Final Report

Sound Effect Devices for Musicians:

Synthesizer

Presented by:

Francisco Alegria, Blake Beyer, Tim Day, Eric Fischer, & Travis Gillham

Delivered to:

Dr. Degang Chen, & Dr. Randy Geiger

Table of Contents

List of figures

List of equations

List of tables

Executive Summary

Acknowledgement

The Sound Effect Devices for Musicians team would like to kindly thank both Dr. Chen and Dr. Geiger for advising us as well as supporting us with knowledge through the duration of this project.

Problem statement

In today's world musicians can be put in a difficult spot when trying to find new equipment that will help them excel in the music world while staying under a reasonable budget. Musical devices, such as synthesizers, can cost anywhere from $500 - $10,000, with this most musicians don't have the cash to afford upgrading their equipment to compete in the competitive industry of music.

Our goal was to create a synthesizer for musicians that is easy to use, modular, and affordable. Musicians thrive on having the latest and greatest devices to create music, so we plan on creating our synthesizer with our own unique spin on it by using tablet in the form of an iPad or a smart phone that will allow users to create there one layout of the user interface. This device we created not only looks like a state-of-the-art synthesizer but also sounds like one too as we gave the musicians ample freedom to make music their way with our device being modular.

Our Solution

The designed synthesizer has met our desired expectations. Since, it has all the major modules that appear to be present in industry standards, uses affordable components, and the interface is user-friendly. The user interface is an app that can be downloaded on many pieces of hardware available to musicians. It is also simple to use, and the labels make it clear what will be adjusted in the hardware of the system. The software fully integrates with the hardware. This wireless connection is what makes our solution unique to previous synthesizer designs.

The housing for the synthesizer keeps all the devices secured as well as provides simple access to the power supply and system output. It also makes the wire management manageable since all of the modules are on different boards.

Intended user(s) and intended use(s)

Intended users of the synthesizer include everyone that has an interest in making music or making noise for entertainment. All musicians, from amateurs to professionals, are the target audience for this product. We aim at making this not only a highly sophisticated tool, but also one that is easy and fun to use.

The intended use is up to the users. Our hope is this tool is being used to help professionals create music but also people that are just having fun. This product in the general realm is intended for musicians who want to make high-quality music on a lower end budget.

Assumptions and limitations

Assumptions:

- Users can read and understand English - to understand the brief introduction

- Users will have access to a wall outlet
- All components will work inside the product
- Product will be taken care of by owners

Limitations:

- People without musical knowledge will have a learning curve
- Group members have conflicting schedules so finding an ample amount of time each week to work is difficult

Expected end-product and other deliverables

The expected final deliverable for this project will aim at handing over a sophisticated synthesizer that is not only easy to use but is also a very powerful tool for musicians to have. The synthesizer will cost an affordable amount, but that will not affect the functionality of the synthesizer. The tool will be fully functional and include eight modules that give musicians the capability to make a wide range of music. With this project we have specifications that it must meet as we want it to be user friendly with the user interface, we also are setting a requirement that we want to make sure we can both hook this up to a speaker and plug in headphones.

The final deliverable for this project that are contained in the housing are these completed modules: an oscillator, low frequency oscillator, mixer, output amplifier, and wi-fi hardware interface. For the demo a pulse width modulation oscillator will be an external input. The first appendix explains the users manual on how to set up the synthesizer and how to test it.

Requirements Specifications

Functional Requirements

For our project we set out with the following functional requirements that we want our final project to be able meet. For the first requirement, we wanted all the modules to be functioning to expectations. Those expectations are defined below in the description of modules section. The idea behind this was that if all the modules were functioning correctly theoretically, they should all be able come together as one system in the end and be more likely to function. And if it didn't work out that way, we at least had a backup of each module individually working. The second requirement was that the interface would be able to change the modules to desired values. This was important because it's what makes our project stand out from similar products. We wanted to have an interface that could interact with our product. That's why this was a very important requirement for our group. The third and final requirement that we had was that the signal integrity was not degraded throughout the system. The was crucial to our group as our project deals with sound. Whenever you are dealing with sound it is very important to keep signal integrity and if you fail to you will know it as you will be able to hear it.

Non-Functional Requirements

Again, for our project we set out with the following non-functional requirements that we want our final project to be able to meet. One of these requirements was using a predefined app over Wi-Fi. We did this so we could ensure the security of the app's data is protected. Also, with using this app one can customize their own page layout however they choose. This allows for more personalization as well as allowing us to show more controllability attributes of our project. The other non-functional requirement was that the user interface functions with iPads, iPhones, and Android phones. This would allow for a wider range of potential users.

Use Cases

Uses for our project include, but are not limited to, studios, live performances, and amateur use. The goal of this was to really allow our product to be used in wherever musicians create/listen to music. With this we also wanted our product to not only be limited to musicians so we wanted to make sure our product could also be used in home use for amateurs to be able to use.

System Design & Development

Design Plan

We have decided to create a synthesis device that implements the basic sound generation functionality of conventional synthesizers, as well as some additional effect processing features and external control features. For the basic functionality we need eight separate modules; Voltage-controlled audio frequency oscillators, voltage-controlled low frequency oscillators, a multichannel mixer, voltage-controlled filters, envelope generators, voltage-controlled amplifiers, a user interface that generates control voltages, and a power delivery module. Figure 1 shows a block diagram showing how the basic modules will be connected inside of the device. With these 8 modules, the user can design new sounds by setting the available parameters and then playing those sounds with the keyboard. All the modules have their inputs and outputs available as control voltage sources and destinations on the hardware user interface. With these inputs and outputs, the user can reconfigure the internal signal paths. The basic modules have controls on the hardware user interface to set all the sound influencing parameters. The controls all go through the keyboard module's microcontroller which acts as the digital brain for the entire device. By having the controls go through the microcontroller, the user has the ability save and recall presets. The microcontroller also allows for a wired communication path via USB with an external PC machine. The PC machine can run a *Max* patch that communicates with an iPad through Wi-Fi. With the iPad the user can have a wireless user interface that mimics the hardware user interface with additional touch and motion controls.

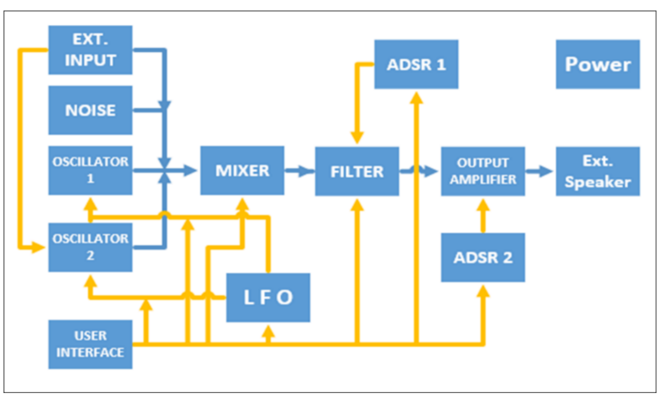


Figure 1

Design Objectives

There are six major design objectives that we planned to achieve to reach an integrated final project.

Objective	**Reasoning**

Module Concept	Within this stage we created a systematic design for the module. It will be clear how the module will interact the inputs and provide the desired output. As well as how it will interact with the control line from the microcontroller. Each module is different this stage results varied however each module from this could be represented as a box where its behavior can be explained.
Schematic Design	This stage creates the concept for how the module should behave. The schematic can then be tested on simulation software to provide proof of concept. This stage is important because it provides a list of parts that need to be ordered for the module. Our objective is to make this an affordable device the parts are then selected and tested accordingly.
Hardware Testing	The testing shows if the schematic functions properly. Within this stage data is collected to make sure that it reaches all the specs. Within this stage we are searching for improvements to make it better.
Hardware Integration	Once the modules before and after are complete they then need to be tested together to ensure that the output of one module does not affect the next. This is an audio device so there should be no significant signal integrity loss across the system
Wi-Fi Interface	All the controls come the graphical user interface. These controls send data over Wi-Fi that is then connected by the microcontroller that is within the housing. This microcontroller then needs to split the I2C bus line to communicate with all the devices.
Full Integration	This stage's objective is to prove that a control can be sent from the user's device over Wi-Fi, have it received by the internal microcontroller, have that control reach the module, and the adjust it accordingly. This will show the entire system integrates together and that the desired sound that the user wants can be reach in the output.

Table 1

Design Block Diagram

Figure 2

System Constraints

The main constraint on the project is time. The team has 32 weeks to design, build, test, and debug the synthesizer. This puts an upper-limit on the number of features (referred to as modules) that the synthesizer has. We decided that each member can reasonably oversee 3 modules. Based on these constraints, we have developed the block diagram shown above. Each block in the block diagram represents a module. We arrived at the conclusion that we could do 10 modules total. Each member has 1 main module worked on independently and 2 modules worked on with a group. With only 3 modules to work on, the initial design can be done by November. This deadline will give the team all of November and December to test the designs and develop PCBs. If we finish all these modules early, we can add more.

The other constraint that we need to work with is that most of us do not know how to code a microcontroller to have it adjust the digital potentiometers in each of the systems.

A challenge we will face at the end of the project is integration. We will all design our modules separately and test them by themselves. This does not guarantee they will all work together though. To mitigate this risk, we have decided to follow the line-level standard and have decided to make all modules have high input impedance and low output impedance. That way, all members know what signals they are receiving and what they need to output. This will be done with buffering outputs which will isolate the modules from each other.

The biggest challenge with this project will most likely be ADSR/envelope. Designing the envelope with analog will be extremely difficult as it will require a strong background in communications theory and circuit design. The envelope will be sent to multiple modules so it will need to work with them. We are considering sampling the waveform and doing this digitally to save time and make this easier. The reason this will be the most challenging is that it uses most of the major modules. Which means the controls of the envelope need to effectively be able to adjust the modules but not entirely overwrite the controls from the microcontroller. This design will be worked on once there is validation that all the other modules work synchronously with each other.

Description of Modules, Constraints, and Interfaces

Voice Oscillator

The voice oscillators provide 4 waveforms with frequencies in the audible range. The chip used to create the waveforms is the AS3340. It provides triangle, saw, and pulse. A sine wave is created by taking the triangle output and passing it through a wave shaping circuit. Each output is attenuated to 2.5Vpp to provide a consistent level for each waveform. A multiplexer is used to select the desired waveform. Before the final output, 2.5 VDC is added to the signal to allow the signal to be passed through digital potentiometers which operate between 0 and 5 V.

To control the frequency of the signal, the 1V/octave standard is used. The AS3340 was selected for its compatibility with this standard. For each additional volt applied to the input of the AS3340, the frequency of the output doubles (increases by 1 octave). Each octave has 12 semitones, so every additional 1/12 V increases the frequency to the next semitone. The MAX5215 was selected to provide the input for the oscillators. The MAX5215 is a 14-bit buffered voltage digital to analog converter. The reference voltage used is 3.3V. With two DACs, a voltage range of 0-6.6V is attainable. This corresponds to a 6.6 octave range. The MAX5215 was also used to control the duty cycle of the PWM output.

The AS3340 features a built-in temperature compensation circuit. Resistors and potentiometer values are selected to provide two equal currents to pins 1 and 2. This allows for the temperature compensation circuit to function.

The AS3340 also provides a sync pin on pin 9. The sync pin is connected to another oscillator's saw output. On the falling edge of the saw, the output of the AS3340 would reset. However, the functioning of this pin proved to be unreliable. Operation of this pin would occasionally fry the entire chip. This function was removed due to short supply of oscillators.

The TL082 op amp is used in the attenuator stage. This op amp was chosen for its low Total Harmonic Distortion (THD) of less than .02%.

Trim potentiometers are used for tuning the oscillator. The frequency range of the oscillator can be changed by adjusting these trimmers. The frequency can also be changed by changing the timing capacitor on pin 11. Increasing the capacitance by a factor of 10 decreases the frequency range by a factor of 10. The capacitor used on the voice oscillator is 1 nF.

All oscillators are built on breadboards. The original plan was to solder them to perf boards, but each oscillator has 83 parts. To create 4 oscillators as planned, 332 parts would have had to been soldered. With our skill level in soldering, we estimated that it would take almost 40 hours to solder.

Low Frequency Oscillator

The Low Frequency Oscillator (LFO) is an oscillator that provides output waveforms in the sub-audible range and the low audio range. The circuitry of the LFO is like the voice oscillators with some small changes to make it oscillate at lower frequencies and at a different amplitude. The following changes were made:

1. The timing capacitor is 100 nF. This moves the frequency range 100 times lower than the voice oscillator.
2. The trim potentiometers were tuned to different values to further lower the frequency.
3. The attenuator circuits attenuate the signal to 2Vpp.

The original purpose of the LFO was to provide modulation to different modules. This was to be done with a digital modulation bay. The mod bay would sample the output and convert the signal to I2C commands that would change digital potentiometers such that the potentiometers change with the same frequency as the LFO. The mod bay would also change the amplitude/depth. The LFO signal could also be routed directly to the input of the voice oscillator to provide frequency modulation of the voice signal. However, the Arduino could not sample and transmit fast enough to make the digital modulation bay feasible. The digital mod bay was then scrapped in favor of an analog mod bay that would simply route the LFO signal to the input of the voice oscillator. This means the LFO would have a fixed amplitude at 2Vpp.

Output Amplifier

The output amplifier circuit consists of two portions, the first portion is for volume control and the second is for ADSR. Volume control is just limiting what can go to the external speakers which results in a louder or softer sound depending on what the user sets the volume at. ADSR, which stands for "Attack-Decay-Sustain-Release" is a more complex portion compared to the volume control. This section adjusts how the signal initial starts when a key is pressed depending on how sliders are set on the user interface. With this the output amplifier is a simple circuit designed with a TL082 Wide Bandwidth Dual JFET Input Operational Amplifier and two digital potentiometers of DS1803 Addressable Dual Digital Potentiometer and the AD5144ABRUZ100 Quad Channel, 128-/256-Position, I2C/SPI, Nonvolatile Digital Potentiometer. The reasons these components where chose is since our project works heavily with audio waveforms. With the TL082 this op-amp is known for the practicality with audio related projects. The reason for the two digital potentiometers is that for the ADSR portion of the circuit it only needs a linear taper digital potentiometer. For the volume control portion of the circuit it requires a logarithmic taper digital potentiometer. After a signal goes through these two portions of it will then have to filtered to have the DC blocked and then it will be sent to the external speakers.

Some constraints that we ran into were that we were originally having troubles powering both external speakers and headphones. From here we decided to just use headphone ports there were available on the external amplifiers.

This circuit interfaces with multiple modules as its input is the output from the filter/mixer. From here the output is directly directed to the external speakers. This module also interfaces with the GUI interface as the goal would have it be able to control the volume and the ADSR values.

Mixer

The idea of the mixer is to use a summing amplifier to combine the two oscillators, noise, and three external inputs. All the inputs come it at 2.5Vpp which is the selected line-level. Once they are summed the output will not be higher than 2.5Vpp. This means that the values need to be attenuated prior to the summing amplifier.

This is where the user will have control and be able to select the percent of each of the input that they expect to receive in the output. An algorithm has been written that will set the values for the digital potentiometers that will adjust resistance to provide the output that the user wants.

To ensure that the output does not exceed 2.5Vpp the algorithm evaluated how many of the inputs have been selected to be zero percent. With every value that is not set to zero is the dividend of the input's voltage. From there the attenuation is increased depending on the percentage that the user selects.

The LM324 op amp was used for this summing amplifier for its compact 4 amps per chip, very low current supply drain, large DC voltage gain of 100dB. For these reasons it makes it an affordable amplifier that can successfully preform the desired summing amp without causing harmonic distortion.

The biggest constraint with the mixer is that there needs to be a 2.5V DC offset to ensure that the digital potentiometers will not saturate the input. This is something that needs to be accounted for before going into the mixer.

The mixer interfaces with the inputs coming in and sends the output to the output amplifier. The digital potentiometers are adjusted on the I2C command line that comes from the Wi-Fi microcontroller.

Noise

The noise generator creates noise from the avalanche breakdown of the BJT. To get the brake down voltage of a 2222 npn transistor 12 volts was placed above the emitter. According to the datasheet the brake down voltage from the emitter to base is 7 volts. It was found that this breakdown did occur, depending of the voltage across the BJT was much the mean and standard deviation varied. This was then amplified by a common emitter amplifier. To be white noise the standard deviation and the mean of the power density function must be equal. Since, this represents a perfect random noise distribution. This means that it is highly probable it will be constant at all frequencies. This was the goal of this device to provide a voltage rms output between 500mV and 1V.

The biggest constraint with this device was having it meet the design specs for the output. Since it is amplified noise the voltage output should not be measured in volts peak to peak. Since the random variation may cause it to go over. Amplifying this waveform and not saturating it was a challenge since the waveform needs a wide range of frequencies so it couldn't go through an amplifier with a low bandwidth.

This device creates a constant generation that is connected to the input of the mixer.

Filters

The design for the low pass and high pass filter is second order Sallen Key filters. Meaning that for low pass there are two sets of resistor-capacitor in parallel connected to the positive pin on the op amp with a unity gain going from the negative pin to the output. Similarly, for the high pass filter the resistors and capacitors are switched. The reason to do the low pass and high pass filter this way is because this will allow us to reach our specification of 12/24 dB per octave. The reason for a Sallen Key design is to achieve maximum flatness on the top of the frequency response, as well as they are commonly used to design filter circuits for sound/music applications. The resistors will be digital potentiometers to make the cutoff frequency adjustable through our digital user interface. The capacitors chosen were 100nF ceramic capacitors to achieve the higher and lower extremes of the cutoff frequencies. The op amp chosen is a TL082. The reason being is this op amp's application is audio, in which it has very low harmonic distortion. Which is what we desire in order to achieve perfect harmonics, or as close to perfect as we can get. We want to have the cutoff frequency to be adjustable between ~20 Hz and ~20kHz, which is the audio range. These values are the frequencies of each octave from the first to the tenth octave. Having the cutoff frequency adjustable through this range allows the user to determine what frequencies they want to let through to create the sound they desire. The transfer function for a low pass filter should have a zero in the numerator that is at infinity, meaning the numerator should be a constant. The high pass filter was designed in a similar way. The transfer function for high pass should have a numerator that is equal to the gain*s^2 which is just s^2 because the gain is a unity gain. The transfer function for the low pass is equation 1, and the high pass is equation 2 when the resistor values are 91k Ohms to achieve a cutoff frequency of 18 Hz, both are shown below.

$$\frac{12075.83625166}{s^2 + 219.78021978022s + 12075.83625166}$$

Equation 1

$$\frac{s^2}{s^2 + 219.78021978022s + 12075.83625166}$$

Equation 2

A major constraint we ran into was getting the digital potentiometers to function properly. It took longer than expected to figure out how to digitally write to the potentiometers and confirm that they indeed changed in value. After this was solved, we ran into a bigger problem when the potentiometers were integrated into the filters. They caused the output to drop significantly and we were essentially seeing a DC bias that killed our signal going into the filter. We were unable to figure out where the problem was arising or why it was occurring.

The filters receive a signal coming from the mixer and sends a signal to the output amplifier. The digital potentiometers are adjusted on the I2C command line coming from the Wi-Fi microcontroller.

Design Trade-offs

Filters

The original layout for the filters were fourth order active circuits. We thought this was the best way to reach the 12/24 dB per octave as well as other specs. After thinking about it more and after trying to implement a fourth order filter, we realized we can reach the same outcome by having a second order filter. This realization made implementing the filters much easier without losing effectiveness in the overall design of the filter module.

Voice Oscillator

The original proposal consisted of the AS 3340 voltage-controlled oscillator chip followed by attenuation and DC blocking circuits. The intention was to provide an audio signal that was 2.5Vpp with no DC offsets to the next module. However, the digital controls were implemented through digital potentiometers which only allows signals in the 0V to 5V range. This resulted in the negative cycle of the signal being clamped to 0V. To solve this issue, the following circuit was put in series with the final output to add a 2.5V DC offset:

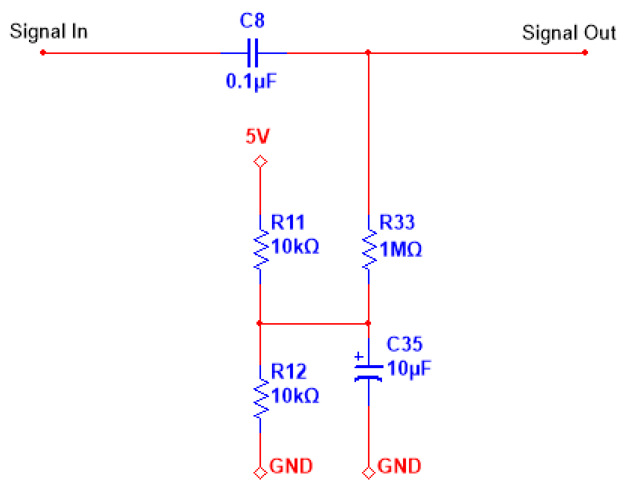


Figure 3

Capacitor C8 removes the slight DC offset that is present in the signal. Resistor R33 provides a high load impedance to the voltage divider to provide an offset close to 2.5 V. With the offset, all digital potentiometers that are used later in the circuit provide the expected results.

Low Frequency Oscillator (LFO)

Due to the switch from a digital mod bay to an analog mod bay, the attenuator circuits for the LFO needed to change. Instead of sampling the LFO then scaling it to provide a variable depth, the LFO signal will have a fixed amplitude and fixed depth. The LFO will then be routed to the control voltage input of the AS 3340.

Output Amplifier

With the output amplifier it went through numerous modification and updates. These changes in the circuit were both due to changes in the project specification, learning more about the project, and failures. The project first started out with the idea of having on board speakers. With this idea in mind the output amplifier needed to have a power amplifier to be able to drive the speakers. With this we decided to use a Class-D audio power amplifier. With this output amplifier circuit [Figure 4] when we tested the circuit it was having numerous problems and was not outputting any sound. After some help from our advisor it came to see that the part didn't have specification that would prove to be useful to our project as the THD was way out of the range of what we want it to be. From here our group decided by the group and approved by our advisors to use external speakers and eventually to use the headphone port that is on the external speakers. The reason for using external headphone ports was with the circuit we had it was having troubles driving the speakers and headphones and due to a time restraint, we decided to move in that direction. These ideas allowed us to come to the point of where we are not with

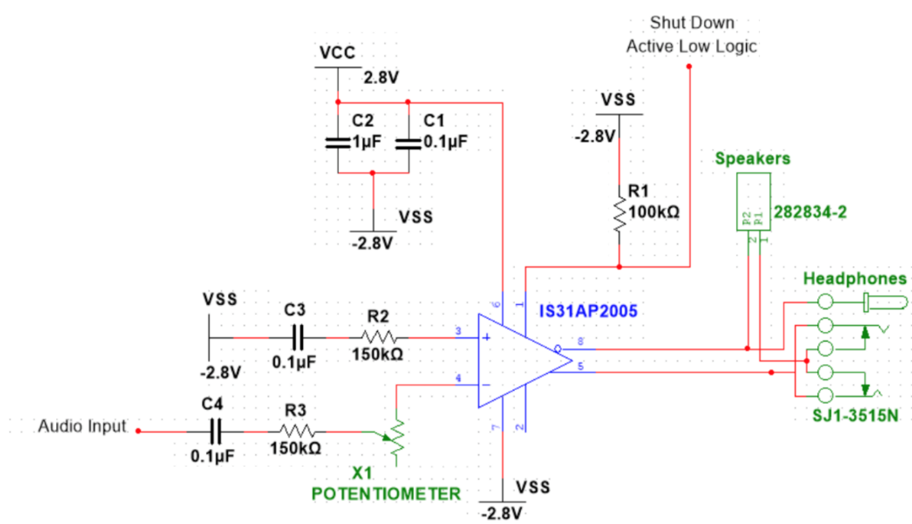


Figure 4

Mixer

The initial design of the mixer was to have the output have a constant line-level voltage of 2.5Vpp. This was done by having an algorithm that contributed for the percent attenuation in all the inputs of the summing amplifier then to regulate the feedback path to ensure the output was 2.5Vpp. To make sure the output would remain at 2.5Vpp the output went through an automatic gain control circuit which had a peak detector and comparator as the control network.

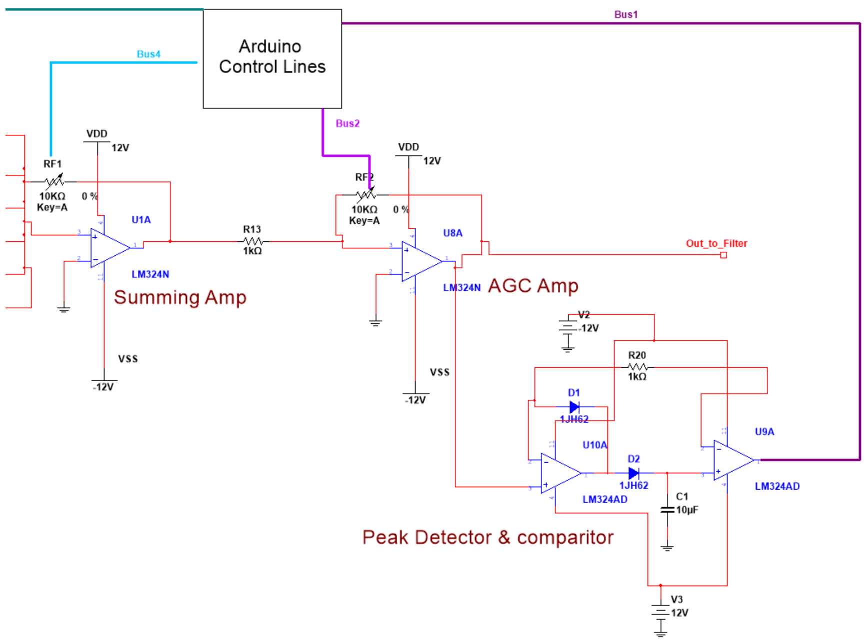


Figure 5

Although this design functioned it was scrapped because it had flaws that took away from the end goal of the design. The biggest flaw in this design was that the output should not be regulated to stay at 2.5Vpp. This takes away the idea of properly attenuating the inputs to the mixer, because the output is always increased to maximum.

For this reason, the algorithm for the feedback resistor was not needed as well as the automatic gain control circuit. Thus, resulting in a circuit that used significantly less components, less inputs/output to the microcontroller, and less space in memory. This over designed the circuit did provide insight on how to use the digital resistors efficiently which was useful in other modules.

Implementation

Implementation Diagram

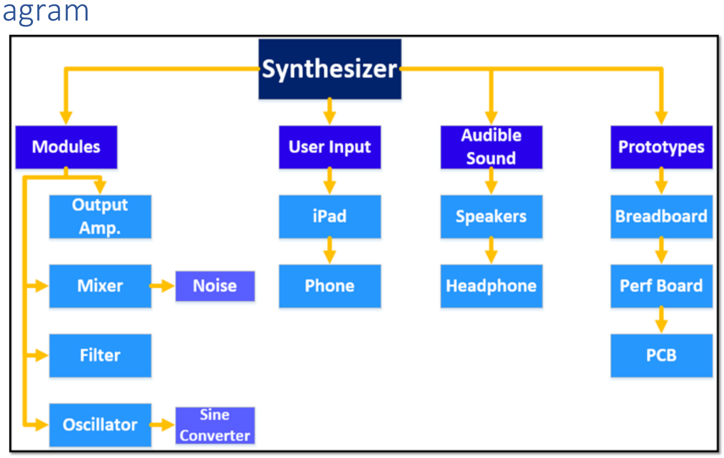


Figure 6

Technology and software used and rationale

Software	Rational
Multisim	This is a program that has a user interface that is preferred for testing and designing circuits. It is an easy program to create new chip designs. Its interface with Ultiboard makes it easy to cross over to do circuit PCBs.
Ultiboard	The cross over between Multisim makes this a great program to do the PCB design on. Since, all the schematics are on this program already. This program provides a simple way to adjust and create a new footprint for a chip.
KiCad	One of our members does not have access to Coover's labs since he is working in Cedar Rapids. This is a free program that is preferred by him to develop PCB design.
Arduino (Technology)	The Arduino is the microcontroller that we selected to be within our system. The tests done on each module used an Arduino Mega. The integrated systems run with the Wi-Fi internet development board that uses the same algorithms that were used on the Arduino Mega.
MATLAB	This program is a useful calculator that processed the data from the noise to create the power density function. Along with doing statistical analysis to find the standard deviation and mean.
TouchOSC	This is the app that was selected to be the graphical user interface of our system. This app allowed us to use the existing shapes and be able to process the data that comes from interacting with the board.

Table 2

Applicable Standards and Best Practices

For frequency control, we will follow the 1 volt per octave standard used in industry for analog synthesizers. This means for each 1-volt increment in control voltage, the frequency of the oscillator will double.

All modules will operate at line-level. This is a standard in the audio industry. The line-level that our synthesizer will use is 2.5Vpp. Each module's input impedance should be very high and each module's output impedance should be very low, i.e. the output should be buffered. This will ensure everyone on the team knows what type of signal they're receiving.

The communication protocols I2C and OSC standards will be used to communicate from the microcontroller to the digital potentiometers. This standard will be followed in all the modules, so that on the microcontroller there is no shift in what is expected in its output to the digital potentiometers and every member of the group understands what the others will be using for their control signals. So, once all the modules connect there will be less confusion of communication from the microcontroller.

We will be following the IEEE code of ethics. The IEEE code of ethics is straight forward and will not be broken. Since our synthesizer is intended for use by the public, we need to observe the first rule [7]

1. to hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, and to disclose promptly factors that might endanger the public or the environment;

Since we are working as a team, we will take special care to observe the following standards as well:

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

8. to treat fairly all persons and to not engage in acts of discrimination based on race, religion, gender, disability, age, national origin, sexual orientation, gender identity, or gender expression;

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

We will be using the test benches in Coover Hall. We will follow the standard lab safety procedures developed by the Department of Electrical and Computer Engineering [8].

Testing Process/Testing Results

Test plan

Each module will be tested with this procedure. If any mistakes are found the testing will go back to the previous step and see if the problem can be addressed.

1.	Outline of Schematic complete.
2.	Schematic tested in NI Multisim.
3.	Schematic will be assembled on a breadboard.
4.	Testing on the breadboard will be done in Coover.
5.	Schematic will then be assembled onto a perf board.
6.	Testing on the perf board will be done in Coover.
7.	Schematic will then be assembled onto a PCB.
8.	Testing on the PCB be done in Coover.

Table 3

Every module needs to be tested. Their difference in what is expected from those tests of the modules is explained in the validation section. The work for all of them has been divided and will be worked on simultaneously.

Module Testing

Voice Oscillator

A functioning voice oscillator provides four output waveforms: saw/ramp, triangle, sine, and pulse. The frequency of the waveforms is determined by the voltage applied to the control voltage pin on the AS 3340. The AS 3340 provides 1 volt per octave mapping. This means increasing the control voltage by 1 volt will double the output frequency.

To test this, the voltages are applied to the input and the frequency of the various outputs are measured. At low frequencies, the just noticeable difference between two semitones is about 5 cents. If measured frequencies are within 5 cents of the expected frequency, then it is considered acceptable.

Low Frequency Oscillator

Since the LFO is used for modulation, the accuracy of the 1 volt per octave scaling is not important. What is important is the oscillator's ability to reliably produce output at low frequencies (below 1 Hz). To test the low frequency capabilities, 0V is applied to the control voltage input. The oscillator is then tuned to progressively lower frequencies. With a .1 µF timing capacitor, the circuit can easily be tuned to frequencies lower than .25 Hz.

Output Amplifier

For testing of the output amplifier circuit, it needs to be able to take an input with a 2.5Vpp. Since the circuit is composed of two sections (volume control, and ADSR) it could be tested in two different ways. Coming out of the first section, volume control should be an attenuated version of the input signal at a level that is dependent on what the digital potentiometer is set at. Coming out of the second section the ADSR portion will [be a signal that does from non-existent to the signal that is coming out of the volume control at a rate that is also dependent on how another digital potentiometer is set. The signal couple also hold at the desired max level for a set amount of time depending on how the sliders are set on the GUI. This section is difficult to simulate as the digital potentiometer will be programmed to change its values at certain intervals so trying to simulate that for over 256 tap position with the timing is almost impossible. Following the design, it was then put on a breadboard and perf board and tested. The algorithm for this section goes as the following. It has two functions, one for the volume control and one for the ADSR section. In the main if the volume slider is adjusted on the user interface it will trigger a function that takes the slider value and sends it to the function to change the digital potentiometer to the desired value. For the ADSR section, this function is triggered whenever a key is pressed on the user interface. From here it will trigger the ADSR function with the value of the slider for the four parts A-D-S-R. These values will then be in the function used to calculate the rate that the resistance value is change which in turn will affect how the signal is modified and how it sounds. The figure below shows how the digital potentiometers on the output amplifier circuit change.

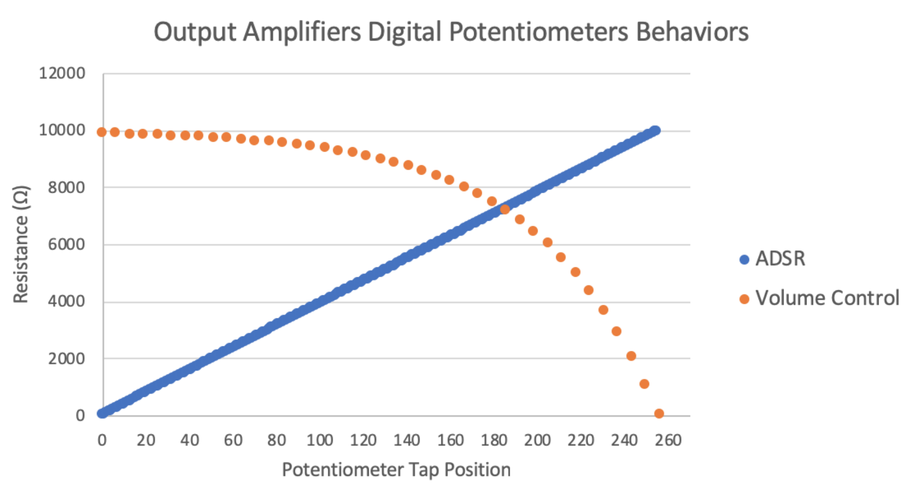


Figure 7

Mixer

To have a functioning mixer the output should not exceed the line level voltage of 2.5Vpp. The way that this circuit was tested was by using all six of the inputs and sending in different signals and varying their percentage. What was tested and measured was the voltage from the output and did that correlate with the expected value from the percent the user selected. The percent attenuation is driven by a digital potentiometer which was controlled by an Arduino during testing.

The error bounds where measured on this device to see if it would negatively affect the desired output. It was seen that these devices error bounds would not distort the signal or be off where the user would not be able to select a value.

This image below shows the combination of a square wave being generated from one oscillator and a sine wave being created from another oscillator. This test result below showed full integration between the hardware and software as the user is able to adjust the percent desired on the controls of the user interface

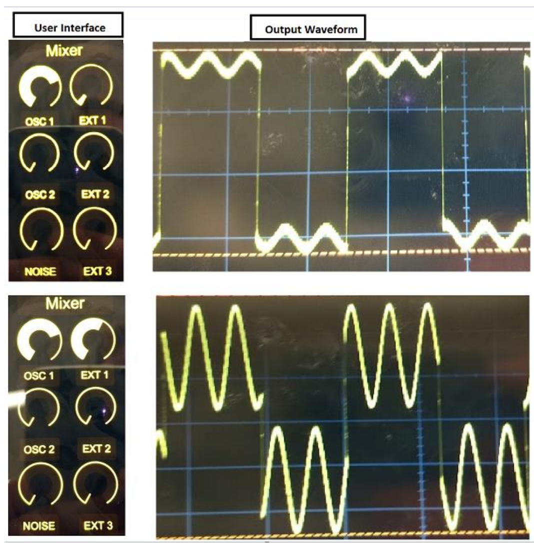


Figure 8

Noise

To have a functioning white noise generator the output needs to be random in the time domain, and the standard deviation and mean are equal in the frequency domain. Since this is a musical device, the frequencies that are of interest are between 20Hz to 20kHz. Also, this device needs to work consistently for long periods of time, so it was tested at multiple periods.

To test this device, the power rails were connected the output and went into an oscilloscope. Using the output from the noise, the values were exported to MATLAB and analyzed. Below are the results in the time domain: output voltage & distribution of noise.

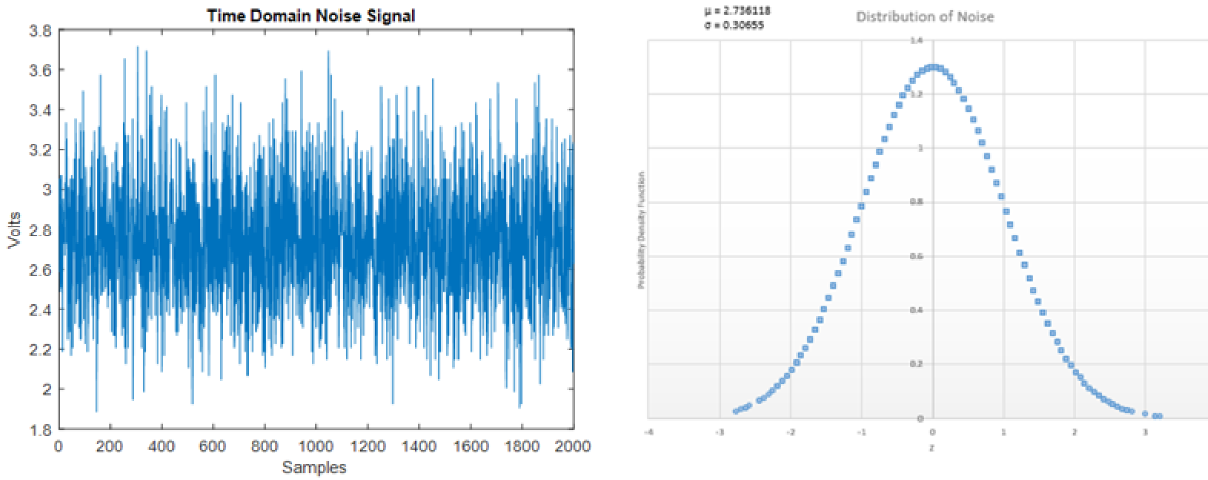


Figure 9

These results confirm that the output voltage for the noise source is random since it perfectly follows a Gaussian distribution. This shows that the output measured as Vrms will be the most useful way to analyze the output of the noise generator.

Below are the results in the frequency domain: power spectral density and distribution of the power density function.

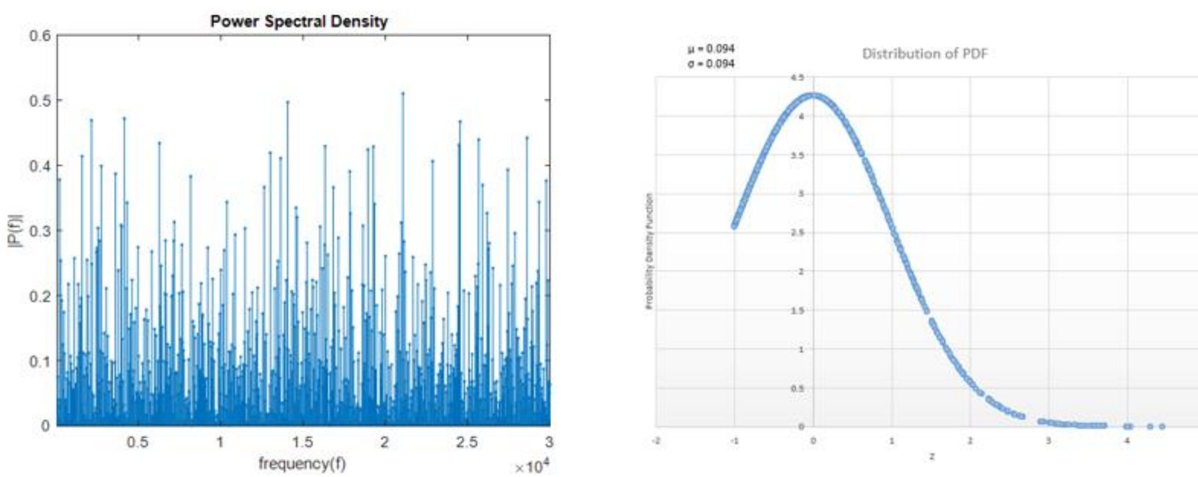


Figure 10

These results show an ideal white noise where the mean and standard deviation are equal. Something that is noticeable in the distribution is that it is cut off on the left slope. This was expected since the power density function is the second power of the Fast Fourier Transform (See Appendix 2). For this reason, there will be no negative values, hence the left wing is cut at 0.

Filters

The testing of the breadboards was done by sweeping the function generator through the audio frequency range. The peak-to-peak voltage was recorded with the corresponding step in frequency. This was then converted to logarithmic and then plotted which showed where the –3dB cutoff was. This process is shown below in Figure 11. The y-axis of this plot is the dB, so going across on –3 where that crosses with the data points is the cutoff frequency, as shown with the blue line and highlighted data. The expected corner was 3500 Hz. We achieved a corner of about 3750 Hz which is less than an eight percent difference. The percentage difference was not all that important. What was important was that we were able to sweep through the entire audio range. The resistor values were chosen based on how accurate the cutoff was after the sweep with what we initially expected. For example, if we expected the corner frequency to be 440 Hz and after the sweep, we achieved a corner of 350 Hz. We would need to adjust the resistance to get closer to the 440 Hz. This process was done for both high pass and low pass filter on the low-end frequency (20 Hz), mid-1 frequency (440 Hz), mid-2 frequency (3500 Hz), and for the high-end frequency (20k Hz).

Hz	pk-pk	20log	Expected corner: 3500 Hz
10	1.09	0	226.5 ohm
50	1.09	0	
100	1.09	0	
500	1.07	-0.1609	
1000	1.05	-0.3247	
1500	1.01	-0.6621	
2000	0.97	-1.0131	
2500	0.91	-1.5677	
2600	0.9	-1.6637	
2700	0.89	-1.7607	
2800	0.88	-1.8589	
2900	0.87	-1.9581	
3000	0.86	-2.0586	
3100	0.85	-2.1602	
3200	0.84	-2.2629	
3300	0.83	-2.367	
3400	0.81	-2.5788	
3500	0.79	-2.796	
3600	0.79	-2.796	
3700	0.78	-2.9066	
3800	0.77	-3.0187	
3900	0.76	-3.1323	
4000	0.75	-3.2473	
4100	0.74	-3.3639	
4200	0.73	-3.4821	
4300	0.72	-3.6019	
4400	0.7	-3.8466	
4500	0.69	-3.9715	
4600	0.68	-4.0984	
4700	0.68	-4.0984	
4800	0.67	-4.227	
4900	0.66	-4.3577	
5000	0.64	-4.6249	

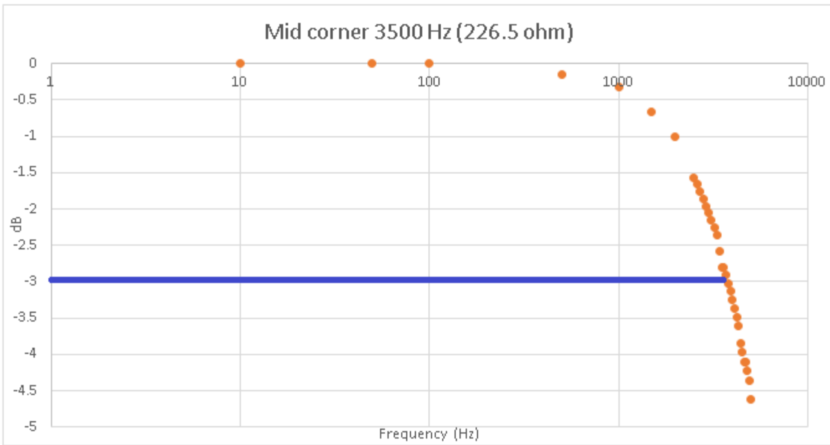


Figure 11

Interface and Integration Testing

To have a functioning user interface the user should be able to interact with the touch display and set the hardware configuration to change or play the sound being synthesized. Each module has a dedicated control section on the graphical user interface. To test the interface, we first had to test that the interface would generate to correct control messages using the Open Sound Control protocol. The next step is to confirm that the messages from the GUI are being received by the Wi-Fi hardware interface. The Wi-Fi hardware interface consists of a microcontroller board with an integrated Wi-Fi transceiver, level shifters, I2C bus multiplexer, and GPIO expanders. Using the Arduino IDE, a test program was made that received the incoming messages and printed out the message to the serial monitor. After all the controls on the interface were tested, we had to test the code that set hardware configuration. To test that program, we first looked at just the Wi-Fi hardware interface to make sure the correct I2C bus was being selected, and that the correct bits were being set on the GPIO expander registers with each control command from the GUI. Once the functionality of the Wi-Fi hardware interface was confirmed, we tested communication between the Wi-Fi hardware interface and each individual module. After we confirmed that each module could be individually controlled with the GUI, we combined all of the modules and tested communication with the full system. To conduct the final test we used the oscilloscope to probe key locations and then ran through all of the controls for the available modules on the GUI. Looking at the wave form at the key locations, we are able to confirm the system is working as expected.

In the audio rate oscillator section, we confirmed that the note pitch can be offset by two octaves with a knob, and by the full range of eight octaves with buttons. For the pulse wave form, we also confirmed that the pulse width could be modulated with a dedicated knob. Finally, with a button we are able to select one of the available wave forms from the oscillator. Other sections of the GUI that control the oscillators are the keyboards and XY touch pad. A key press or XY touchpad press sets the fundamental note that will be played. The offsets are applied to that fundamental note.

The low frequency oscillator section is very similar to the audio rate oscillators. Main difference is that there is no precise frequency selection, just a range to select from with the rate knob. For this module we confirmed the rate knob made the full range frequency range of the LFO available to the user. PWM and wave controls are the same as the audio rate oscillators.

The ADSR section needs to set the correct values for the digital ADSR algorithm. It needs to be done in a way that the settings are logarithmic to give more precision to the small amounts of attack, decay, and release. Along with setting the correct values, the algorithm was tested to confirm that it could run and continuously set the envelope for each note pressed.

In the mixer section we tested that each knob set the correct percentage and that it was set for the correct input.

Since the digital filters were not accomplished, the filter section only had the functionality of switching between no filters, a fixed passive low pass filter, or a fixed passive high pass filter.

The master volume section was tested by looking at the output signal and seeing that the volume knob applied a logarithmic scaling to the output level. This is done so that the human ear interprets the amplitude difference correctly.

Modulation bay section had some issues that we were unable to debug. Due to time constraints, we limited the functionality to only turning on or off the low frequency oscillator that is routed to the oscillator one pitch.

Patch saving was also not implemented due to time constraints.

Validation and Verification

Below is a table that shows the validation of each module. Once the tests were completed each of these had to pass.

Module	**Testing expectations**
Oscillator & LFO	– Output is at line-level 2.5Vpp – Output wave form can be modified using the digital potentiometers.

	– Creates four types of waveforms sine, triangle, saw, and pulse.
Noise	– Output is 1Vpp – Output power spectral density's standard deviation and mean are equal – Time domains output shows a random distribution. – Audible that sounds like noise and does not sound distorted.
Mixer	– Output is at line-level 2.5Vpp – Algorithm ensure that the output remains below or equal to line-level and the digital potentiometers are properly adjusted. – Output is properly modified by adjustments in potentiometers.
Filter (high & low pass)	– Analysis of the circuit that shows the 3dB drop is at the ideal location – Pass band is flat. – Output is properly adjusted by modifying the digital potentiometers.
User Interface	– All controls work as expected for modifying the circuits. – The Wi-Fi connection is not disrupted by continuous sending and receiving signals. – User interface is accepted by users. This will be done by sampling it on a group to see what opinions they must improve it.

Table 4

Below is a table that shows what makes each test form validated.

Test Form	**Validation & acceptance**
Multisim Simulation	The simulation test will show proof of concept that the schematic works as expected. This is a lower strain environment then on the test bench where parts can be changed in dimensions easily. The validation is confirmed where the output matched what is expected from the given Vdd and Vss voltages.
Breadboard test	This test will be a hardware validation. The hardware test will show possible limitations of the components that where selected. Validation is reached once the proper output is achieved with the given Vdd, Vss, and input voltages.
Perf Board test	This test is a continuation of hardware validation. This will show that the parts work when soldered together and that the module works for any given circumstance or corner case that could happen in the application of the synthesizer.
PCB Testing	This is ideally the final test of the module. This validation shows that the module works as expected. This shows that the designer of the module and who they have check their work accept that this module will function for all given circumstances it will come across. If there are any problems in this state, then the procedure needs to be repeated in one of the above test forms to figure out where the mistake originates from if it cannot be detected on the PCB.

Table 5

Project and Risk Management

Task Decomposition & Roles and Responsibilities

To be able to get this entire system done we divided and conquered each module. Each member was assigned a minimum of one module to accomplish. Once that module was completed, they would then begin working on the integration of that device with the ones before it and after it along with testing the Wi-Fi interface.

Below is a table that separates the tasks that were completed/major contributor each member.

Tasks	Tim Day	Francisco Alegria	Blake Beyer	Travis Gillham	Eric Fischer
1	Mixer	Wi-Fi Hardware Interface	Audio Rate Oscillator	Output amplifier	LP/HP Filter
2	Noise	GUI	Low Frequency Oscillator	Master Volume Code	Filter Algorithm
3	Mixer algorithm	Integration Testing	Oscillator Algorithm		
4	Integration testing	Integrating & Debugging code	Integration Testing		
5	Website	PCB Design			
6	Case Design	Modulation Bay			
7	PCB Design	System Code			
8		ADSR Code			
		Filter Bank			

Table 6

Project Schedule - Gantt Chart (proposed vs. actual)

Below are Gantt Charts of what was proposed:

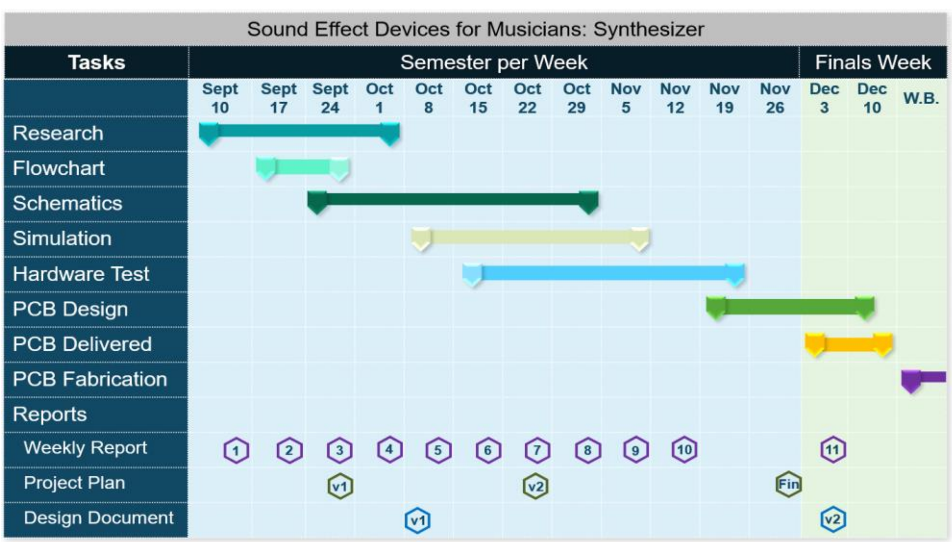


Figure 12

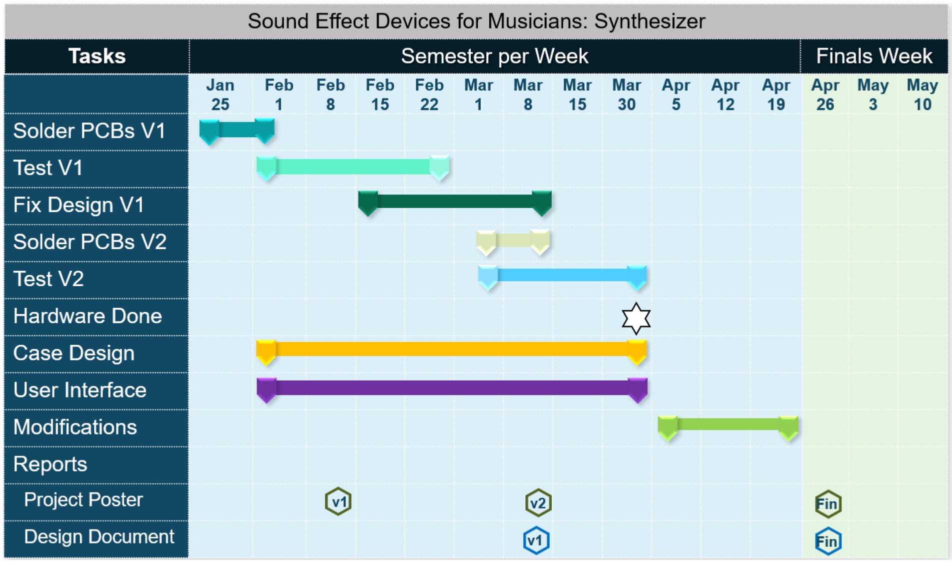
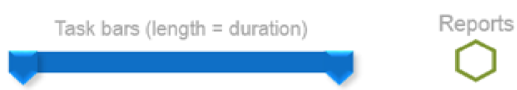


Figure 13



Below are the actual Gantt Charts:

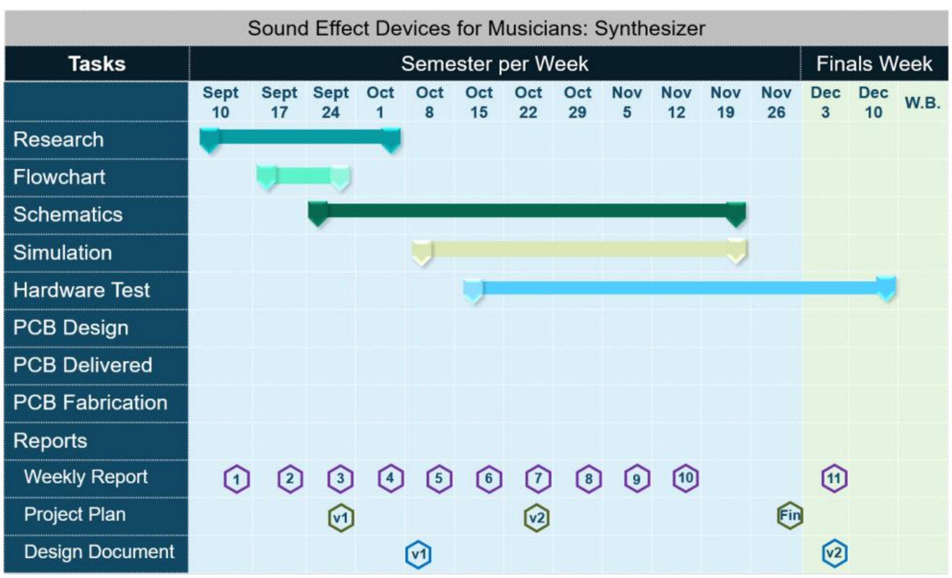


Figure 14

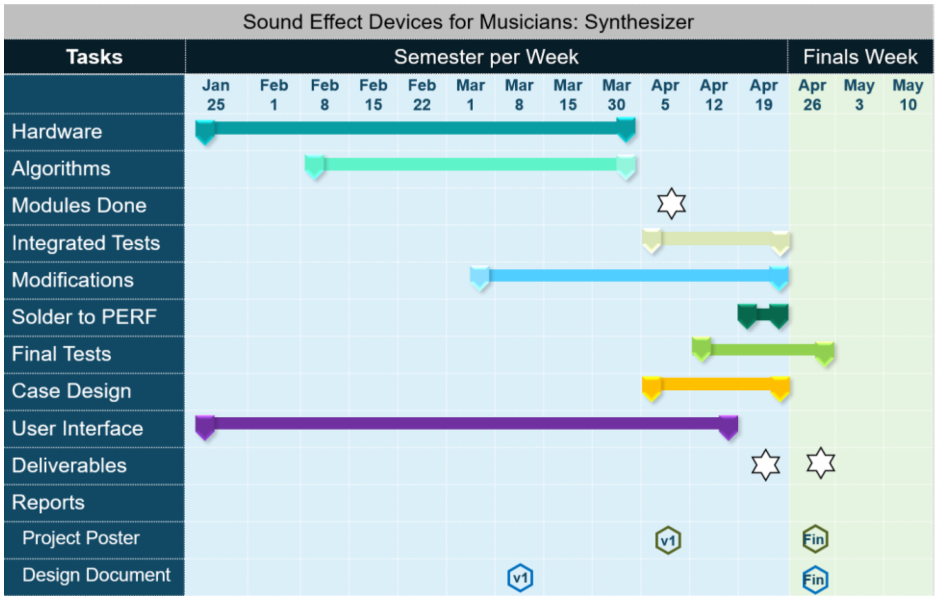


Figure 15

What ended up taking much longer than expected was testing each of the individual modules. Fixing them to meet the desired specs from the schematic was simple for some modules but challenging for others. This delay in progress resulted in a delay getting the PCBs. When the PCBs arrived, there were mistakes in the oscillator's schematic that ended up making those PCBs ineffective. This was accounted for when creating the housing of the system. The other delay that ended up being the hardest hurdle was connecting all the modules onto one bus line. This overdrove the I2C bus and was a problem that we thought we could avoid with the modulation bay. With the lack of time testing the full integration lead to a product that could not work off one microcontroller.

Risks and Mitigation: Potential (anticipated) vs. Actual (happened) and how they were mitigated

Throughout this project there were constant hurdles that needed to be jumped over that were anticipated

Potential Problems	**Risk management**
Time constraints	Updates on progress are given weekly. We meet for 3 hours a week as a group during the week and for 6 hours on the weekend to make sure all the tasks get accomplished.
PCB mistakes	Perf boards are a prototype that is used for testing to confirm that the circuit worked.
Parts malfunctioned	We are finding affordable parts so they can be replaced when they brake.
Pre-integration testing	Every module must function on its own. Meaning that the signal integrity does not degrade across it, when the circuit before and after it is connected. Also, it can work with the Wi-Fi connection to control the device from a separate microcontroller.

Table 7

Actual Problems	Risk management
Time constraints	Members of the group are passionate about this project and put in the extra hours to make sure that their task and other group members tasks are completed.
PCB mistakes	A perf board was made for each module except for the oscillators. To get a working circuit this was made 3 times on breadboards and the casing was designed around it.
Wrong parts delivered	We were dealt a couple time were the wrong part was delivered we worked around this by using parts that we knew worked to compensate for this loss.
Bit Line Voltage	The microcontroller sends out data at a 3.3V level and the circuits within the hardware run on a 5V level. To fix this we created a circuit that will step and down this bi-directional signal.
I2C Bus Problems	When testing we ran into problems with the I2C bus. This was due to not combining the entire system onto one bus line early enough to test this. However, we will avoid this problem by using more than one microcontroller.

Table 8

Lessons learned

Throughout the duration of this project our group learned numerous things about the design process and implementation that goes into completing a project of this size. One of the biggest things we learned was time management. Always give yourself more time than what you originally think. Working on a project like this it is almost certain that at some point you will run into some road blocks along the way. Giving the group some extra time will allow for the project to not be rushed or in a time crunch. Another thing our group learned while working on the project is to make sure everyone is on the same page at the beginning of the project. Going into this project our group consisted of five members and only two of these members had experience or knowledge with musical instruments. Picking up on music knowledge is one thing but having to create a project that implements those ideas was a whole different story and was a problem that we ran into numerous times throughout the duration of this project.

Closure Materials

Closing Remarks for the Project

The current equipment in today's world is very expensive. Our goal was to create a synthesizer that anyone can use. A synthesizer is an electronic music device that creates audio signals that will be converted into sound. The synthesizer we designed utilized a digital user interface via tablet or smartphone that is used wirelessly. This interface allows the user to customize their own layout based on what they want while having the ability to move more freely and accommodate a wide variety of playing styles. A traditional synthesizer usually has physical knobs and switches directly on the case. We think the digital interface will allow for more customization as well as provide ways of manipulating the sounds and waveforms that could not be done using knobs. For example, having an XY-plane on the touch screen that when you slide your finger across it will adjust one aspect of the sound and when two fingers slide across it adjusts a different aspect. The housing for the system has the PCBs on the top and the breadboards for the oscillators underneath the false floor. This was made to make wire management easier for the final device. Challenges did arise, but we did have a good approach to eliminate these issues. Part of the approach was to split up the project into modules and assigning each member to take charge of a specific module, as we discussed in our project tracking procedure. In the end we were successful in being able to accomplish our goal.

Future work

There a few things to consider for the future. One of those is fixing the I2C bus to communicate to all devices. When doing the system integration testing, we ran into a major fault where we overloaded the I2C bus which caused the devices to not communicate. If we could fix this issue and not have an overloading bus, then all the devices could communicate to each other and we could run our tests smoother. Another part to work on in the future would be the filter module. As stated earlier, the filter module did not get finished because of the problems associated with communicating with the digital potentiometers along with the digital potentiometers killing the input signal. Figuring out why the potentiometers are causing this signal loss would allow us to finalize the code to be able to choose between the filters. The other part to work on in the future would be changing the oscillator PCB design. The layout needs to be changed in a way that allows for easier testing that reduces the chances of burning out components.

Major References During Design Process

[1] Novation Music, *Bass Station II User Guide*, Focusrite Audio Engineering Limited, 2013.

[2] “List of classic synthesizers,” Wikipedia, 23-May-2018. [Online]. Available: https://en.wikipedia.org/wiki/List_of_classic_synthesizers. [Accessed: 01-Dec-2018].

[3] Ray Wilson. *Make: Analog Synthesizers.* Sebastopol, CA: Maker Media, 2013.

[4] CMOS Analog Circuit Design. Third edition. Phillip E. Allen, Douglas R. Holberg

[5] Sedra A. S., & Smith, K. C., “Microelectronic circuits.” 7th edition, Rinehart and Winston.

[6] B. M. Wilamowski, J. D. Irwin, “Noise in Semiconductor Devices,” in Fundamentals of Industrial Electronics,2nd Edition, CRC Press, Boca Raton, Florida.

[7] “Language Reference,” *arduino.cc*, 2018. [Online]. Available: https://www.arduino.cc/reference/en/. [Accessed: Nov. 30, 2018].

Team Information

Tim Day

Tim has work experience from Micron, Orbital ATK, and Clow Valve that have helped prepare him in developing a circuit design from start to finish. He is an outdoor enthusiast that highly enjoys mountain biking, hiking, trail running, and traveling. Tim will begin his career at Texas Instruments as a characterization engineer working on analog signal chain for temperature and humidity sensors. He is excited for this opportunity and is looking forward to moving to Tucson, Arizona.

Francisco Alegria

Francisco has musical experience in performance, theory, music production, sound design, and software instrument design. He also has experience in leading teams for a radio and TV station. His passion for music has led him to learn how to design discrete component circuits and integrated circuits with the intention to use those systems for musical applications and signal processing. Francisco's career goal is to be a mixed signal IC design engineer with a focus in analog IC circuits. He wishes to be part of an organization that provides analog and digital integrated circuit components. To achieve this goal, he is looking to continue his education at Iowa State University pursuing a Master of Science degree with a focus in analog integrated circuit design.

Eric Fischer

Eric has worked at NSK Americas which has given him experience in taking a project from start to finish; going from ideas to design modeling to building and fabrication. He has a passion for cars and hopes to get a full-time position at a company that specializes in designing innovative solutions for any kind of vehicles. He loves the traveling and the outdoors especially playing sports, hiking, and boating. Eric is very interested in system and controls and hopes his career is in this area.

Travis Gillham

Travis has work experience from John Deere, ALMACO, and Hawkeye Pedershaab that has helped in preparing him for a career in Electrical Engineering in the area of controls. Outside of academics Travis enjoys anything that involves being active as he not only enjoys watching sports but playing them as well. Following graduation, Travis will be working as an Electrical Engineer at John Deere in their Power Systems division.

Blake Beyer

Blake has previously interned at Glow Networks in Richardson, Texas. This has given him experience in RF design and managing technicians in the field. Blake plays guitar and has designed and fabricated a fuzz effect pedal giving him relevant experience in musical effect design. Blake is interested in sales engineering and competed in the 2018 National Sales Engineering Competition in Orlando, Florida. He hopes to pursue a career in this area.

Appendix 1 – Operations Manual

Setting up the system

To set up the system one would first want to connect all the power rails inside the case to the corresponding voltage levels of the power supply. After that is finished and double checked, you would then want to hook up the speakers to the output of the box. For the interface you would want to, before you start, configure the pages to your liking. You can do this first by going to hexler.net. Here you can navigate to find the download for the application TouchOSC Editor and from here you will download this on to a desktop or laptop of your liking. When this is downloaded you will start off with a blank page and from here you can go in and edit the pages and what they look like. You do this by selecting the options you would like to have on the page and editing their parameters in the app and then you will go in and with your code you will modify how those options interact with the system. Once you have the code modified to match the parameters with the page you created, you can sync the page with the app by having all devices connected to the same Wi-Fi network. This will also allow you to be able to play on different devices. If you want, you can download the TouchOSC app on your cellular device and use it on there or you can download it onto your iPad and use it on there. Finally, set a few parameters and change them a couple times to make sure you are getting the sound you expect. This will tell you if everything is connected correctly or not.

Demo the system

Once the setup is confirmed, select a few parameters you would like to showcase. Go through a detailed explanation of how the demo is going to work and give a short example of what you want the user to do. Then, give the interface to the user and have them run through and change some values of these parameters. You should have a good understanding of what to expect depending on how the user changes the values. Once a value has been changed, explain what happened and why it did change the way it did. After changing the parameters, ask the user if they understood what happened when they changed a value of a parameter. This will let you know how well you explained what was going on. Then have the user change a few more parameters, however, this time see if they can give a reasonable explanation of what should happen when they change a value. If they can give reasonable explanations, then that means you conveyed what you needed in order to have the user understand the system.

Test the system

For testing of the system, the synthesizer should be able to be tested and used for multiple songs by multiple users and be able to perform at the standards each musician would like to be at. For this to happen each musician will more than likely have their different set up of pages on the interface so it will need to be able to change the interfaces each time a new musician comes up. With this the product all needs to be able to function properly for each musician.

Appendix 2 – Code

Mixer: Frequency Response

```
Fs = (ms3(2,1)-ms3(1,1))^-1;          % Sampling frequency
T = 1/Fs;            % Sampling period
L = 2000;             % Length of signal
t = (0:L-1)*T;        % Time vector


F = fft(ms3(:,2)) %Fast Fourier Transform of output

pow = F.*conj(F)/2000 %Power Spectral Density Equation

stem(f,pow,'.')
xlim([50 30000])
xlabel("frequency(f)");
ylabel('|P(f)|');
title("Power Spectral Density");

plot(ms3(:,2))
xlabel("Samples");
ylabel('Volts');
title("Time Domain Noise Signal");
```

GitHub

Throughout this project, to keep the all the code in a centralized location that was accessible by all as we had a member that was working remotely during the second semester while participating in a Co-Op. To accomplish this we used GitHub, attached is our [link to our GitHub] that includes all our code that was used to complete the project.

Appendix 3 – PCB Designs

Output Amplifier

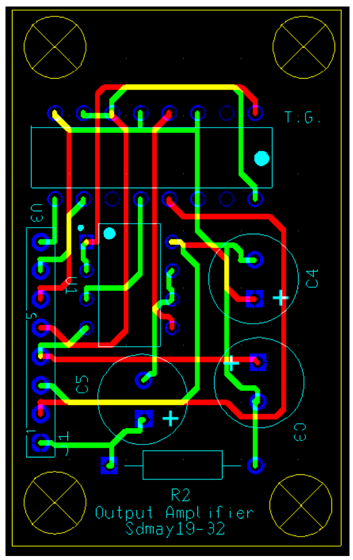


Figure 16

Mixer/White Noise

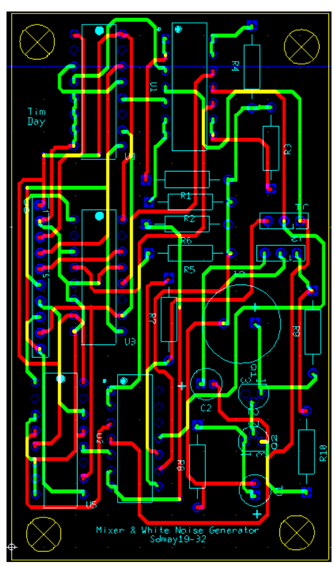


Figure 17

Oscillator

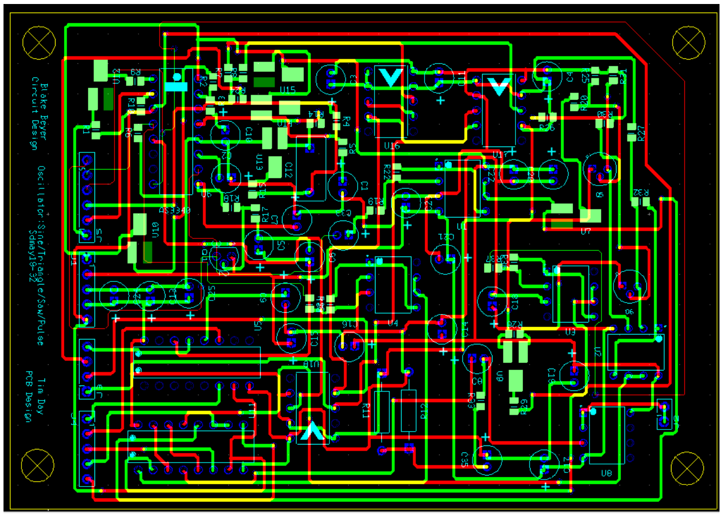


Figure 18

Mixed Signal Modbay

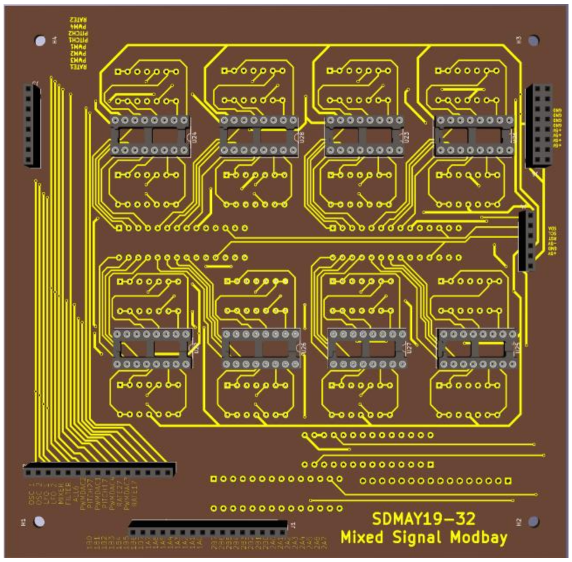


Figure 19

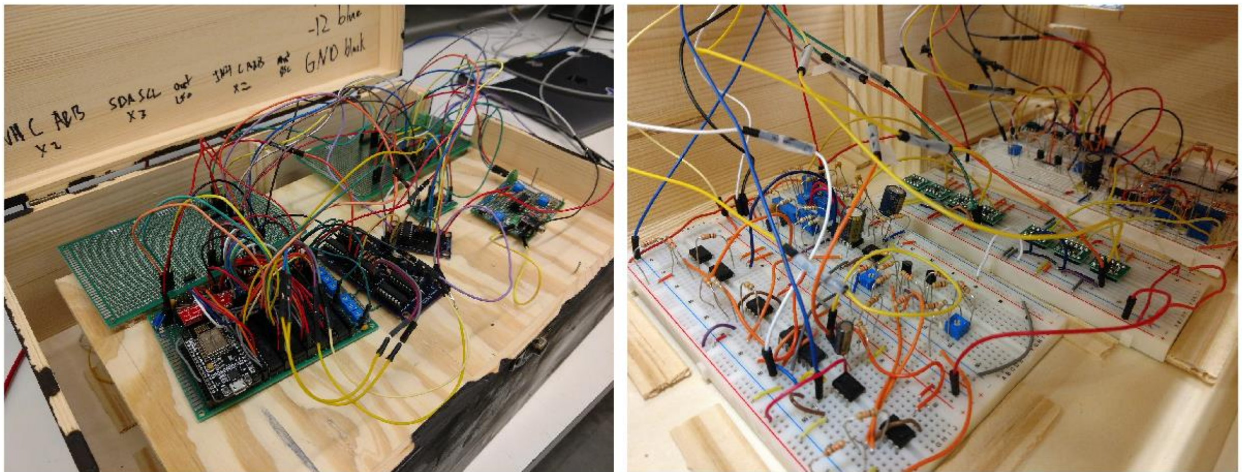
Appendix 3 – Addition Parts

Microcontroller

The Arduino was used for testing the modules individually. The node mcu is used in the final product as the wifi interface and microcontroller.

Housing

The housing for our project was a 16.6'' x 10.9'' x 6'' wooden box that contain two shelves. The top self holds the modbay, Wi-Fi interface, mixer, and power amplifier PCB's and the power rails. There is an input for the power to connect in the side of the box to the rails and a location for the output. The bottom self holds the oscillator breadboards since the PCB had mistaken from an incorrect schematic.



Top Shelf **Bottom Shelf**

Figure 20