome, and Expandome-to-computer.

Each message will begin with a prefix that says where the message is coming from for easier parsing and debugging. For example, all SenseBox messages will have the prefix "SB" in hexadecimal, i.e. 53 42.

Startup/Default program:
The Startup/Default program is initiated so that when the Expandome is powered on, all of the lights turn on in a sequence starting from the top row to the bottom row. This program allows the user to verify that all light emitting diodes are working properly.

Initialize Sequence:

- This program runs each time the device is loaded and turned on.
- All LEDs illuminates twice for 0.5 seconds ON, 0.5 seconds OFF and repeat.
- Beginning from the top row of the device illuminate, one at a time, each row for 0.1 seconds, than illuminate the next row. Begin at row 1 thru *n*.

- Beginning from the left column of the device will illuminate, one at a time, each row for 0.1 seconds, than illuminate the next column. Begin at column 1 thru *m*.
- Repeat Row and Column LED sequence 5 times.
- End program with all LEDs not illuminated.
- As each row is illuminated, a pre-programmed sound file will be played, different sound for each row and column. This will allow the user to verify that the sound is working properly and that sound files are present on the SD card.
- In the case where there are no beat files located on the SD card, than a monotone beep is sounded. This indicates an error in the SD card reading, but that the other functions are working.

Standard Program features:
Beat Button: Each program can assign a beat in the form of a .wav file to each button in the array. The user will press a button to turn on a beat once. The button will use both on and off button events. The beat or sound can be held if the button is held down to different effect.

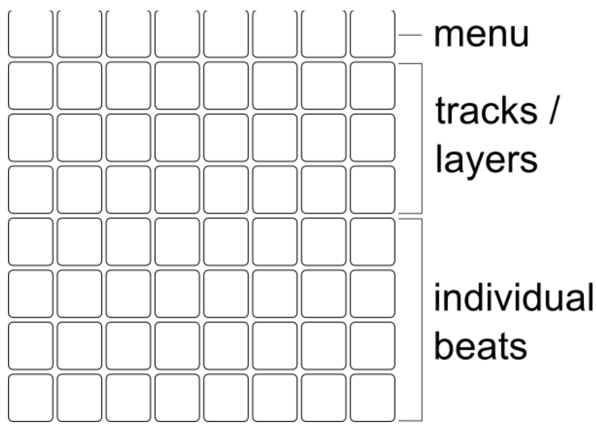

Figure 57: Typical grid menu and beat assignments

Each program includes a standard menu bar of four buttons along the top row, right hand corner of each matrix. These buttons shall be defined to allow the user to customize their music by repeating, recording, replaying and buttons which clear these settings from the board. The buttons will be programmed with the following functions:

- Beat Repeat: Each program can have an assigned button for the command Beat Repeat. When the user wants a beat to repeat indefinitely, the user will press this button before pressing the desired beat. Each button shall be illuminated when beat is being played.
- Clear Beat Repeat: Each program can have an assigned button for the command Clear Beat Re-peat. When the user would like to turn off or clear a beat which is currently repeating, the user will press this button before pressing the button which shall be turned off.
- Clear All: Each program can have an assigned button for the command Clear All. When the user would like to end or clear all beats that are on, they may press the Clear All button to end all beats. When the Clear all button is pressed, all buttons shall illuminate for one second and clear all beats and illuminated buttons to off.

- Record: Each program can have an assigned button for the command Record. The user may choose to record the sequence of buttons played. The user shall press the assigned record button to begin recording. When recording is complete, the users will press the record button again to stop recording. The record button shall stay illuminated for the duration of the recording and will turn off when recording has been stopped. The recorded sequence will be saved.
- Play Back: Each program can have an assigned button for the command Play Back. The user who has chosen record a music sequence may play back the last recorded event by choosing the assigned Play Back button. The Play Back button shall stay illuminated while the pre-recorded music is being played. The button will turn off when the music sequence has ended. The corresponding Play Back buttons will also illuminate.
- Track Assign (8x8 grid device only): For larger beat files, it may be more applicable to assign an entire row for the file like a track. When assigned, the LEDs will pulse as the track is played.

6.3.3.1.2.1 Expandome / EMI Browser

An Expandome browser, called EMI Browser, was developed to allow additional utilization of the SD card beyond general memory expansion. It allows for quick prototyping of programs and is a unique user interface that is another iterative development on the Arduinome's easy-grid concept. This browser parses through the file structure and automatically enumerates every file that begins with the letter 'e' and that takes the form of 'e####.lua'. Quick edits are now actually quick, no flashing onto any Atmel chips. The Expandome Browser, and subprograms, are being made using Löve Lua[51], a 2D game library for Lua. Screenshots from the Löve EMI browser are shown below. Concurrently developed, is an EMI Browser made in the Processing environment that largely contains the same functionality. Both graphical browsers use a Pure Data backbone, discussed in the Pure Data section.

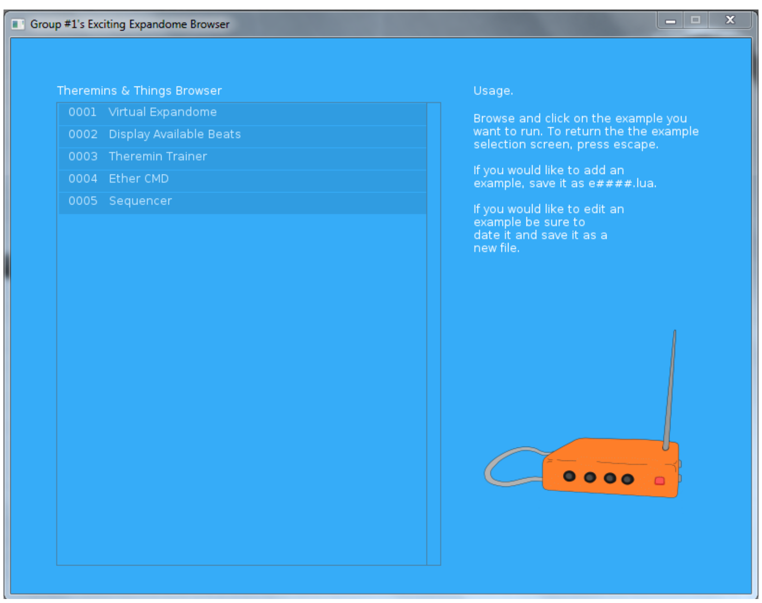

Figure 58: The Löve Lua implementation of the Expandome Browser main menu.

6.3.3.1.2.2 SenseBox Mode Selection

The SenseBox has four modes selectable; Pitch Mode, Controller Mode, Arp Mode 1 and Arp Mode 2. The Mode Selection program will display the current mode as well as give the option to change the mode.

Mode	Message (ASCII)	Message (HEX)
Mode 0	(EB) SB 0	45 42 53 42 30
Mode 1	(EB) SB 1	45 42 53 42 31
Mode 2	(EB) SB 2	45 42 53 42 32
Mode 3	(EB) SB 3	45 42 53 42 33

Table 14: SenseBox Mode Messages

The above table is of the SenseBox messages to the program which indicates which mode it is in is formatted with only the SB prefix. Messages from the program to the SB have the additional prefix of "EB".

6.3.3.1.2.3 SenseBox Data Display Console

On the right hand panel of the SenseBox subprogram is the Data Display Console. This displays all incoming and outgoing MIDI and Expandome data messages. If the user double-clicks on the Data Display Console, the left hand panel changes to the Ether CMD window. The messages are sorted based on the device prefixes.

When the Ether CMD console is used to interact with the SenseBox it has the ability to send MIDI messages manually for debugging purposes. It can also export all data messages to .txt file on the SD card. Space is pre-allocated, if there is less than 1kB then a warning is displayed before continuing.

6.3.3.1.2.4 Expandome

All Expandome messages will have "EX" with which number, in increments of 4 they are away from the primary Expandome device. Direction is also noted by 'U', 'D', 'L', or 'R'.

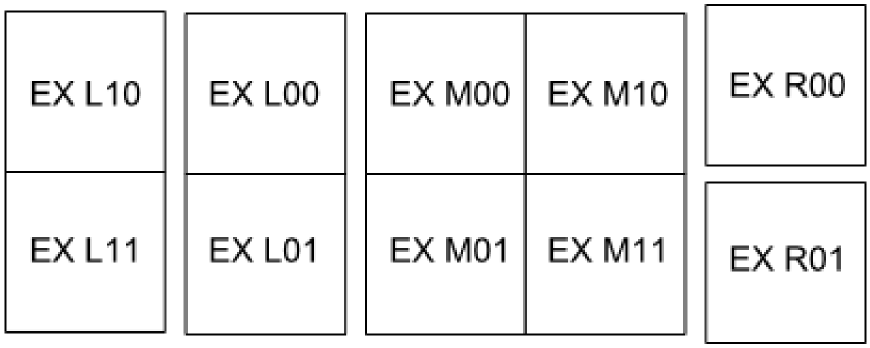

Figure 59: Expandome prefixes and orientation

4x8 Exp.	4x8 Exp.	Main 8x8 Expandome		Two 4x4 Exp.
EX L10	EX L00	EX M00	EX M10	EX R00
EX L11	EX L01	EX M01	EX M11	EX R01

Table 15: Expandome Prefixes and Orientation

The above table is an example of Expandome IDs (in ASCII) for a setup of two 4x8s to the left of the primary 8x8, and two 4x4s to the right of the primary 8x8, one on top of the other. Notice that their IDs are indistinguishable from a single 4x8.

6.3.3.1.2.5 Expandome Virtual Grid Display and Button/Row Assignment

A virtual Expandome device representation was made for button-assignments. Without an interface, the only way to assign beats is to manually edit the file structure on the SD card. This display is also created as a supplementary aid for debugging.

All lit button LEDs will be reflected on this display. Upon clicking a virtual button, the button will light and hold on the physical device, and the corollary LED should light on the Expandome. Also, when a physical Expandome button is pushed, the vice versa should occur on the virtual grid. Another use is for tracking down I/O errors. For example, if the Expandome behaves normal, such as when a button is pushed and it lights, but it is not reflected on the virtual display, then the user can know if an I/O error has occurred.

If a virtual instrument is already integrated with the instrument, the virtual grid button will behave the same as the Expandome, but slightly more indirect. Its message shall be sent to the Expandome, light the button, and make the Expandome repeat back the message as if it were physically pushed, then any typical programs or features are used as is the norm.

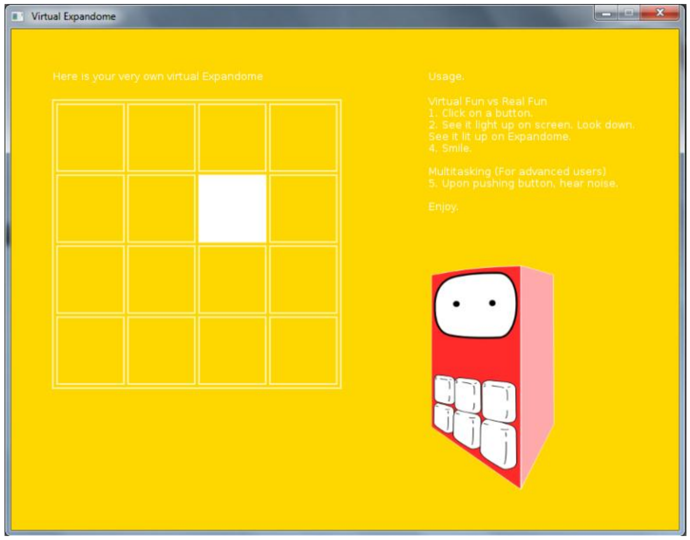

Figure 60: Screenshot of the Virtual Expandome when a virtual button is pushed. The corollary physical button on the Expandome should now be lit.

The grid size will be automatically resized as with all the Expandome devices connected. This is easily done by the devices' prefixes setting flags.

Expandome Button and Row Assignment:

To exhibit the Expandome in a standalone capacity, a Button and Row Assignment subprocedure was added to the Virtual Expandome. This assumes that no virtual instrument program is in use, although it probably would not impede the use of such a program, but would likely cause some inharmonious clashing of sound (unless of course that is what the user wishes).

When the user starts the Button and Row Assignment subprogram, the program first scans through the SD card's file structure as described in the following SD Card Reading section. The dropdown menu essentially rearranges the beat files into such a structure.

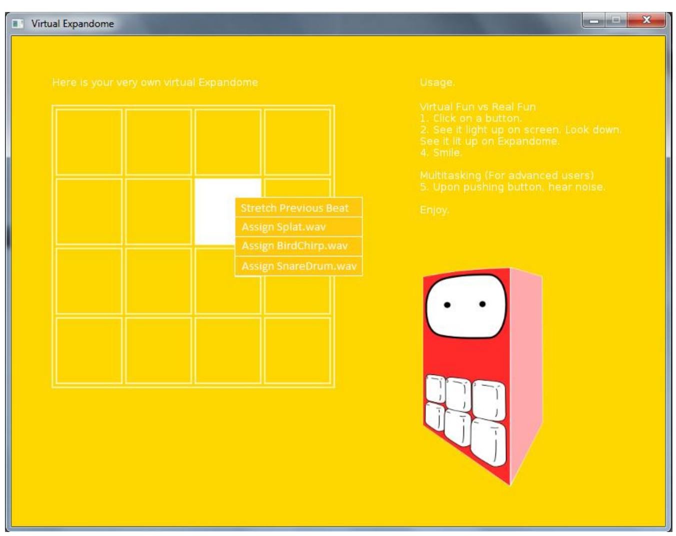

Figure 61: Upon rightclicking a virtual button, a dropdown menu is given to assign a beat from the SD card.

6.3.3.1.2.6 Expandome SD Card Reading

MIDI messages are capable of being recorded upon pressing the ‘record midi data messages’ button, or through using the Ether CMD. These messages are saved on the SD card. This is useful for debugging, recording a small song, or setup.

Beat files in the ‘Beats’ folder are automatically enumerated and displayed in the Expandome Browser’s Beats menu. When using the dropdown row assignments, these displayed beats are the ones that are selectable.

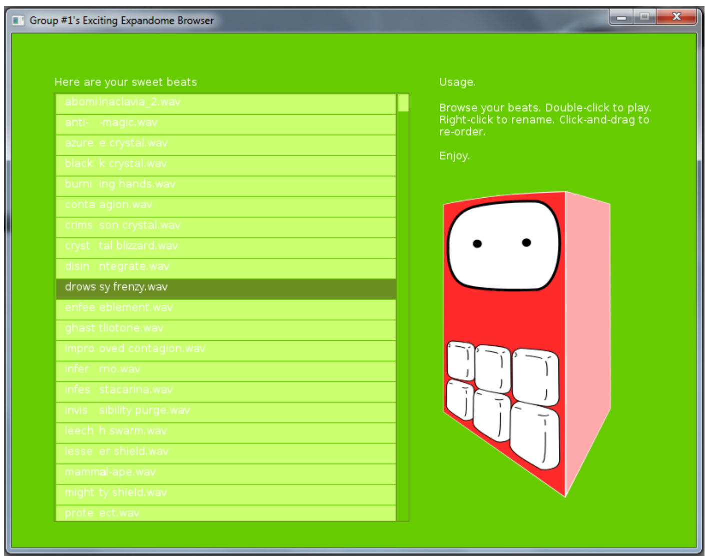

Figure 62: Screenshot of enumerated beats on the SD card.

Unused or available beats are stored in the ‘/Beats’ folder. The assigned files are moved to folders to represent each row of the Expandome, the auto-enumeration makes it clearer which button they are assigned to.

```
../Beats
    /Row1
      ├─b0000_exbeat.wav
      ├─b0001_superbeat.wav
      ├─b0002_etcbeat.wav
      └─b0003_applebeat.wav
    /Row2
      └─b0004_mytrack.wav
    /Row3
      ├─b0005_sample1.wav
      └─b0006_sample2.wav
    /Row4
      └─b0007_beatslicetrack.wav
```

Figure 63: File structure for beats on SD card

If the SD card is manually edited, the above figure shows the Beats folder structure that the Beat Button and Row Assignment uses. To sample the beat file from the generated list, the user may double-click on the file from the list to have it played.

6.3.3.1.2.7 uWave Theremin Pitch Trainer

A hard-coded message is used when connecting the uWave Theremin to the SenseBox. This is so that the default program Pitch Trainer can be opened automatically. The data message “UW” (55 57 HEX) is passed from the SenseBox directly to the computer, or also passed through the Expandome to the computer. If the uWave device is not recognized then the SenseBox program is instead opened by default. The Pitch Trainer program can of course be opened manually from the main menu if used with some other MIDI instrument, because it uses the typical MIDI pitch messages.

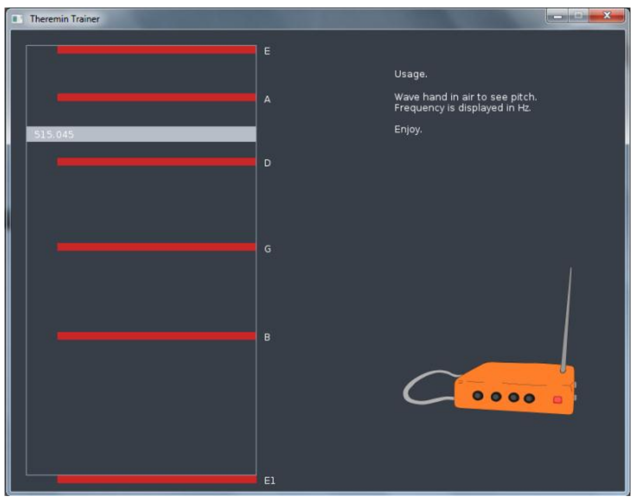

Figure 64: The pitch trainer with linear scale displayed.

Individual pitch notes can be selected or typical scales which are displayed as bars along the left in Hertz. The incoming MIDI pitch messages are rapidly updated to swing the arm to display where the pitch is. Additionally, the scale can be toggled between linear or logarithmic scale for user preference.

6.3.3.1.2.8 Ether CMD

For debugging purposes a commandline styled interface is used. In one implementation, it can display the code for the running programs, because the .lua files are not compiled, but are actually held in a zipped folder renamed from .zip to .love. If used as an editor, the program is automatically copied, dated and enumerated in the Expandome Browser. It can be immediately ran without any kind of reloading of the program.

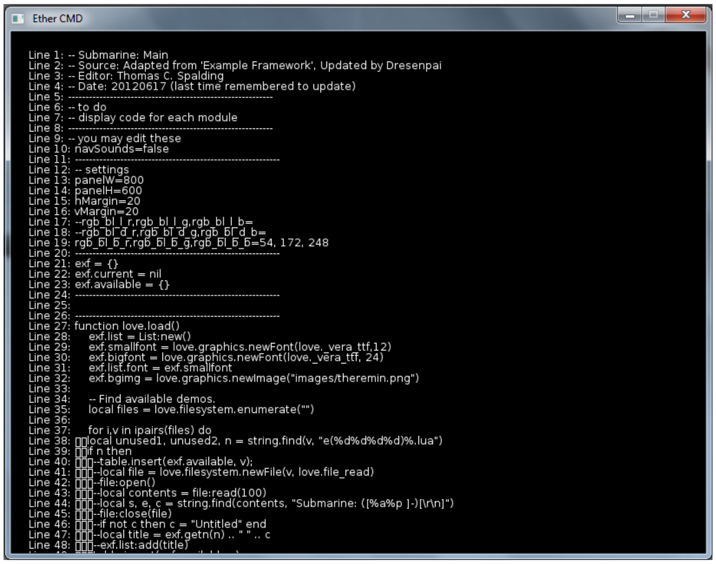

Figure 65: The Ether CMD displaying the main menu code with line numbers.

The Ether CMD can also be used for input and output messages for both MIDI data messages, and the inter-device data messages. This is highly useful for debugging purposes. Logs of all data messages between the instruments can be saved and exported to text using the typical command line carrot. Such an example is “EX M01 [midi msg] > mymessages.txt”.

6.3.3.1.2.9 *Pure Data Interface*

The SenseBox and Expandome devices are both Digital Audio Workstation capable. Any standard DAW will be able to interact with the devices. The EMI Browser comes with some example Pure Data setups that allow an extra flexible compatibility layer. Due to OSC’s ability to be sent over UDP, the theme of ease-in-expansion is additionally reinforced. A wide variety of apps and programs are able to interface with this group of devices including, but not limited to, Androidome[39], Control[41], and TouchOSC[40]. Future work will delve further into Libpd[52] as well as uOSC[36] for embeddable devices.

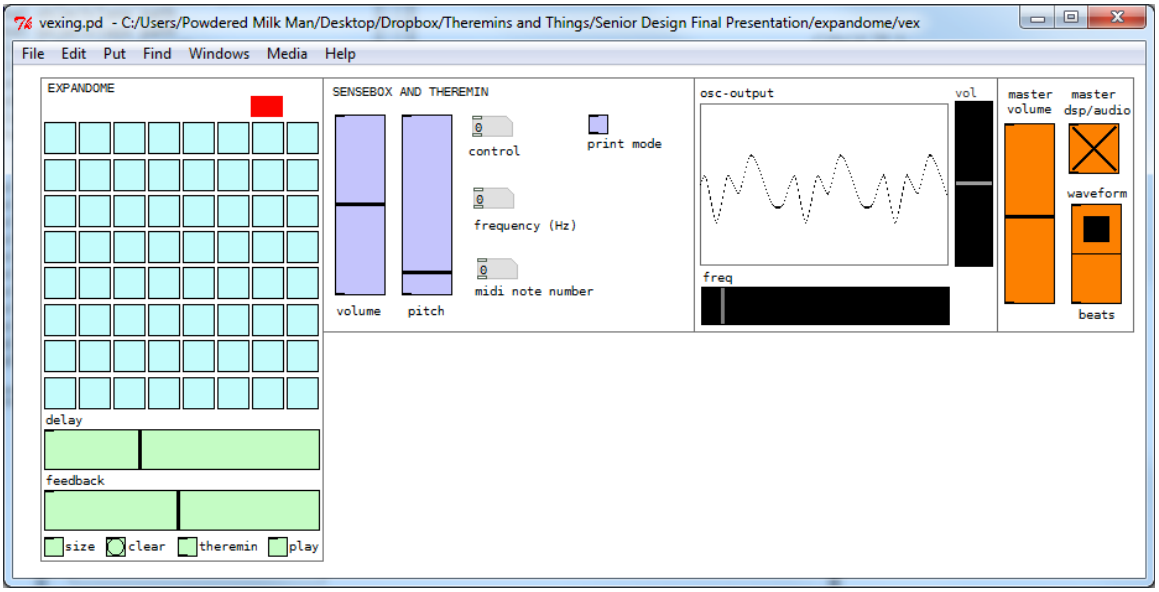

Figure 66: An example Pure Data patch that implements both a beat sequencer, theremin display, and waveform editor.

An example Pure Data patch for the virtual Expandome is shown above. It has a beat sequencer, theremin display that routes control over to the delay/feedback or vol/freq sliders, and a waveform shaper that was based on unsymbol's mnmewvtbl app[53].

6.3.3.2 Device Recognition & MIDI Software Interaction

Device recognition was achieved similarly as described by the Flipmu group's Flashing a new serial number via FTDI to mimic the monomer serial. Interaction is done through implementing a serial protocol called Open Sound Control[54], which is tailored for communication between computers and sound synthesizer type of devices over a network. From the Pure Data router, it is then easily joined with any DAW, MIDI controller, or other Pure Data or Max-esque patches.

When an 8x8 Expandome device is connected to the computer, the default Expandome Explorer window will popup. This can be used immediately without additional software installed, or ignored if the user wishes to use the device with a virtual instrument program. If a 4x4 or 4x8 device is connected via MIDI-to-USB cable, then the device will be recognized as a MIDI-capable device, but there are no default programs other than the devices primary Expandome routine. A virtual instrument program will be required in this case.

6.4 Power Supply

6.4.1 uWave Theremin

The theremin design requires a 12 volt dual power supply. After careful consideration of several possible power supply configurations, the option to provide AC power directly from the wall adapter connected to AC to DC converter circuit was the most feasible. The uWave Theremin is the only design element in this project which requires positive and negative 12 volts. A dedicated power source will ensure sufficient power to the device. The 12 volts will also be used to provide power the SenseBox.

Use a 16 volt AC power, 3-prong, wall adaptor with a minimum of 200 milliamp, 5.5 x 2.1mm, center-positive barrel jack. Connect this AC power to two half wave rectifiers and positive and negative voltage regulators to give +/-12 volts. This circuit will require the following components: two 1N4001 diodes, two 2200uF polarized electrolytic capacitors, tow 0.47uF capacitors, two 0.1uF capacitors, one LM78L12 regulator (+12V), and LM79L12 regulator (-12V). Install a PCB mounted female 5.5x2.1mm, center-positive barrel jack.

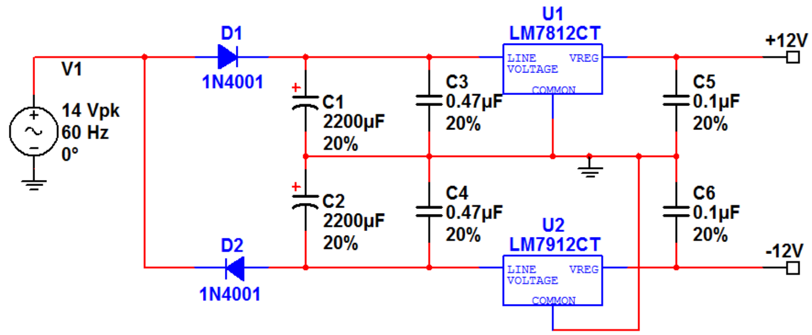

Figure 67: uWave Power Supply

6.4.2 SenseBox

The SenseBox converter will require +5 volts DC for the microcontroller and the LCD display. This device will be primarily used in conjunction with the uWave Theremin but will always be connected to the computer for programming and data transfer, because of this configuration, the USB connection will be used to provide a +5 volt DC power supply to the SenseBox device.

The SenseBox shall have a PCB mounted B type Female connector. Connect the SenseBox Pitch-to-midi converter to the computer with a USB cable. Use a 4-pin USB cable with standard A-to-B connectors. The USB has 4-pins: Pin1 Vcc(+5V), Pin2 Data-, Pin3 Data+, and Pin4 Ground. Connect Pin1 to obtain the +5 volts and Pin4 to connect to ground. The SenseBox printed circuit board shall have a B type female USB connection.

6.4.3 Expandome

The Expandome is unique such that one has the capability to connect multiple devices to the main device. Each device will require 5 volt DC power supply. The power configuration for the 8x8 device, which acts as the master device, is different from the design of the smaller devices such as the 4x4 and 4x8. After careful consideration of each power supply option, the design for the Expandome power is as follows.

8x8 Power Expandome power design will obtain +5V DC and ground through the USB connection to the computer. It contains one standard USB 2.0 cable with A-to-B Male/Male type connectors. Cable is a minimum of 1 foot in length. The Expandome printed circuit board has one B type female USB connection surface mounted. Use two pins for power; connect Pin-1 to obtain the +5 volts and Pin-4 to connect to ground. A standard 2.1mm DC jack female connector is surface mounted onto the Expandome printed circuit board.

4x4 and 4x8 Expandome power design has dual power sources. They will by default run on battery connection. When a viable power source over the MIDI power pin is detected, it will use its switching components to take this input.

6.5 Enclosure

The uWave Theremin enclosure shall be made of clear acrylic material. The dimensions shall be 5 inches in width, 4 inches deep, and 19 inches in length.

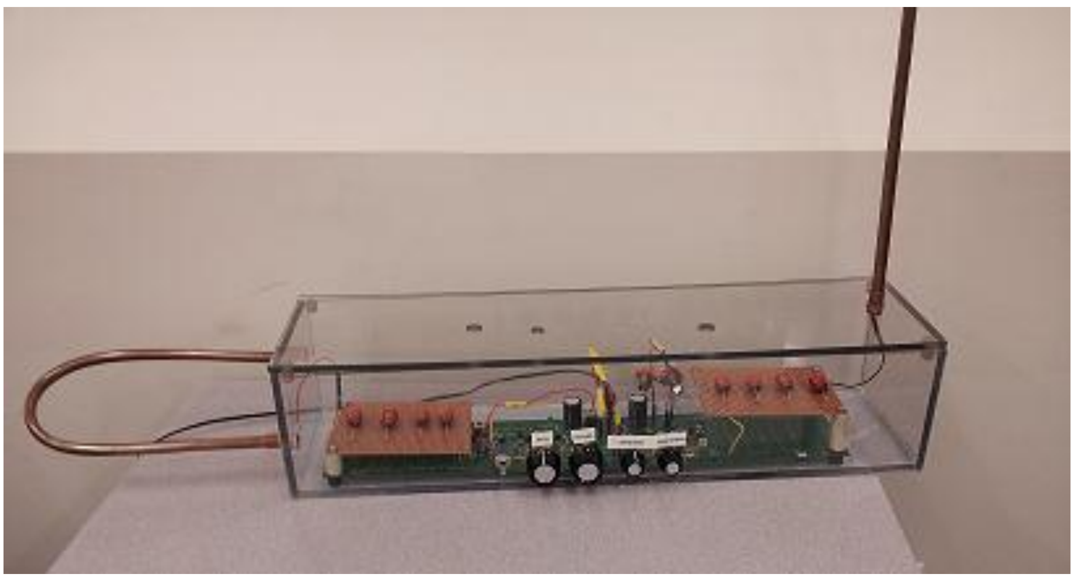

Figure 68: uWave Theremin and enclosure

There will be four knobs mounted on the front of the uWave Theremin. These knobs will be labeled Brightness, Waveform, Volume, and Pitch Tuning. On the back panel there will be three openings to pass cables; Audio, Power, Pitch Preview. The antennas are mounted on opposite sides of the uWave Theremin perpendicular to each other.

6.5.1 SenseBox

The SenseBox converter is made of clear acrylic. The dimensions shall be 6.5 inches in width and length and 3.5 inches deep. The LCD display is mounted on the top along with two toggle switches and two rows of LEDs.

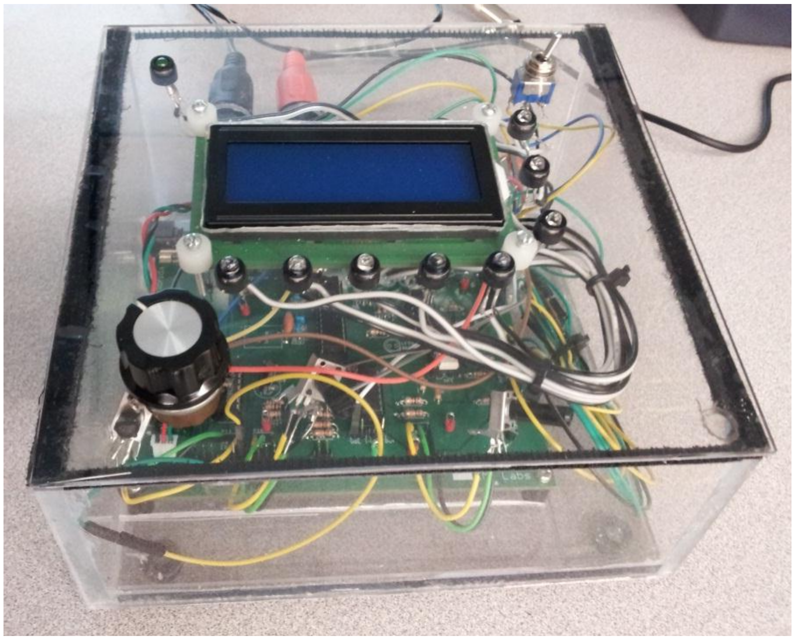

Figure 69: SenseBox and enclosure

The above, is the SenseBox in the final acrylic enclosure. The LED rows are placed such that they are perpendicular to each other; one represents volume level, 3 blue LEDs, and the other representing pitch, 5 LEDs. There are two openings for MIDI cords and one for the power cord.

6.5.2 Expandome

The Expandome device is made of acrylic, inlaid in white, and has a wooden grid faceplate. For each Expandome, there are two pairs of holes, spaced as depicted in Figure 16. The 4x4 and 4x8 devices' enclosures were made entirely with wood.

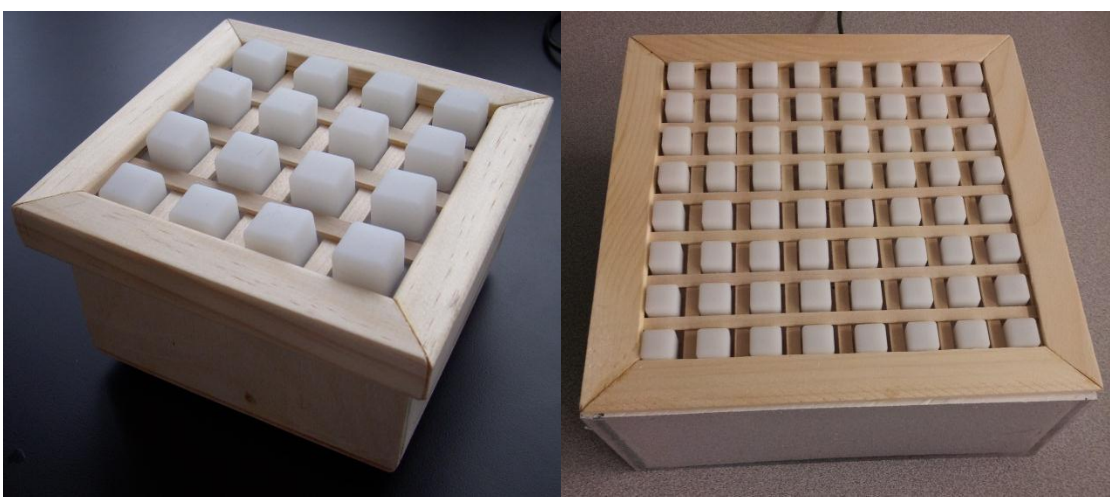

Figure 70: 4x4 and 8x8 Expandome device prototypes

The above images show our primary prototypes. For the 8x8 grid devices there is a hole center-spaced on the 'top' side of the enclosure. For the 4x4 and 4x8 grid devices the bottom plate will have a 3.5 x 3.5 removable plate for the battery enclosure. On each device there is a single hole on the 'bottom' side of the enclosure on the far right, of a ¼" diameter for an 'on/off' switch.

6.6 Design Summary and Bill of Materials

This integrated musical system includes three devices; the uWave Theremin, the SenseBox and the Expandome. These three components can be operated together or independently during musical performance. The uWave Theremin and the SenseBox interact closely with each other to produce musical notes that can be sent to the computer and manipulated using the EMI Browser or a DAW, such as Reaper, running on an external computer. The Expandome can act as a MIDI controller used to interact with the Reaper software during musical performance to set things like record, loops, enable or disable tracks etc. A functional block diagram showing how interaction between each device will be performed is below.

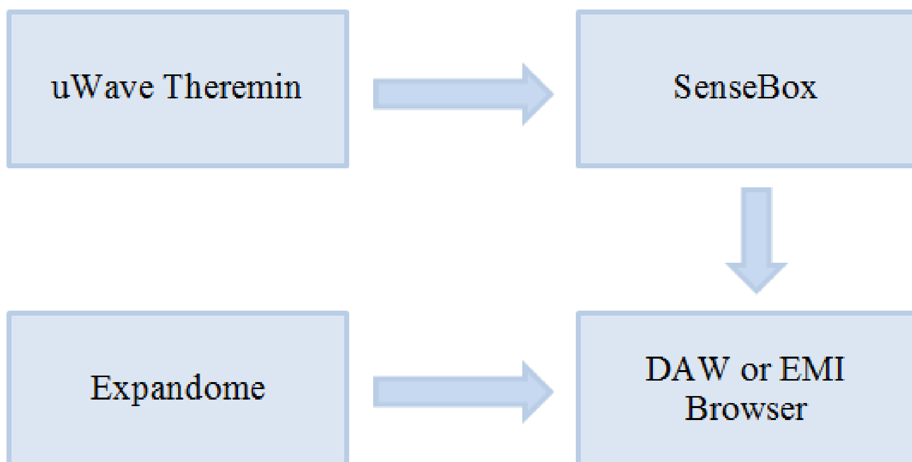

Figure 71: Flow of full interaction between devices.

The uWave Theremin includes the following passive and active electronic components:

uWave Theremin Electronic Components

Integrated Circuits

Designation	Description	Qty
LM7812CT	12V positive regulator	1
LM79L12	Negative Regulator	1
LM13700N-ND	Dual operational transcondance amplifier	1

Transistors

Designation	Description	Qty
Q1 - Q8	2N3904 NPN	8

Diodes

Designation	Description	Qty
D1,D2,D4	1N4148 Signal Diode	3
	1N4001 Power Diode	2

Capacitors

Designation	Description	Qty
C2,C11	3,900 pF/50V, 5%, polypropylene or polystyrene	2
C10,C14,C24	22 pF/50V, 5%, NPO (zero temperature coefficient) ceramic	3
C3,C7,C12,C20	0.01 µF/50V, 10%, polyester	4
C8,C9,C13,C19,C22,C23,C26,C27	1.0uF/35V tantalum	8
C25	33pF/50V, 5% NPO (zero temperature coefficient) ceramic	1
C4	1,000 pF/50V, 10%, ceramic	1
C6	2,200 pF/50V, 5%, polypropylene or polystyrene	1
C21	0.1 uF/50V ceramic	3
C15	4,700 pF/50V, 10%, ceramic	1
C18	3,300 pF/50V, 10%, ceramic	1

Inductors

Designation	Description	Qty
L1,L2,L3,L5	10 mH, 3-section, RIF choke (J.W. Miller #6306)	4
L4,L6	100pH, hi-Q, variable inductor (Toko RWRS-T1015Z)	2
L9,L10	2.5mH, 3-section, RIF choke (J.W. Miller #6302)	2
L11,L12	5mH, 3-section, RF choke (J.W. Miller #6304)	2
L13	62.8uH, hi-Q, variable inductor (Toko 154ANS-T1019Z)	1

Potentiometers

Designation	Description	
R43,R44	5kΩ linear taper, cermet or conductive plastic (Clarostat 53C1-5K or equivalent)	2
R45,R46	50kΩ linear taper	2

Resistors (1/4 W, 5%, metal or carbon film)

Designation	Description	QTY
R6,R7,R10,R11	1 Kohm	4
R26,R27	1.8 Mohm	2
R13,R30,R32,R38,R39,R40	10 Kohm	6
R19	100 Kohm	2
R17	150 Kohm	1
R4,R8,R12,R33,R41	2.2 Kohm	5
R31	2.7 Kohm	1
R35,R37	33 ohm	2
R16	330 Kohm	1
R14,R18,R22,R23,R24,R25	4.7 Kohm	6
R28	4.7 Mohm	1
R1,R9,R21,R29	47 Kohm	4
R3,R5,R34,R36	470 ohm	4
R2	560 Kohm	1

The uWave Theremin is designed with knobs and connectors which will allow the user to manually tune the instrument and make simple connections to the computer, amplifier and power source. The following items are included in the enclosure design to accommodate these features.

uWave Theremin Enclosure Hardware and Connectors	
Switch	
Description	**QTY**
SPST miniature power switch	1
Connectors	
Description	**QTY**
Insulated 1/4-inch phone jack (Switchcraft N-111 or equivalent)	1
3.5mm phone jack (Switchcraft 41 or equivalent)	1
Parts Not on Schematic	
Description	**QTY**
16-pin IC socket for U3	1
Connector set with at least ten conductors for connections between the main circuit board and front panel	1
Wall-wart transformer to provide 12 to 15 ILIAC with at least 200mA (Cui-Stack DPA 120020-P1-SZ)	1
Two large knobs for PC and P2	2
Two small knobs for P3 and P4	2
Two 24-to 36-inch x 3/8-inch straight copper tubes for antennas	2
Tube bender for volume antenna	1
Atlas AD-AA B microphone-stand mounting flange	1
10-inch x 4-inch prototyping circuit board	1
Two 4-inch x 3-inch prototyping boards for the antenna circuits	2

The design for the Expandome devices includes the following passive and active electronic components:

EXPANDOME Electronic Components		8x8	4x8	4x4
Microcontroller				
Part #	**Description**	**Qty**	**Qty**	**Qty**
ATmega328	Atmel AVR 8-Bit Microcontroller 32K Bytes In-System Programmable Flash		1	1
Diodes and Capacitors				
Part #	**Description**	**Qty**	**Qty**	**Qty**
COM-08588	Diode Small Signal - 1N4148	64	32	16
COM-08375	Capacitor Ceramic .1uF	3	3	3
LEDS	Various (Blue/Yellow)	64	32	16
P828-ND	10UF 50V Mine Alum Elect (KA)	1	1	1
Integrated Circuits				
Part #	**Description**	**Qty**	**Qty**	**Qty**
MAX7219CNG+	IC DRIVER LED DISPLAY 8DGT 24DIP	1	1	1
MM74HC164N	IC REGISTER PAR-OUT 8BIT 14-DIP	1	1	1
MM74HC165N	IC REGIST PAR-IN/SER-OUT 16-DIP	1	1	1
Other				
Part #	**Description**	**Qty**	**Qty**	**Qty**
ED90053-ND	IC SOCKET 24PINMSTIN/TIN .300 (socket for MAX7219)	1	1	1
ED90049-ND	IC SOCKET 14PINMSTIN/TIN .300 (socket for 74HC164n)	1	1	1
ED90050-ND	IC SOCKET 16PINMSTIN/TIN .300 (socket for 74HC165n)	1	1	1
770-101-R100KP-ND	RES NET 9RES 100K OHM 10PIN	1	1	1
S1012E-36-ND	CONN HEADER .100 SINGL STR 36POS	2	2	2
ASC16H-ND	CONN IDC SOCKET 16POS 15 GOLD	4	4	4
AE16G-5-ND	CABLE 16 COND 5′ GRAY RIBBON	1	1	1
RESISTOR	370k Ohms	1	1	1

The Expandome enclosure includes the following hardware for functionality and connection to the computer and other devices:

EXPANDOME Enclosure Hardware and Connectors				
Boards and Breakouts for Development and Testing				
Part #	**Description**	**Qty**	**Qty**	**Qty**
COM-08033	Button Pad 4x4 - Breakout PCB	4	2	1
COM-07835	Button Pad 4x4 - LED Compatible	4	2	1
Cables: Power & Data Transfer				
Part #	**Description**	**Qty**	**Qty**	**Qty**
CAB-00512	USB A-to-B Connection	1		
CAL-30-483	5 Pin Din Female Panel Mount Jack	2	2	2
CAL-30-458	5 Pin Din Male Inline Plug	2	2	2
PRT-10512	9V Battery Holder to Barrel Jack Adapter	1	1	1

The SenseBox requires the following passive and active electronic components:

SenseBox Electronic Components		
Capacitors		
Designation	**Description**	**Qty**
C1,C2	22pf (option)	2
C3,C4,C5,C7	.1uF	4
C8	.1uF	1
C12,C13	.1uF	2
C6,C9	10uF, polarized	2
C10	47uF, 25 V, polarized	1
C11	10uF, 16V, polarized	1
Resistors		
Designation	**Description**	**Qty**
R12,R13,R14,R15,R16,R17,R18,R19	150	8
R4,R5	220	2
R21,R26	1k	2
R27	1k	1
R3	3.3k	1
R1,R6,R7,R9,R24,R25	10k	6
R10,R11	10k Potentiometer	2
R8	20k	1
R2,R22,R23	100k	3
R20	1M	1
Diodes		
Designation	**Description**	**Qty**
D1,D2	Diode	2
D3	Green 3mm LED	1
D4	1N4001	1
LED1-9	3 red, 2 yellow, 1 green, 3 orange 3mm LEDs	9
Integrated Circuits		
Designation	**Description**	**Qty**
IC1	ATMEGA328-PU 28 DIP	1
IC2	LM324N OPAMP	1
IC3	7805 5V Regulator	1
V1	74595N Output Latch	1
OK1	4N25M	1
C10,C11	?? unknown cap	2
Other		
Designation	**Description**	**Qty**
Q1	16MHz Crystal (option), ceramic resonator	1
OK1	Phototransistor Optocoupler	1
Optional		
Designation	**Description**	**Qty**
LCD	4 line LCD panel	1
LCD 117	LCD Serial Driver, Drives an LCD panel from one serial output line	1

The SenseBox enclosure includes the following hardware for functionality and connection to the computer and other devices:

SenseBox Enclosure Hardware and Connectors

Interconnections, sockets and cables

Designation	Description	Qty
S1	4-pin Omron (momentary?) switch for RESET	1
SV1	3-header for EXT	1
	6 pin IC socket	1
	14 pin IC socket	1
	16 pin IC socket	1
	28 pin IC socket	1
M1, M2	5 pin DIN MIDI Socket	2
PP1, Vol1	1/8" Audio Socket (mono, TR type)	2
	USB-to-FTDI Cable	1
J1	2.1mm Jack	1

SECTION 7.0 PROTOTYPING & BUILD

7.1 PCB Design

All devices and their subsystems are mounted on a PCB. There are multiple directions that could be taken when it comes to PCB routing, fabrication and mounting. In order to design the PCBs, the group chose to use the EAGLE PCB software which can be downloaded at www.cadsoftusa.com. This software was used to create the board layout and the schematic drawings of the prototype microcontroller and the uWave Theremin circuitry. The printed circuit boards were ordered from www.4PCB.com website. In order to be eligible for the student rate of $33, our boards needed to be within 72sq-inchesand have a maximum of 2 layers. The parts for the Expandome were purchased from various venders such as Digi-Key, Newark, SparkFun, Jameco, and Texas Instruments. All soldering, including parts that required surface mounting techniques were done by the group members.

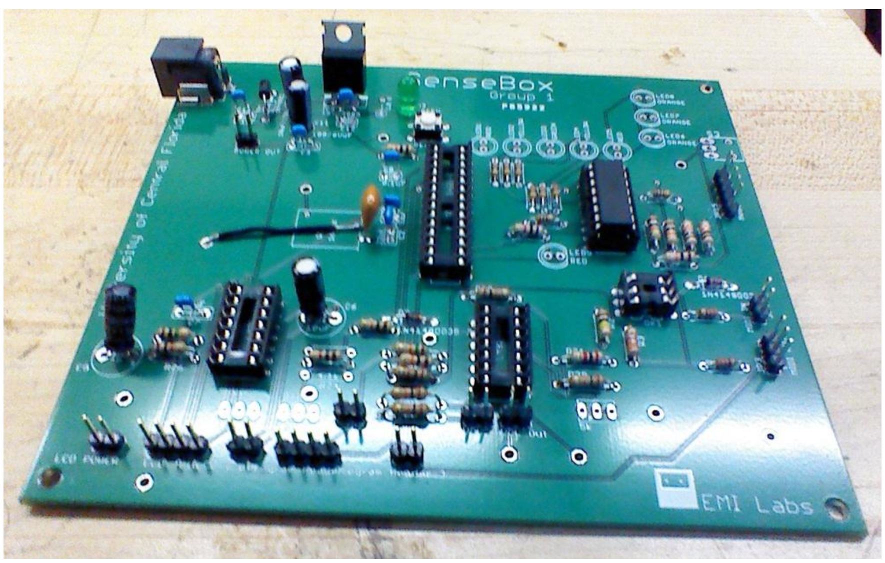

Figure 72: SenseBox board

7.2 uWave Theremin Construction Plans

Since we are modeling our uWave Theremin similar to that of Robert Moog's design, much of the information on how it will be constructed can be seen in his DIY paper. This section will discusses the plan the group will take in order to construct a uWave Theremin that fits the scope of this project. We will investigate different vendors for aspects that pertain to PCB construction, enclosure construction, tube antennas, and other materials needed for uWave Theremin construction.

7.2.1 Tube Antennas

As mentioned in the design portion of this paper, the group decided to use soft copper tubes for the antennas. These tubes can be found at local hardware stores like Home Depot and Lowes. A 10 feet soft copper tube can be purchased from home depot while a tube bender can be purchased at Lowes for about $20 and $15 respectively. A simple hacksaw was used to cut the tubes to the dimensions required for the project. Using the tube bender, we bend the volume antenna in order to give it an arc shape and a length of 9 inches. The ends of the bent volume antenna are separated 3 ¼ inches from each other. The full CAD drawing of the pitch and volume antenna can be seen in Figure 73 below.

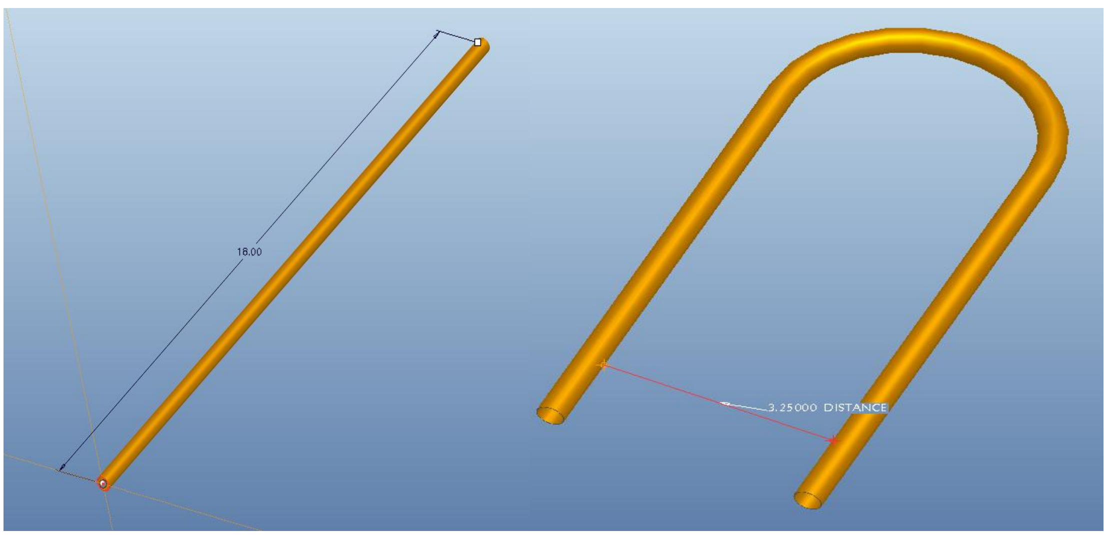

Figure 73: CAD drawing of pitch antenna (left) and volume antenna (right) with dimension in inches.

To connect the antennas to the enclosure of the uWave Theremin, Moog suggests using tube-to-pipe connectors. These can also be found at local hardware stores like Lowes and Home Depot. The connectors of interest should be connectors that allows for easy removal and insertion of different tube antennas with a diameter of 3/8 inches. An option from swagelok.com can be seen below.

Figure 74: Tube-to-pipe connectors used to connect tube antennas to uWave Theremin enclosure.

As mentioned in the research section, the tube material shouldn't have much impact on the antenna circuits, which opens room for the possibility of experimenting with different tube materials to judge the appearance of the uWave Theremin.

7.2.2 PCB Discussion (uWave Theremin)

The components of the uWave Theremin are all mounted on a PCB with the exception of the inductors which had to be mounted on a perfboard in order to reduce effects of conductive materials on the PCB. In some uWave Theremin forums[10] it was advised that the antenna circuits shouldn't be mounted on a breadboard because the copper pattern can add unwanted capacitance to the circuit. For testing purposes, the group mounted the antenna circuitry on perfboards and the remaining components on a breadboard.

From the dimensions seen in Figure 76, the uWave Theremin is 19 inches x 4 inches. This gives us an estimate on what the size of the uWave Theremin PCB would be. The uWave Theremin PCB constructed has a dimension of 16 x 3 inches.

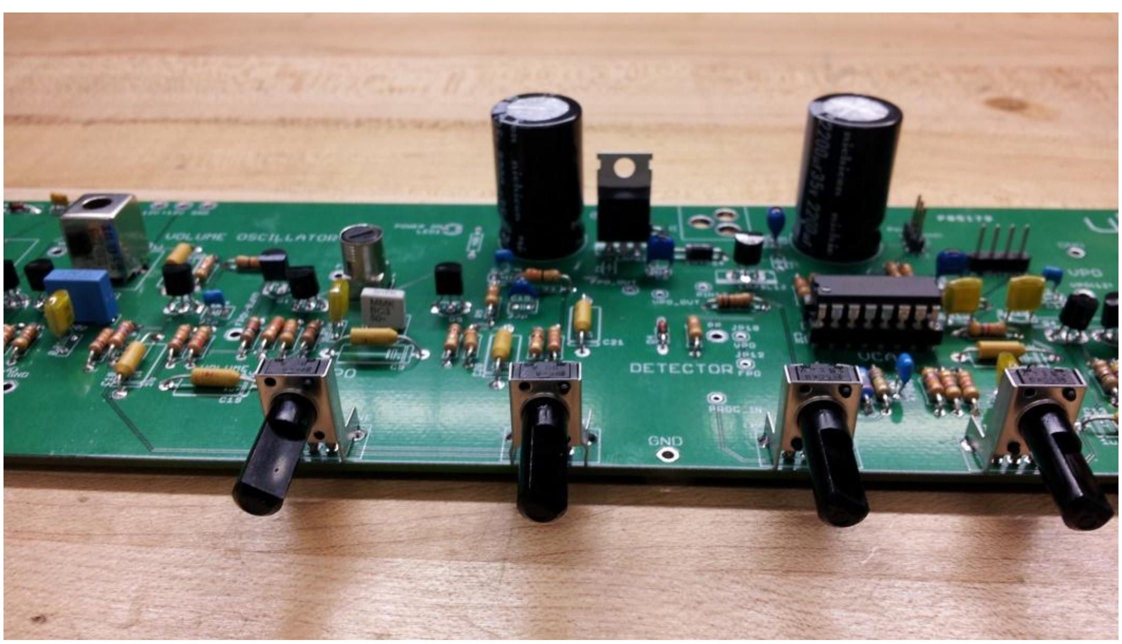

Figure 75: uWave Theremin PCB

The pitch and volume antenna circuitry are mounted as close as possible to their respective antennas. The inductors on the antenna circuitry are mounted as seen in Figure 21 and Figure 23, one inch from each other. The VPO and FPO are separated by only a couple of inches so they synchronize at low beat frequencies. C8, C9, and C13 are placed very close to their respective oscillator circuits in order to maximize decoupling[8].

7.2.3 Build Discussion

The uWave was made of clear acrylic. The enclosure is a custom size and was ordered from Instant Display Cases[55] with the dimensions provided below. The holes and openings for the knobs, antennas, power and audio connections will be pre-cut by the manufacturer as described in the drawings.

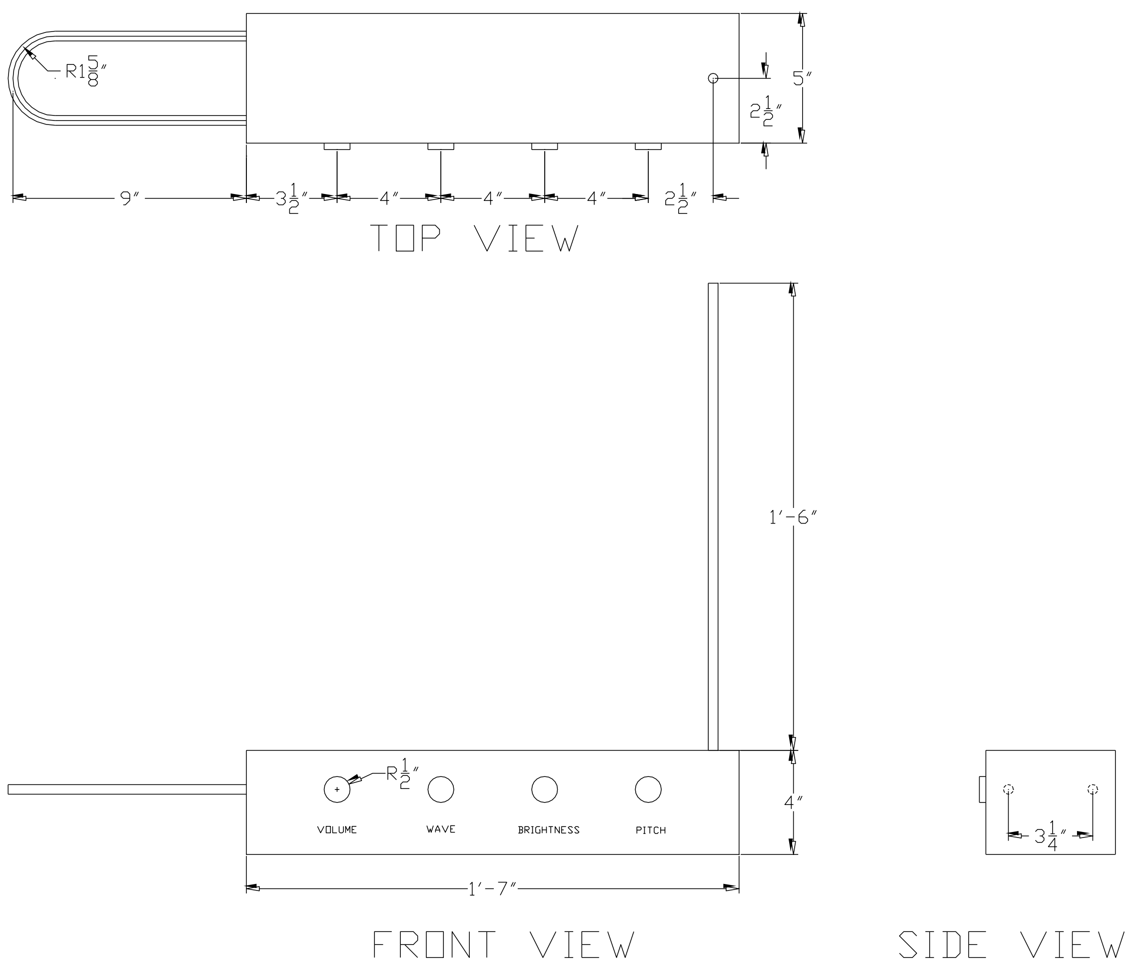

Figure 76: uWave Enclosure Dimensional Drawing

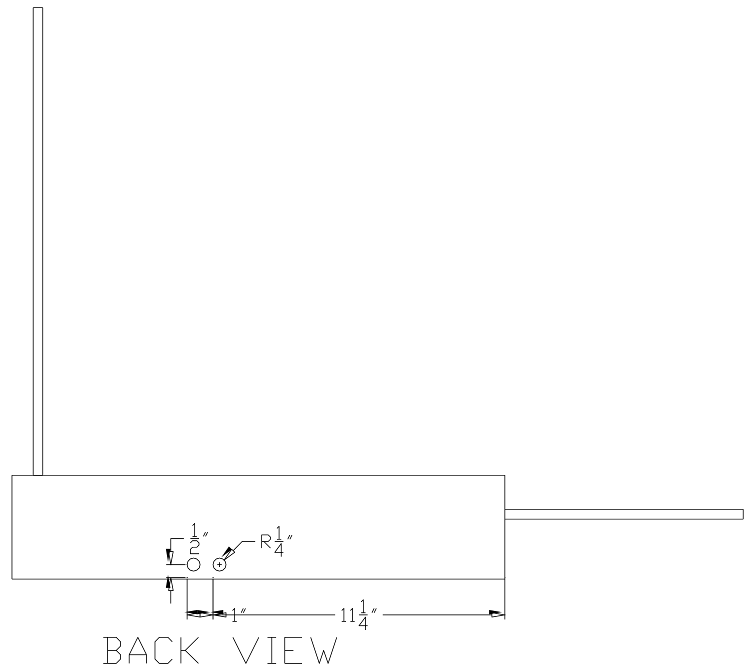

Figure 77: uWave Enclosure, Rear View

7.3 SenseBox

7.3.1 Build Discussion

The SenseBox enclosures will be made of clear acrylic with a wooden base. The enclosures will be custom ordered from Instant Display Cases[55] with the dimensions provided below. The enclosure will include openings and holes for the LEDs, Switches and LCD mounts as indicated in the drawings.

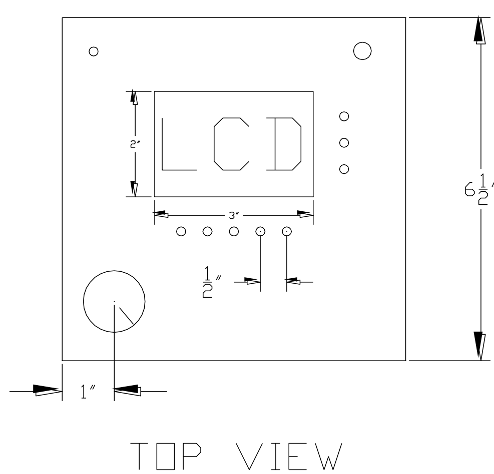

Figure 78: SenseBox Enclosure, Top View

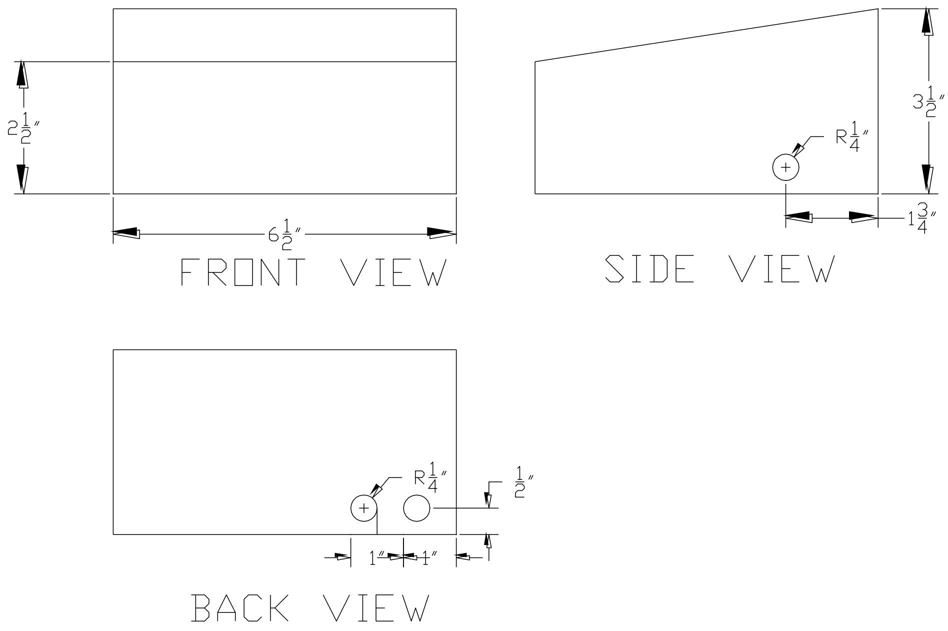

Figure 79: SenseBox Enclosure

7.4 Expandome

7.4.1 PCB and Routing

Each Expandome is designed such that all hardware components are mounted to one main printed circuit board (PCB). The dimensions of the PCB are no greater than 6 inches by 10 inches. The printed circuit board is laid out using the free version of the EagleCad PCB Design Software which can be downloaded at www.cadsoftusa.com. This software was used to create the board layout and the schematic drawings of the prototype microcontroller. The printed circuit board was ordered from www.4PCB.com website. The maximum size to be eligible for the student rate of $33 is 6 inches by 10 inches and shall have a maximum of 2 layers.

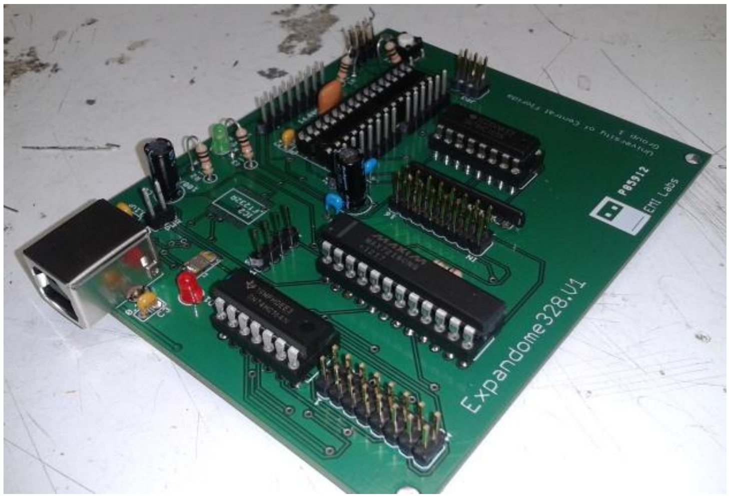

Figure 80: Expandome board

The parts for the Expandome were purchased from various venders such as Digi-Key, Sparkfun, Jameco, and Texas Instrument. The parts were surface mounted by members of the team where possible. Components may be mounted by Quality Manufacturing Services, if the budget and schedule permits. Each Expandome device required a custom PCB which includes a microcontroller, power supply circuit, LED driver, shift registers, I^2C and MIDI connections. The dimensions of each board will not exceed the dimensions of the enclosure

7.4.2 Build Discussion

The uWave, SenseBox and Expandome enclosures were made of clear acrylic. The Expandome had wooden faceplates. The enclosures were custom made using materials from the local hardware store with the dimensions depicted in Figure 83.

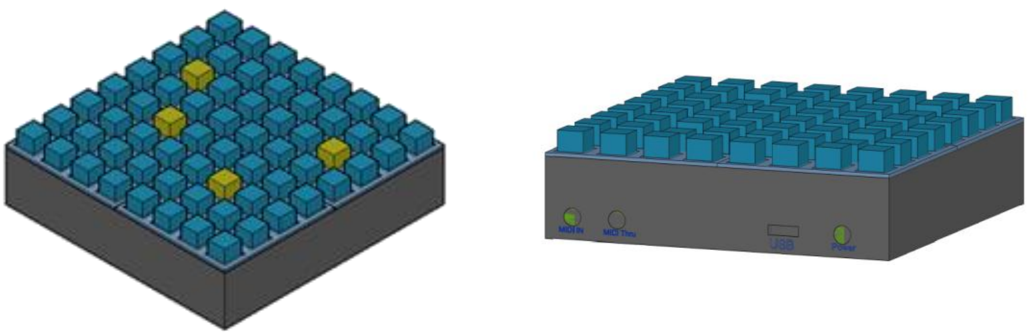

Figure 81: 3D Concept Drawing of 8x8 Expandome

Each Expandome enclosure will be made of clear acrylic with a wooden faceplate. Each enclosure will be custom made with the dimensions provided. The locations of the MIDI connectors were placed such that the devices may be connected and appear as one device. The enclosures include openings for the USB connection, MIDI connections and the SD card.

Figure 82: 4x4 prototype enclosure without faceplate

The face of the 4x4 Expandome enclosures are 104mm square. The face has 16 18mm x 18mm openings for the button pads. The back side will have an opening for a surface mounted USB female connection. And the side shall have two surface mounted MIDI 5-Pin DIN female connectors, as shown below. To keep the enclosure design uniform across all devices, the height of this enclosure will be 5 inches, the same as all other devices.

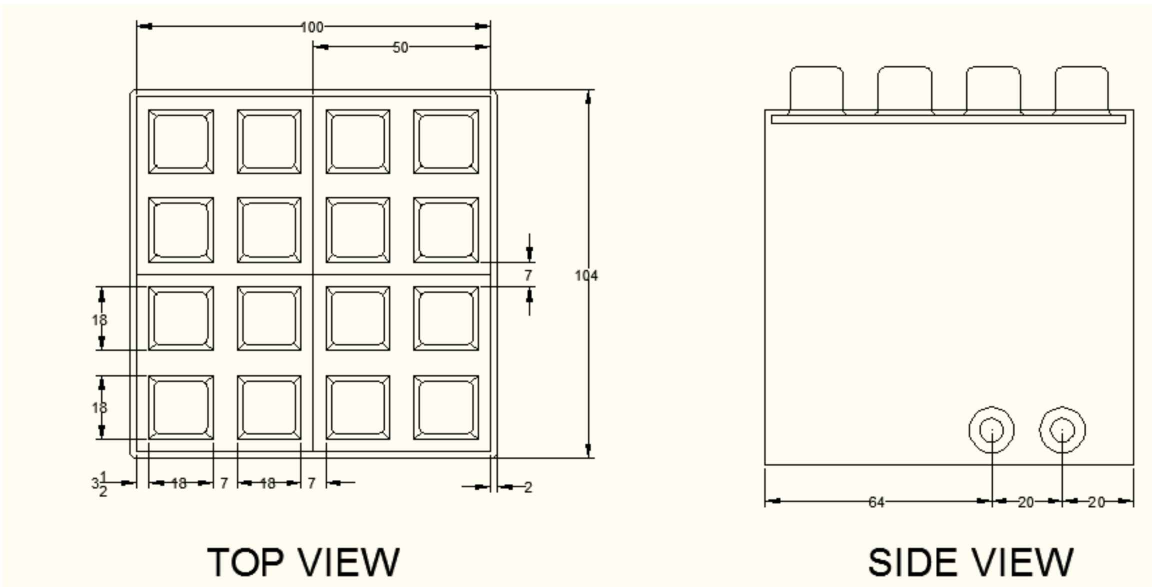

Figure 83: 4x4 Expandome Enclosure

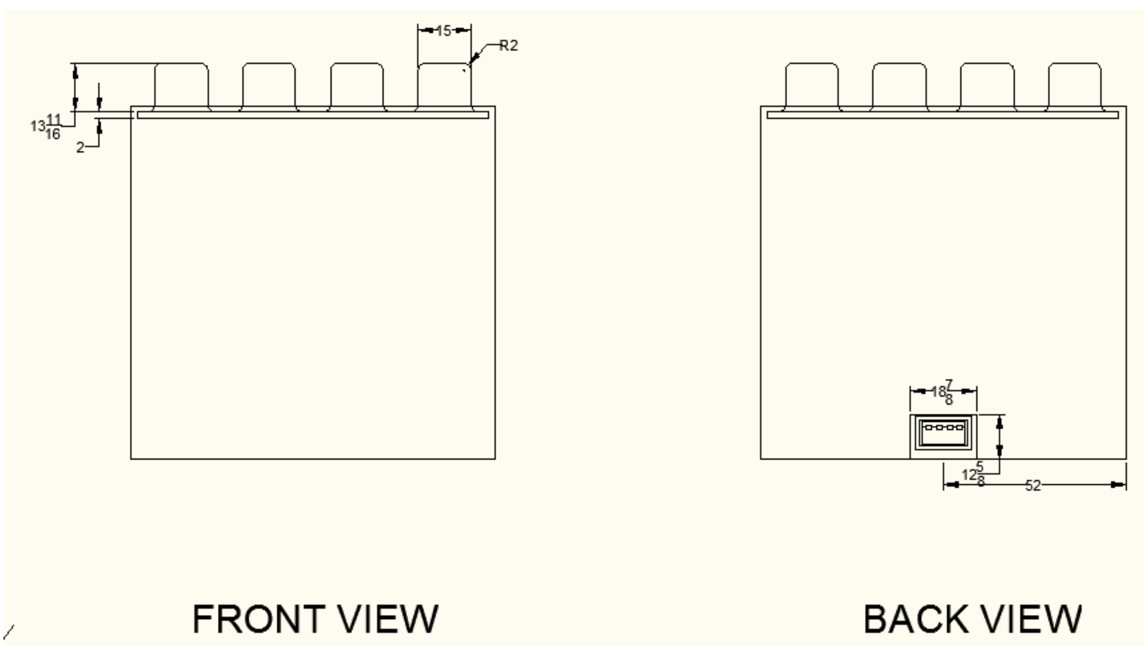

Figure 84: 4x4 Expandome Enclosure, side views

The face of the 4x8 Expandome enclosures is 104mm x 204mm. This device will have 32 precisely cut 18mm x 18mm openings for the button pads. This device will have the USB port and 4 MIDI connectors be a height 5 inches tall.

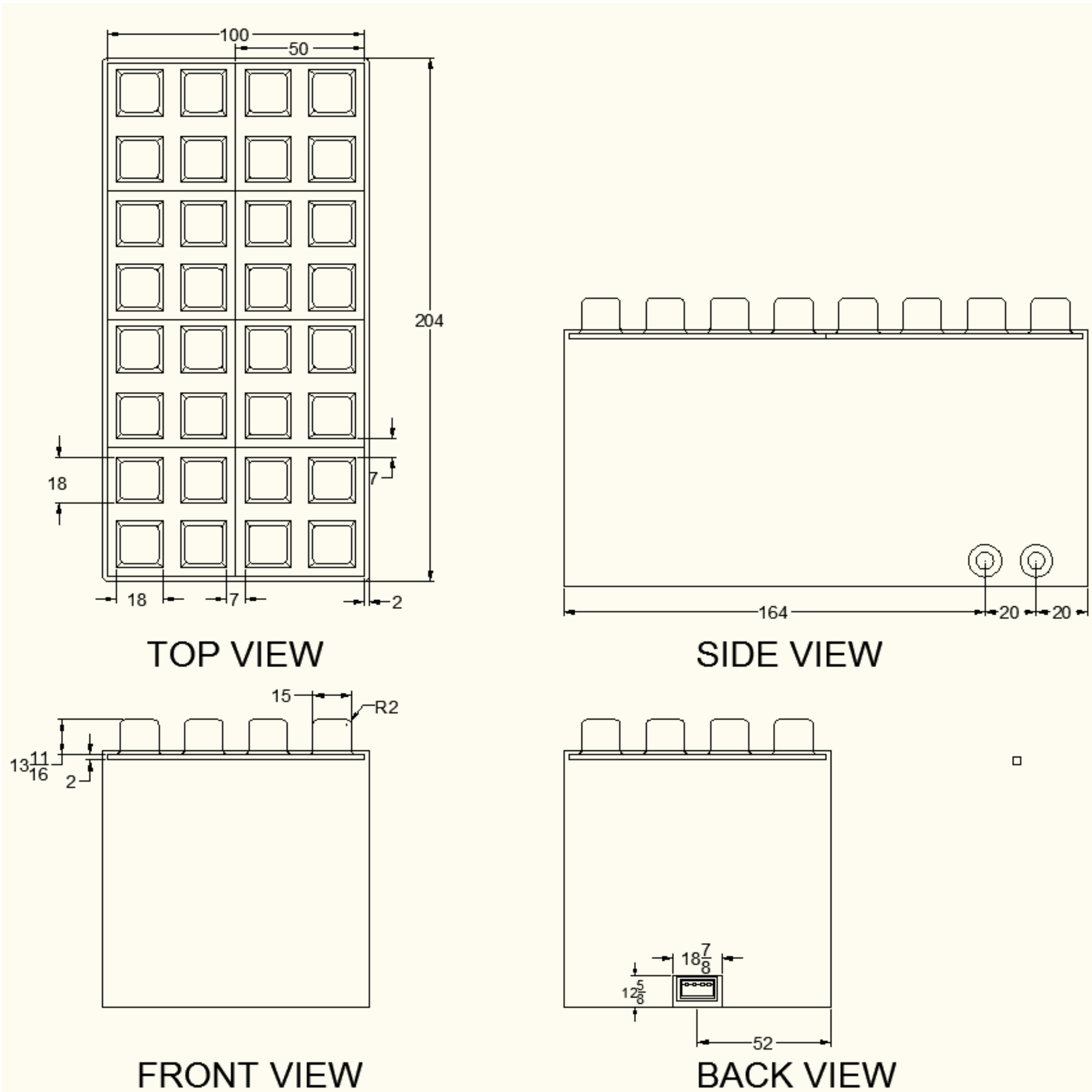

Figure 85: 4x8 Expandome Enclosure

The face of the 8x8 Expandome enclosures is 204mm x 204mm. This device will have 64 precisely cut 18mm x 18mm openings for the button pads. This device also has the USB port with a total height of 5 inches, the same as all other devices.

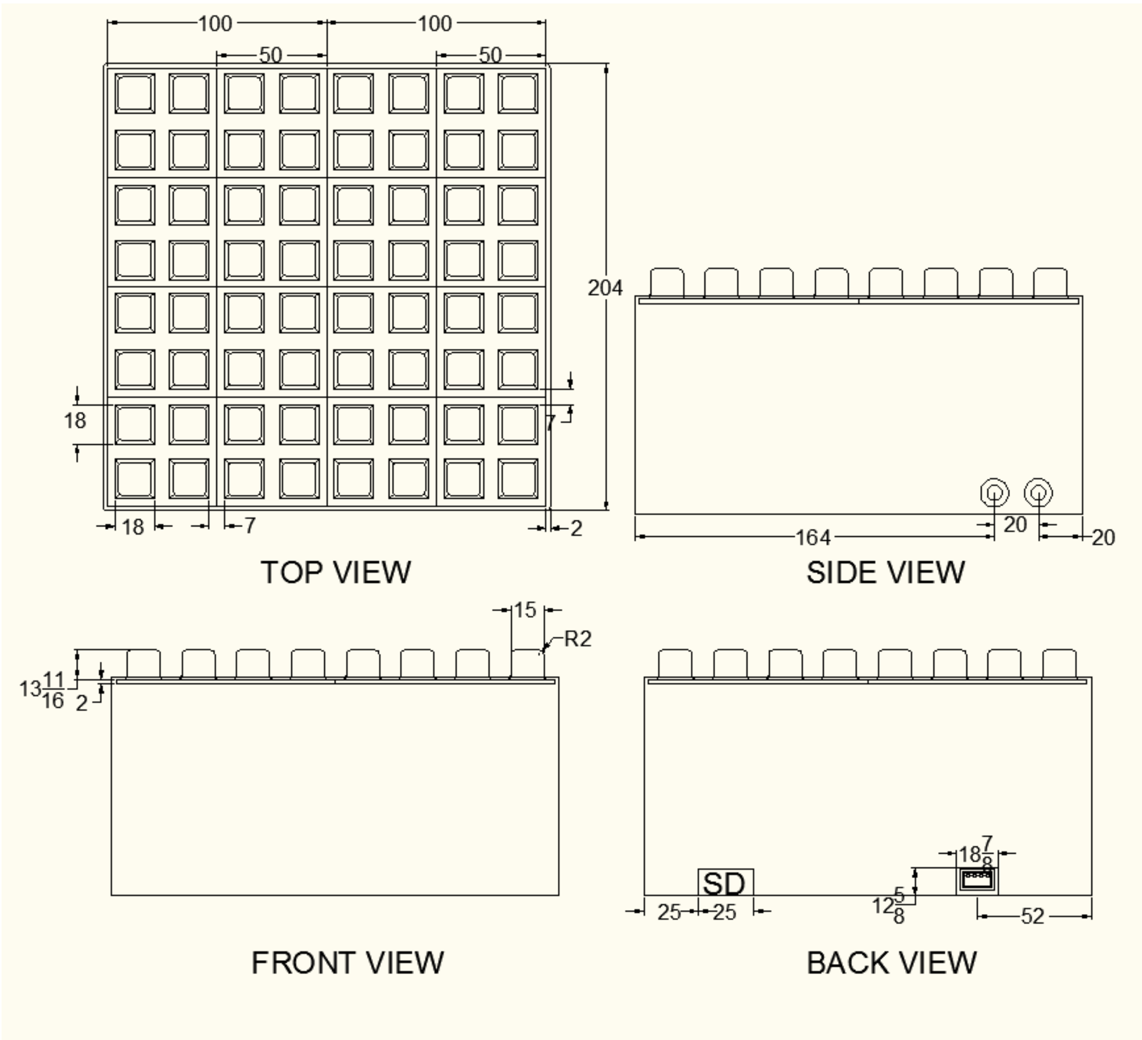

Figure 86: 8x8 Expandome Enclosure

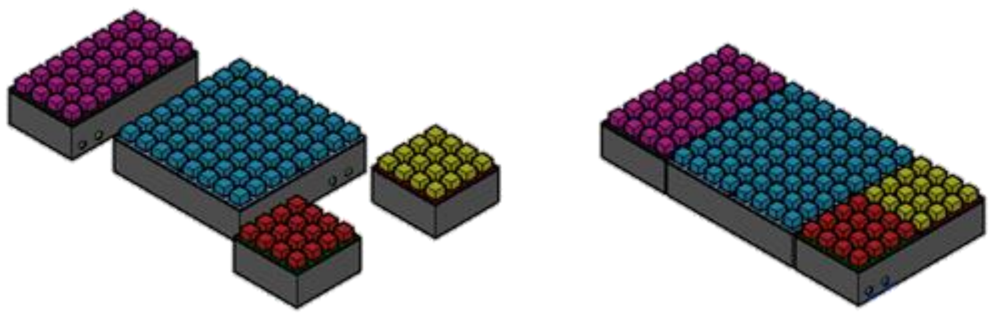

Figure 87: 3D Rendering of the Expandome Concept

The above figures illustrate the Expandome concept of adding devices of different sizes to the main 8x8 Expandome device.

SECTION 8.0 TEST

8.1 uWave Theremin

In this section, we will discuss the steps taken to test the uWave Theremin and validate that it meets requirements and is producing sufficient outputs needed by other devices in the system. Equipment needed for testing include, a multimeter, oscilloscope, low power speaker or an amplifier such as a guitar amp. Multimeter and oscilloscope are all that is necessary if testing input to pitch-to-MIDI converter.

8.1.1 Pre-Assembly Testing

The following test was carried out in room temperature and with very minimal devices in the room in order to reduce interference. In this section, we test for the response of the beat oscillator, volume oscillator and VCA individually in order to make sure that they are operating as expected. The following steps was done with the antenna circuits initially disconnected and all the circuits constructed on a breadboard.

Initial Testing

1. Upon applying power to the uWave Thermin, use a multimeter to check that the DC voltage at the collector of Q1 through Q8 is about 12V.
2. Check that the DC voltage at the emitter of the BJTs in the oscillator circuits (Q1 through Q6) are about -0.6V
3. Check that the DC voltage at the emitter of the BJTs in the tuning circuits (Q7 and Q8) are about -0.2V.
4. If the voltages mentioned above are not measured, check that the wires are connected properly and that there are no shorts or opens anywhere.

Variable Pitch and Fixed Pitch Oscillators Test

1. Measure with an oscilloscope, the AC signals across the variable inductors L4 and L6 seen in their respective oscillator circuits mentioned in the design section. If

the peak voltages are about 10 V then the oscillator is producing a waveform.

2. With the FPO connected to nothing but ±12V, connect an oscilloscope at the free end of C14 in order to measure the output response of the FPO. The output should be something similar to that seen in Figure 25. Check that the peak voltage is about 2.5 V. Turn the adjustable inductor in the circuit to verify that frequency changes as it is turned. Set the initial frequency of the oscillator to 285kHz using the adjustable inductor mentioned.

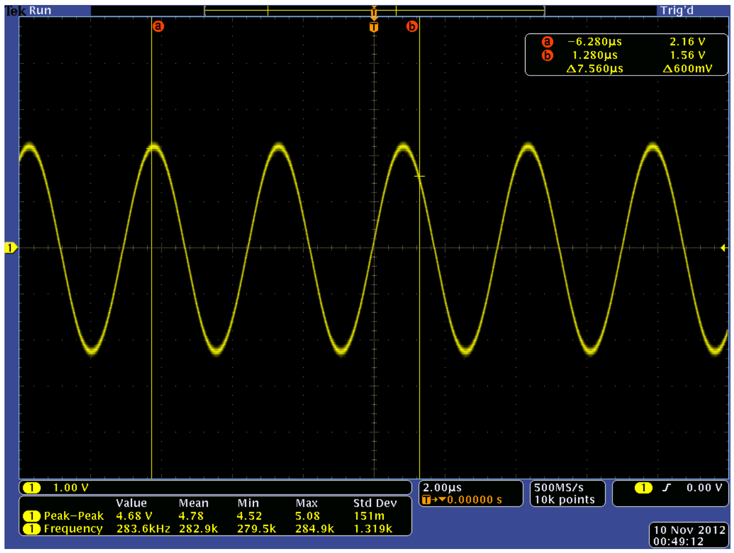

Figure 88: Output of the FPO

3. Connect the pitch tuning circuit to the FPO by making a connection between pins labeled "Pitch_tuning_in" and "Pitch_tuning_out" in Figure 24 and Figure 27.
4. With the pitch tuning circuit connected to the FPO, verify that it is operational by turning the potentiometer labeled R44 in Figure 26. With the oscilloscope still connected to C14 of the FPO, check that the frequency changes as the potentiometer is turned.
5. Repeat step 2 for the VPO circuit but this time, connect the oscilloscope to the free end of C10 rather than C14. The output should be something similar to that seen in the FPO.
6. With the VPO set to oscillate at 285 kHz, make a connection between pins labeled "Pitch_Ant_in" and "Pitch_Ant_out" seen in Figure 20 and Figure 27.
7. Now that the pitch antenna circuit is connected to the VPO, move your hand around the pitch antenna and check that the frequency changes as the hand moves about the pitch antenna.

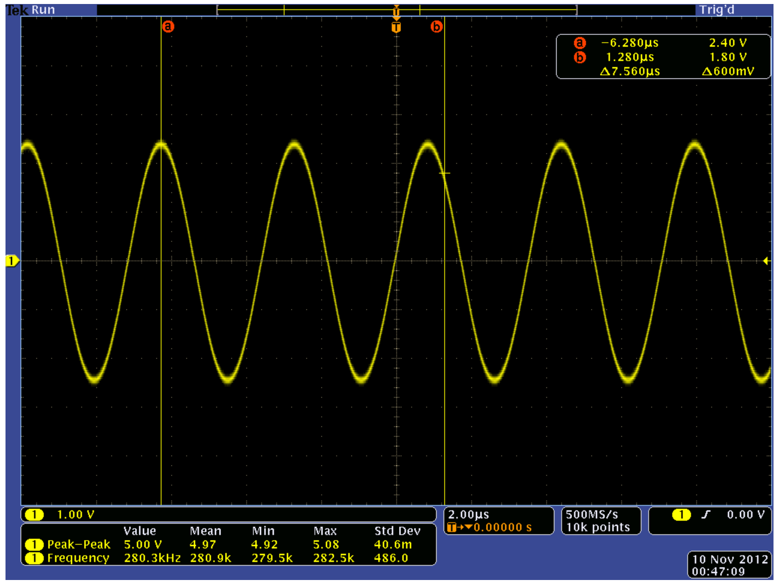

Figure 89: Output of VPO when hand is directly on pitch antenna

Beat Detector Test

1. With the pitch antenna disconnect from the VPO circuit, connect the ends of C10 and C14 to D1 (labeled "Pitch" in Figure 31) of the detector circuit.
2. Measure the DC voltage across R14 using a multimeter. Note that this is a RMS voltage. If it is around -0.5V then according to Moog, the detector is working properly[8].
3. Connect both probes of an oscilloscope and measure the response at D1 and R14. It should be similar to that seen in Figure 33.

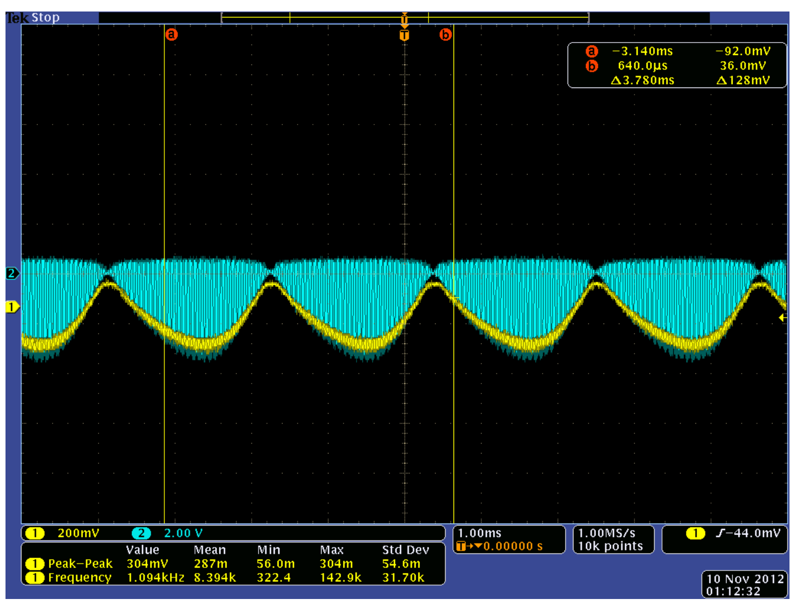

Figure 90: Output from the detector

4. According to Moog, temporarily connect a small powered speaker across R14. Turn the variable inductors, L4 & L6, counterclockwise until they can't be turned anymore.
5. Turn L4 two turns clockwise. Then turn L6 clockwise slowly until a high-pitched whistle can be heard. Turn L6 until the frequency response is about 1 kHz.
6. Now while turning the potentiometer R44, a change in pitch should be heard. If everything in steps 4 & 5 occur, then the oscillators and beat detectors are working properly.

Volume Oscillator Test

1. With the volume oscillator connected to nothing but ±12V, measure the AC signal across L13.
2. Turn the variable inductor L13 and connect an oscilloscope at the free end of R5. The output should be an oscillating signal similar to that seen in Figure 28. Check that the initial peak voltage is about 3V and that the frequency changes as the inductor is turned. Use this inductor to set the oscillator to oscillate at a frequency of 432 kHz.

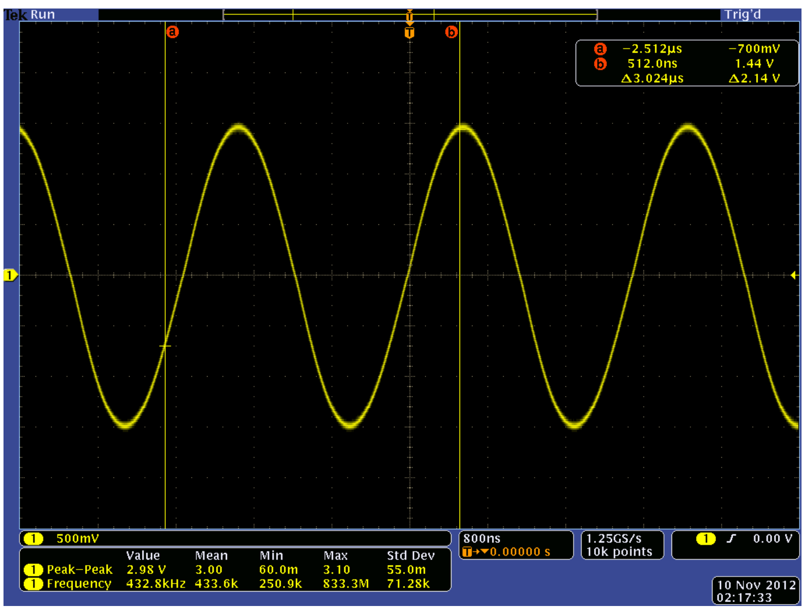

Figure 91: Output from volume oscillator

3. Connect the volume tuning circuit by connecting the pins labeled “Vol_Osc” and “To_Vol_tuning” seen in Figure 28 and Figure 30.
4. With the volume tuning circuit connected, turn the potentiometer R43 and verify that the output frequency seen at the end of R5 is varying with the turn of the potentiometer.
5. With the volume oscillator at 432 kHz, connect it to the volume tuning circuit by making a connection between the pins labeled “Vol_Ant” and “To_Vol_Osc” as seen in Figure 22 and Figure 28.
6. With the volume antenna circuit connected measure the DC voltage appearing at R2 as you put your hand by the volume antenna. Record the findings below.

Distance from antenna (in)	Voltage (V) at R2
20	-10.2
16	-9.6
12	-8.4
8	-4.4
4	-0.5

Table 16: Correlation between voltage at R2 and hand-antenna distance

VCA, Waveform and Brightness Test

The following test procedures are excerpts from Moog's DIY paper.

1. Temporarily connect pin 12 of U1 (seen in VCA processor section) to ground. The VCA should be turned on now.
2. Connect a small powered speaker across R25. You should hear a loud tone.
3. Connect an oscilloscope at R25. Use the pitch tuning control (R44) to set the tone's pitch to approximately middle C (261 Hz).
4. Turning the brightness control (R45) should change the sound from muted to bright.
5. Turning the waveform control (R46) should sound from a narrow waveform to a wider waveform.
6. Disconnect the temporary ground connection to pin 12 of U1 and connect that pin to -12V. There should now be no audio in this configuration. If there isn't the VCA is working properly. Return the connections to its original state.

8.1.2 Post-Assembly Testing

After testing in the previous section has been carried out and the uWave Theremin schematics have been built on a printed circuit board, another test was conducted to make sure that the uWave Theremin still behaves similar to the findings in the previous section and that it coincides with the design requirement and specifications. First we conduct a test on the sensitivity of the uWave Theremin as a players hand is brought closer and closer towards the antennas. We use the information gathered in Table 17 to verify this. Then a test on the control knobs (volume tuning, pitch tuning, waveform, and brightness) was carried out to make sure the effects they have on the audio output are as expected. After the PCB have been checked and verified that it still behaves as seen in the pre-assembly test, then wemount the PCB and antennas into the enclosure and will tune the device for performance. This is the subject of the next section.

8.1.2.1 Tuning

Before the uWave Theremin can be ready for use, it had be tuned for operation. In Robert Moog's DIY paper he gives an in depth step by step procedure on how to tune the device. In his procedure he has constructed the theremin on perfboards but the steps he goes over will be applied to our device constructed on a PCB. The following procedures, retrieved from Moog’s paper, were performed in a clear space in order to reduce conductivity from devices like lamps or oscilloscopes.

Pitch Tuning

1. On the main board, ground pin 12 of U1 by connecting a white wire to that pin and ground temporarily. This turns off control of the volume section. A guitar amplifier can be connected to the output of the device to make the audio audible.
2. Set Pitch Tuning control (R44) to its middle position or 50% of the potentiometer.
3. The pitch antenna will be held with one hand and the FPO variable inductor (L6) will be adjusted until the beat frequency is zero – that is no audio should be heard. Then turn L6 until a pitch of about 3 kHz can be heard. If the note cannot be distinguished, a simple frequency detector device like a guitar tuning software in smart phones can be used.
4. Slowly retract the hand from the vicinity of the pitch antenna. This will cause the pitch to go down (lowers frequency).
5. If the pitch does not go down to zero when the hand is completely away from the pitch antenna, this means the inductance of L4 is set too high. Adjust L4 by turning it slightly clockwise and then repeat steps 3 and 4.
6. If the pitch goes to zero and then gets higher as you retract the hand from the pitch antenna, the inductance of L4 is set too low and should be adjusted slightly counterclockwise. Then repeat steps 3 and 4.
7. Repeat steps 3 and 4 until the frequency is zero when the hand is away from the pitch antenna, and begins to increase when the hand is getting closer and closer to it and reaches about 3 kHz when the hand touches it. This concludes pitch tuning for the uWave Theremin. This is a similar process when tuning the device with pitch tuning knobs during performance.
8. When the criterion in step 7 has been reached, multiple frequency measurements will be taken relative to hand-to-antenna distance and recorded in the table below. These measurements will be used as a reference when testing the accuracy of the SenseBox.

Distance from antenna (in)	Frequency (Hz)
11	27.5
10.4	55
8.4	110
7.5	220
5.5	440
2.6	880
1	1760
0	3000

Table 17: Output frequency measurements at multiple octaves of note ‘A’ during interaction between hand and antenna.

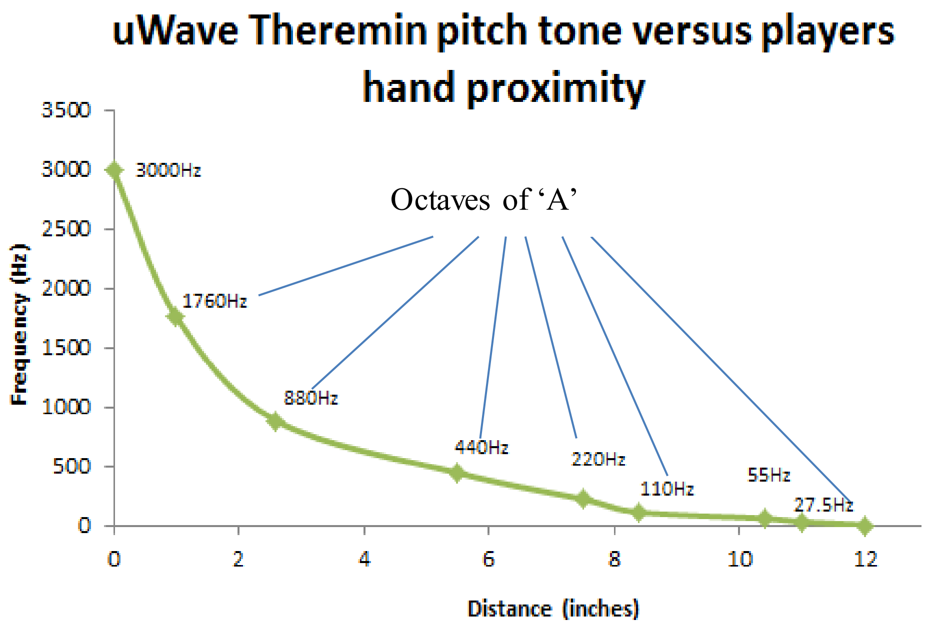

Figure 92: Graph showing response of the uWave pitch antenna at different octaves of 'A'

After the previous steps have been carried out, the volume section was tuned. First we will remove the temporarily grounded pin 12 of U1 and connect a voltmeter in order to measure voltage with respect to ground.

Volume Tuning

1. Set the volume tuning control (R43) to its middle position or 50% of the potentiometer value.
2. Adjust the volume oscillator variable inductor (L13) counterclockwise until it is at its beginning position. Check that the voltage on the voltmeter is about -12 V.

Distance from antenna (in)	Voltage at VCA Processor pin 12 (V)
18	10
10	8
9.5	5
8	1
7.75	-1
6	-5
5	-8
4.75	-10
4	-11

Table 18: Voltage readings at pin 12 of the VCA at different hand distance from the volume antenna

3. Turn L13 clockwise until the voltage at pin 12 of U1 starts to rise from -11 V and stop when it passes 0 V and becomes positive.
4. When the hand is near the volume antenna, the voltage at pin 12 should start to reduce. It should be about -12 when the hand is 2 or 3 inches away from the volume antenna. Record the voltage readings as the hand is moved closer and closer to the volume antenna.

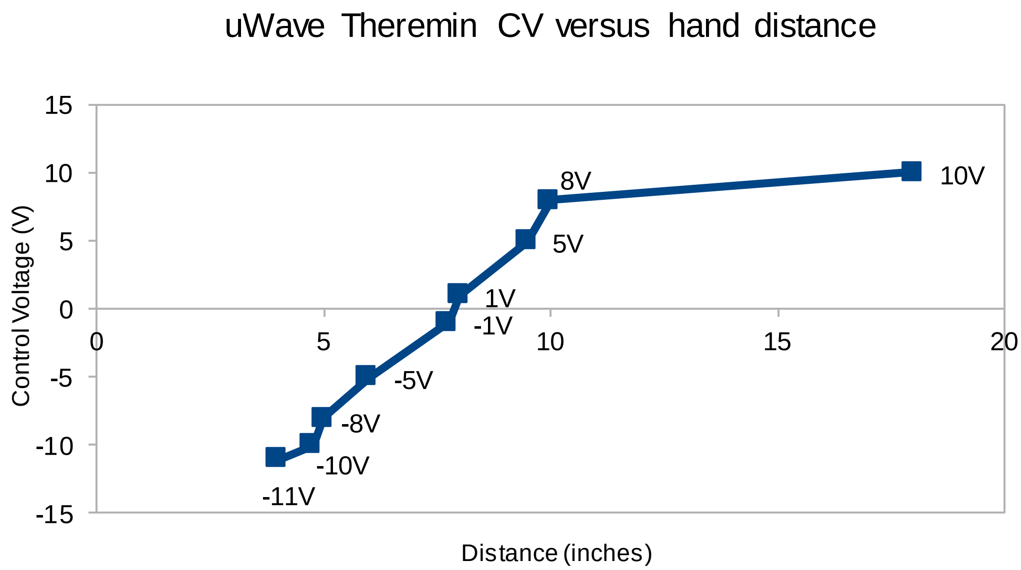

Figure 93: Volume antenna response vs hand presence

The above process was done when tuning the volume antenna during performance only this time, the volume tuning knob will be adjusted rather than the enclosed variable inductor.

8.2 SenseBox

In this section, we will discuss the steps taken to test the SenseBox and confirm that it meets requirements and receives/sends MIDI data. Equipment needed for testing include a multi-meter, oscilloscope, a MIDI keyboard, DAW software.

8.2.1 Pre-Assembly Testing

Since we will be testing the SenseBox using the uWave Theremin, the following test procedure will be carried out in a room temperate with very minimal devices in the room in order to reduce interference. In this section, we will test the audio input system, which includes biasing network, envelope follower, differential amplifier, buffer & low pass filter and threshold comparator - these components will be tested one at a time. The source of the input will be from the pitch preview and audio outs of the uWave Theremin. The software aspects will be covered in the Post-Assembly section.

Initial Testing

1. Before testing the SenseBox, check that the uWave Theremin's output has the range frequency from 0 to 3 kHz at 0.8 VRMS in the audio output.
2. Check the voltage supply is providing 5V. Use a multimeter to check it at R7.
3. If the uWave Theremin's frequency and audio output are met, as well as the voltage supply, check that the wires are connected properly and that there are no shorts or opens anywhere.

Biasing Network

1. Using with an oscilloscope, measure the DC voltage across R8 - this will be the offset voltage for the incoming AC signal. If the voltage is 2 V, the configuration setup is correct.
2. Check at AC signal across R8 and C5. The oscilloscope should measure an AC signal with an offset voltage of 2 from the original AC signal. The following screenshot should resemble in your oscilloscope

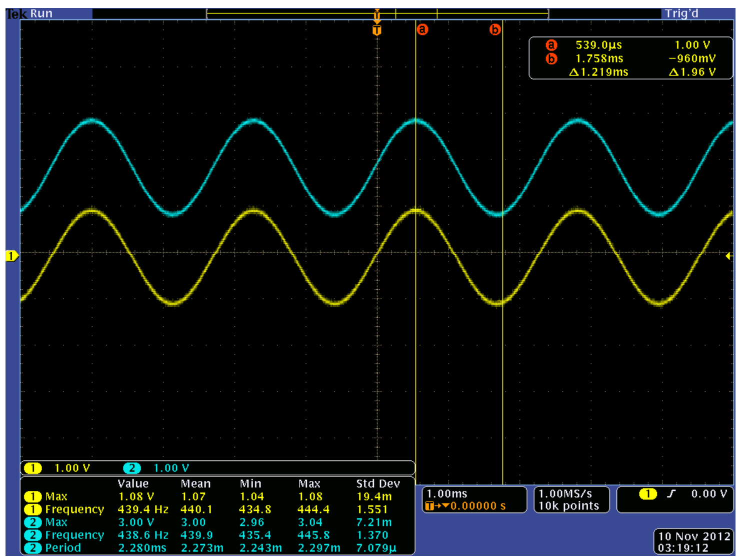

Figure 94: Blue waveform indicates out the biased audio signal

If met, connect the output to the envelope follower's input.

Envelope Follower

Using an oscilloscope, have two channels available to compare the input and output of the envelope follower. Set channel 1 to the input, which corresponds to the original AC signal that had an offset voltage of 2V, and set channel 2 to the output, which corresponds to the DC signal that envelopes the AC signal into half-wave rectification. If the oscilloscope shows this behavior, proceed to the next section. Otherwise, check your configuration.

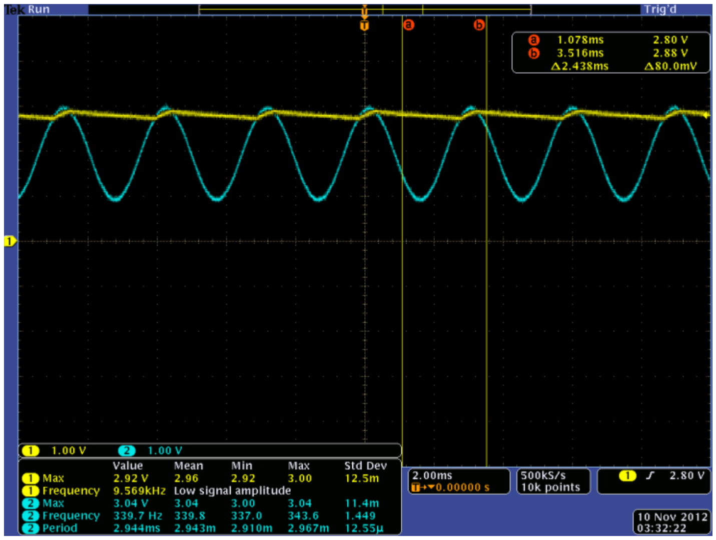

Figure 95: Output of the envelop detector

Differential Amplifier

1. Connect the appropriate pins for LM324N as shown Figure 42. The non-inverting pin should be connected from the the voltage reference we designed and the inverting pin should be connected from the output of the envelope follower.
2. Although in theory if using an oscilloscope and probing at the input and output terminals of LM324N, it should resemble Figure 43. However, that may not be the case. To confirm that it is working correctly, refer the following section.

Figure 96: Filtered output from the differential amplifier

Buffer and Low Pass Filter

1. Connect the appropriate pins for the buffer as shown in Figure 44.
2. Using an oscilloscope, connect across the end of C9. The output of the buffer should resemble the output of the Schmitt Trigger. The DC voltage is routed to

the analog pin of ATmega328p, which is how the volume data is obtained for the SenseBox. The following screenshot demonstrates a polished DC voltage signal

Threshold Comparator

1. Connect the pitch preview output of the uWave Theremin into appropriate pins as shown in Figure 47. Using an oscilloscope, have two channels available to compare the input and output of the threshold comparator. If done correctly, the output should be a square wave signal, which will represents the binary data that is sent into the digital pin of the ATmega328p—this will be the pitch data for the SenseBox.

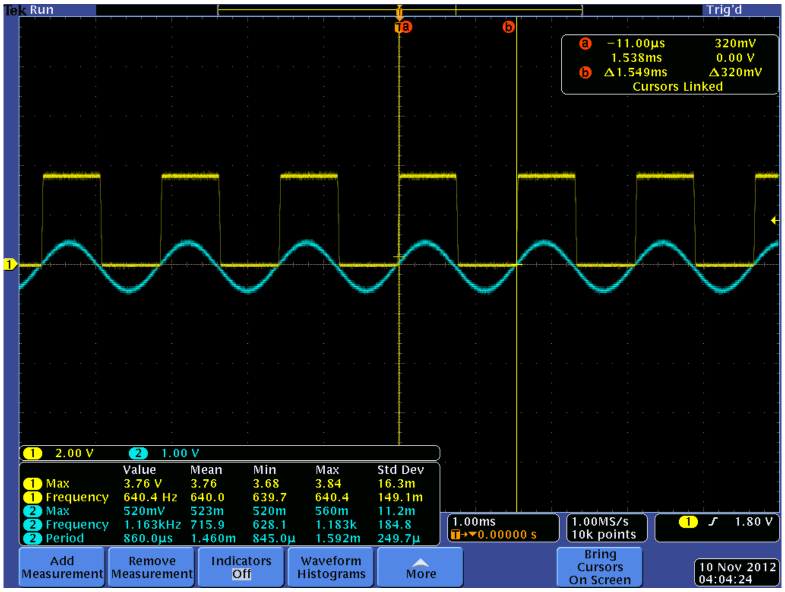

Figure 97: Output of the threshold comparator

8.2.2 Post-Assembly Testing

After testing the previous section, another test will be conducted to confirm that the SenseBox behaves and coincides with the design requirement and specification. Using the results in Table 17, we will test to see if the converter is receiving the analog signal from the uWave Theremin and accurately converting it into MIDI data. Next we will confirm if the four modes operates correctly. We will use Reaper as our DAW and a MIDI keyboard to verify that the SenseBox is working properly.

8.2.2.1 Analog to MIDI

Pitch Accuracy

1. Connect both the audio and pitch preview outputs of the uWave Theremin to the SenseBox's appropriate pins.
2. Using a MIDI cable, connect the SenseBox to a computer that has a DAW (in this case Reaper will be our DAW). Select device MIDI input as our SenseBox in the MIDI sequencer. Choose channel 1 as our main input.

As you put your hands near the pitch antenna, the LCD screen panel should denote the correspondent MIDI note number. The closer your hand is to the pitch antenna, the higher frequency it will emit. To confirm if it is denoting the right MIDI note number, use the Table 3 to validate its performance. Place your hand to one of the positions listed in the table and using the frequency that was recorded, verify if the MIDI note number displayed in the LCD corresponds to the value of the equation below.

Equation 5: $$\boldsymbol{p = 69 + 12 \times \log_2 \frac{f}{440Hz}}$$

Note that when placing your hand near the pitch antenna, it may produce certain frequencies that do not match the MIDI note number. Using the LED's mounted in the PCB board, make sure you position your hand where the green LED will turn on in order to capture the desired frequency. If done correctly, the frequency should match the MIDI note number.

Volume Accuracy

1. Repeat steps 1 and 2 from Pitch accuracy section
2. Position your hand near the volume antenna. As you draw further away from the antenna, the volume should increase. Monitor the LED's—as you do so, the LED's should light up sequentially.

8.2.2.2 Mode Operation

Pitch Mode

1. Have the rotary switch position where the LCD indicates PITCH mode. This will enable the SenseBox to PITCH mode.
2. Open the MIDI editor in the Reaper program. As you play the theremin, you should see that the MIDI editor's input is the SenseBox.
3. If you hit record, the DAW should be able to capture the MIDI data. If so, the SenseBox works properly in PITCH mode.

Control Mode

1. To enable CONTROL mode, rotate the rotary switch until the LCD indicates CNTRL mode.
2. Open the Reaper program. Select a parameter that you wish to control. In the preference, simply connect port#35 to the desired parameter.
3. As you hover your hand near the pitch antenna, the parameter should change as so: the closer your hand is to the pitch antenna, the more magnitude you are emitting. If acting as so, the SenseBox works properly in CONTROL mode.

ARPEGGIO1 Mode

1. Rotate the rotary switch until the LCD indicates ARP1 mode.
2. Connect a MIDI keyboard to the SenseBox MIDI in.
3. Open the MIDI editor in the Reaper program.

4. Choose a root note by pressing a key on the keyboard. The LCD screen should display the corresponding note key.
5. Refer to Table 11 for choosing the type of chord. When choosing the chord type, the LCD screen should display the chord type.
6. While holding the keyboard, hover your hand near the pitch antenna. The SenseBox should output the notes of the chord type sequentially.

ARPEGGIO2 Mode

1. Rotate the rotary switch until the LCD indicates ARP2 mode.
2. Connect a MIDI keyboard to the SenseBox MIDI in.
3. Open the MIDI editor in the Reaper program.
4. Press any keys in the keyboard that you wish to apreggiate.
5. While holding the keyboard, hover your hand near the pitch antenna. The SenseBox should output the notes you want arpeggiate sequentially.

8.3 Expandome

8.3.1 Confirming Specifications Met

To ensure that every LED properly lights and that the flicker between them is quick enough connect the Expandome to a computer. The backlit grid has the capability of having the appearance of being fully lit due to rapidly multiplexing each row. In turn push each button and verify that each button is lit on the screen. Ensure that each remains lit and that each row has a uniform appearance.

The device should have a maximum responsiveness towards multiple button presses. Clear all buttons and then take a hand and place each finger on a button. You should have 5 buttons selected, but not pushed. Now, push all of them at the same time. Alternately, you can take a ruler or other flat device to push multiple buttons at the same time. This has a more uniform effect. After pushing all at the same time, remove hand or ruler and verify that all buttons pushed are on. The device shifts the databits along very very fast and all should have been successfully lit. If so, then the device has maximum responsiveness towards multiple button presses.

Repeat this action, but ensure that each button has an assigned noise. Take a sound file and copy it 5 or so times. Place these files on the SD card and load them onto the Expandome. Repeat the 'maximum responsiveness' test. Due to the sounds all being the same, you should be able to hear if there are any abnormal responses. If this sounds normal then you have confirmed that there will be minimum delay in noise responsiveness, between pushing a button and immediate data and noise reply.

To ensure that the device is fully capable in standalone mode, remove the device from all other devices other than a computer. Default programs on microprocessor will allow the Expandome to perform without any virtual instrument programs. This should auto-load on the host computer, but if not you can navigate to the device in the device manager and direct it to run. Go through all the programs and verify that they work. Some program-specific tests are listed below.

The device should be capable of general device recognition and interacting, via MIDI and USB, with any typical computer virtual instrument program. Connect the device to a computer and select a standard virtual instrument program. Verify that the device is recognizable, and after so, each button on the device should be assignable as any keyboard or other MIDI device on the market.

The device should draw 5V max and have a maximum cascading current draw of 500mA for up to three devices. Take three devices and connect them flush/cascade. Connect an 8x8 gridded device to a computer, with tie-ins to the USB pins exposed inside the enclosure. Using a digital multimeter verify the current and voltage in and out of each instrument.

The smaller gridded devices are able to run on battery power. Remove all connections to the devices and insert full batteries into the battery packs. Turn the instruments on. Current draw for each row is set with the *Rset* resistor. Two resistances will be chosen, one for maximum brightness, and a second for lower power consumption by minimalizing the current draw. Because of this, when running on battery power, the LEDs will be dimmer, this is on purpose. A false-positive case however is when the batteries are low. This can be prevented by using wall-adapter battery packs available at most electronic surplus stores.

Connect a an 8x8 gridded device to a computer. Next ensure that the battery pack in a smaller device is fully charged. Connect the smaller device to the larger device flushly. Perform a mild set of beats and save them. Unceremoniously, unplug the smaller device from the larger, wait a moment, and replug back in. If the tracks continue as previously played then the device has been confirmed to run on batteries when unplugged from a power source and no interruptions shall occur, if devices are working in their standalone capacity. This shall aid in power cascade.

<u>TEST DEFAULT PROGRAM FUNCTIONS:</u>

Program Load

1. Connect device to the computer using the USB port.
2. Open Arduino Environment
3. Choose Tools > Serial Port > Select the Com Port for the connected device.
4. Choose Tools > Board > Select the item which best describes your device
5. File > Open > Select desired program to be loaded to the device.
6. Press the Verify button to scan the program for errors.
7. Choose the Upload button to load the program to the device.
8. When the program has been successfully loaded, all LED's shall blink twice (0.5 seconds on, 0.5 seconds off).
9. Verify that beginning from the top row, the device illuminates one row at a time, each row for 0.1 seconds, than illuminate the next row. Begin at row 1 thru *n*.
10. Verify that beginning from the left column of the device, the column will illuminate, one at a time, each row for 0.1 seconds, than illuminate the next column. Begin at column 1 thru *m*.

11. Repeat Row and Column LED sequence 5 times.
12. The Initialized sequence shall play for 5 seconds (Initialized Sequence: This sequence will flash one row at a time, than one column at a time.)

Beat Button:
1. Press a beat button.
2. Verify that the LED turns on while pressed
3. Verify that programmed beat is played

Beat Repeat:
1. Press Beat Repeat button.
2. Verify that the LED turns on.
3. Select and press a beat button.
4. Verify that the beat button LED stays ON while beat is repeating.
5. Verify that the desired beat is played and repeats.
6. Verify that the Beat Repeat button is illuminated.

Clear Beat Repeat:
1. Following the sequence to turn on Beat Repeat.
2. Select the Clear Beat Repeat button.
3. Verify that the Beat Repeat LED turns OFF.
4. Verify that the selected repeated beat LED turns OFF.
5. Verify that the repeated sound turns OFF.
6. The Clear Beat button only illuminates when pressed.

Clear All:
1. Select multiple beats to played.
2. Select the Clear All button to Stop and Clear all selected beats
3. Verify that the entire board illuminates while the button is pressed.
4. Verify that all selected beats stop.
5. Verify that all LEDs are no longer illuminated.

Record:
1. Select the Record button.
2. Verify that the button remains illuminated during recording session.
3. Choose or play any sequence or beats.
4. Select the illuminated Record button to STOP recording and clear all selected beats.
5. Verify that all LEDs and beats are cleared.
6. Verify that the Record button is no longer illuminated.

Play Back:
1. Following the Record Sequence
2. Select the Play Back button
3. Verify that the button stays illuminated while recorded sequence is playing.
4. Verify that the recorded sequence plays back as desired.

5. Verify that the button is not illuminated when the recorded sequence is completed.

User Interface Software Test

TEST EXPANDOME BROWSER AND VIRTUAL GRID

1. Connect Expandome device to the computer with USB cord
2. Verify that the Expandome User Interface appears on the desktop
3. Verify that the interface automatically detect and displays the correct grid dimensions
4. With the mouse, select a button in the interface
5. Verify that on the connected device, the associated button lights up and sounds
6. Select a button on the device
7. Verify that on the interface, the associated button lights up and sounds

SD Card Read/Write Test Button/Row Assignment

1. Connect the Expandome device
2. Verify that the Expandome User Interface appears on the desktop
3. Verify that the interface automatically detect and displays the correct grid dimensions
4. On the interface, select the Assign Mode options
5. On the interface, select the desired button to be programmed
6. Select the desired beat to be assigned
7. Repeat this step until all buttons have been selected
8. Choose Save Beats option
9. Press a button on the device
10. Verify that the button illuminates while depressed
11. Verify that the programmed beat is played

Test file parse numeration

1. Have the beat sound files already located on a host computer
2. Remove SD card from Expandome
3. Insert SD card into computer
4. Navigate to the "/BEATS" folder
5. Insert some sound files into the "/BEATS" folder and into the rest into folders named "ROWX", where X is any number from 0 to 7.
6. Remove SD card from computer
7. Insert SD card back into the Expandome
8. Connect Expandome to a computer
9. Turn Expandome on
10. The Expandome Browser will appear on the screen, navigate to the *Browse Beats* program.
11. Verify if all files inserted into the "/BEATS" folder are in this list
12. Navigate to the *Virtual Expandome* program
13. Right-click on each virtual button in succession and view the assigned beat for each.

14. Verify that each row has beat assignments that correspond to the beats placed in the respective row folders.

TEST SENSEBOX MODE SELECTION

1. Connect the Expandome to a computer
2. Connect the SenseBox to the computer
3. After the Expandome Browser program opens, navigate to the SenseBox Mode Selection program
4. Select a mode on the SenseBox and verify that the same mode is displayed on the virtual program
5. Repeat the above steps, but with the SenseBox connected via MIDI to the Expandome instead of the computer

TEST SENSEBOX DISPLAY CONSOLE

1. Connect the Expandome to a computer
2. Connect the SenseBox to the computer
3. Connet the uWave theremin to the SenseBox
4. After the Expandome Browser program opens, navigate to the SenseBox Mode Selection program
5. Verify that MIDI data messages are updated in real-time on the display console.
6. Repeat the above steps, but with the SenseBox connected via MIDI to the Expandome instead of the computer

SECTION 9.0 ADMINISTRATION

The following section covers the project schedule milestones, budget and financing, summarizing remarks and final thoughts on future work.

9.1 Milestones

	WEEK 1	WEEK 2	WEEK 3	WEEK 4
MAY			Form Design Team and Research Project Ideas	Submit Project Proposal and obtain approval.
JUNE	Divide and assign responsibilities for research report sections and format. *(Goal: Write a minimum of 6 pages per week each)*	Discuss and define hardware design and power requirements, define methods connection and signal processing (Report: 6x4 =24pgs)	Discuss and define software requirements, data structure and implementation. Order design parts and build prototype modules. (Report: 12x4 =48pgs)	Define material, component, and equipment listing and sources. Build and test circuits for research of signal processing. (Report: 18x4 =72pgs)
JULY	Document simulation output, photograph prototypes, flow charts signal/data. Component diagrams specs. (Report: 24x4 =96pgs)	Test software on research prototype. Program and test function. (Report: 30x4 =120pgs)	All members submit reporting sections. Draft report for review.	Finalize report details, corrections, and format.
AUGUST	Submit Final Senior Design I Paper (Report: 30x4=120 pages)	Order/acquire materials for final build (PCB, enclosure, power supply, controller, audio out speaker)	Order/acquire materials for final build. Build box/enclosure. Review DSII Report requirements and format, assign members to sections.	Finalize simulations, research prototype models. Begin building components per design.
SEPTEMBER	Build Theremin and test.	Build Pitch to MIDI Interface, connect to Theremin. Test output.	Build Arduino Board, connect to PC and MIDI Tracker, program and test.	Connect all components for initial test. Review DSII Report Status and format.
OCTOBER	Program and test. Cut material for final model enclosure.	Build enclosure for final prototype model. Program and test.	Complete construction of all components and enclosure.	Program and test. Work on report.
NOVEMBER	Program and test. Work on report	Program and test. Work on report.	All members submit reporting sections. Draft report for review.	
DECEMBER	Project is 100% complete. Presentation.	Graduate!!!		

9.2 Budget and Finance

Below are the costs absorbed and proposed for this project. At the time of writing, this project is fully self-financed. The final budget will be divided evenly amongst the group members. Respective costs have been estimated using unit prices by the most common manufacturer. Total costs given below have been rounded up to account for deviations in materials or planning.

COST ESTIMATE

Project Title: uWave/SenseBox/Expandome Date: 7/29/2012

Group: GROUP 1

ITEM	DESCRIPTION	QTY	UNIT	UNIT COST	LINE TOTAL
1	uWave Theremin	1	each	$ 150.00	$ 150.00
2	Expandome 8x8	1	each	$ 200.00	$ 200.00
3	Expandome 4x8	1	each	$ 150.00	$ 150.00
4	Expandome 4x4	1	each	$ 100.00	$ 100.00
5	MIDI Interface	1	each	$ 100.00	$ 100.00
6	Power Supply	1	each	$ 50.00	$ 50.00
7	Software (Open)	1	each	$ 50.00	$ 50.00
8	Enclosure	5	each	$ 100.00	$ 500.00
	TOTAL ESTIMATE				**$ 1,300.00**

9.3 Donors and Funding

We do not have any project sponsors and this project has been fully self-financed. There are still many features and devices we wish to develop further and we wish to continue pursuing the design of musical interfaces and computer music research. Towards this, we will be releasing a Kickstarter in the near future.

SECTION 10.0 PROJECT SUMMARY

The initial goal of making a highly adaptable set of musical instruments has been pursued with the uWave theremin, SenseBox, and the Expandome easy-grid OSC controller. These devices were developed with a design mentality to provide a virtuosic experience with ease and to complement a musician's mental organization.

Three initial designs were examined and molded to work in concert: the theremin invented by Léon Theremin, the Pitch to MIDI Tracker by Stephen Hobley, and the Monome and Arduinome developed by the Monome and Flipmu groups respectively. The final results are purposeful iterative designs that expand the capabilities of the originals in novel ways.

In summary, we submit our set of adaptable performance instruments to the community of design for further iterative development and hope that the philosophy of open collaboration engenders further work in new and innovative instruments and interfaces.

SECTION 11.0 CONCLUSION

This senior design project has helped the members of the team understand the process of an engineering outcome. With the amount of hours spent doing research and acquainting ourselves with the project scope, we have gained new knowledge about several electrical engineering topics such as high frequency applications, digital signal processing applications, soldering, programing etc.

Each device that was investigated and implemented brought unique findings about the world of Electrical Engineering. In the case of the uWave Theremin, knowledge was increased in regard to understanding how sensitive devices can be when operated with high frequencies. Understanding how much factors play in the stability of the device's antennas helped gain knowledge of this. Also, being able to work with oscillators that were used to produce musical notes played a part in the team's knowledge of analog devices.

Another source of knowledge that was gained during this project was learning more about analog-to-digital conversation. In our project, a MIDI device was created to convert analog signals from the uWave Theremin into a digital format known as MIDI. Learning more about MIDI protocols helped us gain more familiarity with music theory and MIDI applications. In addition it helped incorporate new understanding of MIDI with a microcontroller using the Arduino language. It was also learned that with the help of the SenseBox, which is a pitch-to-MIDI converter, the uWave Theremin can be used for many other applications other than music.

With the help of the Expandome, the player has the ablilty to interact with the uWave Theremin and SenseBox in a way that allows for customizability adequate enough for unique performances.

This project has enabled us to take an old but powerful technology which requires quite a few skills and dedication to operate, and made it a little easier and more intuitive for people with minimal musical understanding to enjoy. This was achieved by using the supporting devices, the SenseBox and the Expandomes to produce a feedback system the user can rely on during performance. With that being said, this senior design project sets a good example of what the purpose of engineering is, which takes a problem or a need and produces a solution.

SECTION 12.0 APPENDIX

12.1 Acronyms and Nomenclature

AF Audio Frequency
AM Amplitude Modulation
ASIO Audio Stream Input/Output
BJT Bi-junction Transistor
CV Control Voltage
DAW Digital Audio Workstation
DIN Deutsches Institut für Normung
DIP Dual In-line Package
FPO Fixed Pitch Oscillator
IC Integrated Circuit
LCD Liquid Crystal Display
LED Light Emitting Diode
MCD Milicandella
MTS MIDI Tuning Standard
OSC Open Sound Control
PIC Peripheral Interface Controller
RCA RF coaxial connector (historically derived from Radio Corporation of America)
RIF Radio Influence Field
RLC Resistor/ Inductor/Capacitor
SD Secure Digital
SMD Surface Mount Device
TRS Tip Ring Slave
TS Tip Slave
TTL Transistor-transistor Logic
uOSC micro-OSC
USB Universal Serial Bus
VCA Voltage Controlled Amplifier
VPO Variable Pitch Oscillator
VST Virtual Studio Technology
XLR X-Linked Receive

12.2 Permissions

Steve Hobley

Contact Steve

The best way to contact me is by email, using the form below.

Your Name (required)

Kenzo Mendoza

Your Email (required)

kenzo_mendoza@knights.ucf.edu

Subject

Pitch-to-MIDI Design--Permission

Your Message

Hi Steve

My name is Kenzo Mendoza and I am in an electrical engineering senior design class at University of Central Florida in Orlando, FL. I am currently working on a project that involves integrating a theremin with a Pitch-to-MIDI converter, similar to yours.

I would like to request permission from you if we can include your name and your design in our documentation. Our model of the converter will be based on yours, with some changes in our implementation, both hardware and software.

Please let me know if we would have permission to use this for a strictly academic and informational purposes in our project research documentation.

If you have any questions, please do not hesitate

Thank you

Kenzo

12.3License

This work is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License[56].

All source and schematics are available on GitHub[57].

12.4References

[1] **The Chronome: A Case Study in Designing New Continuously Expressive Musical Instruments** Vallis, O., Hochenbaum, J., Murphy, J., Kapur, A., Proceedings of the Australasian Computer Music Conference (ACMC). 2011. Auckland, New Zealand.

[2] Theremin, Léon S. “Method of apparatusfor the generation of sounds.” US Pattent US1661058. 28 Feb. 1928.

[3] See Monome <http://monome.org>

[4] See Flipmu <http://flipmu.com/work/arduinome>

[5] **A Shift Towards Iterative and Open-Source Design for Musical Interfaces** Kapur, A., Darling, M., Murphy, J., Hochenbaum, J., Diakopoulos, D. Proceedings of the 2011 International Conference on New Interfaces for Musical Expression. June 2010. Sydney, Australia.

[6] Mundell

[7] **Physics of the Theremin** Skeldon, K., Reid, L., McInally, V., Dougan, B., Fulton, C., Am. J. Phys. Vol. 66, No. 11, November 1998.

[8] **Build the EM Theremin** Moog, R., Electronic Musician, pg. 86-100, Feb., 1996

[9] Gaudi. Urs Gaudenz. 27 June, 2011. Creative Commons Attribution-NonCommercial-ShareAlike 2.5 Switzerland License. <http://www.gaudi.ch/OpenTheremin/>

[10] Thermin World. June 13, 2012. <http://www.thereminworld.com/Forums/T/28594/antennas-antenna-coils-for-em-theremin?Page=1>

[11] **Basic Electronics.** United States Bureau of Naval Personnel. USA: Courier Dover, pg. 338, 1980.

[12] **Understanding and Calibrating the Theremin Front-End** Mundell Fred. Fundamental Designs Ltd.

[13] See <http://arduino.cc/>

[14] **Practical Electronics for Inventors.** Scherz Paul, McGraw-Hill, November 14, 2006

[15] **Firewire Vs. USB Audio Interface.** Josefsson Jesse, eHow. Demand Media Inc, 2012. <http://www.ehow.com/about_5446149_firewire-vs-usb-audio-interface.html>

[16] Stack Exchange. July 5, 2012. Stack Exchange Inc, 2012. <http://avp.stackexchange.com/questions/25/what-is-latency-in-a-digital-recording-studio>

[17] See <http://midi.org>

[18] **Pitch Perfect**. Dingfelder F. Sadie, American Psychological Association, Februrary 2005, Vol. 36, No. 2, pg. 32. <http://www.apa.org/monitor/feb05/pitch.aspx>

[19] Tech-FAQ. July 13, 2012. The Tech-FAQ 2012. <http://www.tech-faq.com/lcd.html>

[20] Altadox Electronics Design & Manufacturing. Altadox Inc, 2008. <http://www.altadox.com/lcd/knowledge/lcd_display_types.htm>

[21] Display Future, July 14, 2012. Display Future LTD, 2012. <http://www.displayfuture.com/Display/Introduction.asp>

[22] **LiquideCrystal – “Hello World!”**. Arduino, June 6, 2012. <http://arduino.cc/en/Tutorial/LiquidCrystal>

[23] Webstore. Internation Electrotechnical Commission, August 2004. <http://webstore.iec.ch/preview/info_iec61076-2-103%7Bed1.0%7Db.pdf>

[24] Basic Car Audio Electronics. June 13, 2012. < http://www.bcae1.com/rcacable.htm >

[25] Sparkfun Electronics. May 29, 2012. < https://www.sparkfun.com/products/8033 >

[26] Sparkfun Electronics. May 29, 2012. < https://www.sparkfun.com/products/7835>

[27] 1N4148 Datasheet. NXP Semiconductors <http://www.nxp.com/documents/data_sheet/1N4148_1N4448.pdf>

[28] **Mini Monome: An RGB Monome based on the Arduino.** The Box. June 13, 2012. <http://www.thebox.myzen.co.uk/Hardware/Mini_Monome.html>

[29] Monome. June 1, 2012. < http://docs.monome.org/doku.php?id=tech:mk:led>
[30] **The MAX7219 and Max7221 Led drivers**. Arduino, June 6, 2012. <http://www.arduino.cc/playground/Main/MAX72XXHardware>
[31] MAX7219/MAX7221 datasheet. Maxim. < http://datasheets.maxim-ic.com/en/ds/MAX7219-MAX7221.pdf>
[32] Read, Paul; Meyer, Mark-Paul; Gamma Group (2000). *Restoration of motion picture film*. Conservation and Museology. Butterworth-Heinemann. pp. 24–26. ISBN 0-7506-2793-X.
[33] **Wire Library**. Arduino, June 6, 2012. < http://arduino.cc/en/Reference/Wire>
[34] Monome. June 1, 2012. <http://monome.q3f.org/wiki/ArduinomeSerial>
[35] See Arduino < http://www.arduino.cc/en/Main/Software>
[36] See CNMAT, Berkeley <http://cnmat.berkeley.edu/research/uosc>
[37] See Lua <http://www.lua.org>
[38] See VVVV <http://vvvv.org/>
[39] See Androidome <http://ewanhemingway.co.uk/programming/androidome>
[40] See TouchOSC <http://hexler.net/software/touchosc>
[41] See Control <http://charlie-roberts.com/Control/>
[42] Wright, M., Dannenberg, R., Pope, S., Rodet, X., Serra, X. and Wessel, D. Panel: Standards from the Computer Music Community *International Computer Music Conference*, International Computer Music Association, Miami, FL, 2004.
[43] Freed, A. and Schmeder, A. "Features and Future of Open Sound Control version 1.1 for NIME", CNMAT, Berkeley, CA, 2009.
[44] **Understanding, Customizing, And Hot-Rodding Your Etherwave Theremin.** Moog Music Inc. 2003.
[45] **Modern Device LCD117/ PH Anderson's Serial LCD Driver Commands.** <http://cdn.shopify.com/s/files/1/0038/9582/files/LCD117_Board_Command_Summary.pdf>
[46] **SoftwareSerial Library**. Arduino, June 6, 2012. < http://www.arduino.cc/en/Reference/SoftwareSerial>
[47] See UT63B3-41-URE3 datasheet by Ledtech, Jameco part#334749
[48] See LVX2643-1 datasheet by Ligitek, Jameco part # 333307
[49] See Arduinome firmware <http://sourceforge.net/projects/arduinome/>
[50] See Chronome firmware <http://sourceforge.net/projects/thechronome/>
[51] See Löve2d <https://love2d.org/>
[52] P. Brinkmann, P. Kirn, R. Lawler, C. McCormick, M.Roth, H.Steiner, *Embedding Pure Data with libpd*
[53] See mnmewvtbl app <http://docs.monome.org/doku.php?id=app:mnmewvtbl>
[54] See Open Sound Control <http://opensoundcontrol.org>
[55] See Instant Glass Cases <http://www.instantdisplaycases.com>
[56] See Creative Commons <http://creativecommons.org/licenses/by-sa/3.0/>
[57] See GitHub <https://github.com/antivapor>