PSUEDO-RANDOMLY CONTROLLED ANALOG SYNTHESIZER

By

Jared Huntington

Senior Project

ELECTRICAL ENGINEERING DEPARTMENT

California Polytechnic State University

San Luis Obispo

2009

TABLE OF CONTENTS

ILLUSTRATION INDEX

INDEX OF TABLES

Abstract

The goal of this project is to design and build a portable box recreating classic sounds heard in science fiction films (See Appendix C for project schedule). These include sounds ranging from droning hums to bloops and bleeps. Currently no other device is made specifically for generating these sounds. The user controls and manipulates the sounds produced using a combination of switches and knobs to shape the sound generation. The sounds are generated using analog electronics. The project was successful in meeting the desired goal by designing and building a Randomizer, MIDI controller, VCO, LFO, VCA, Ring Modulator, and Fuzz Section. The resulting device is useful for generating a unique atmosphere of sounds shaped by the user.

Introduction

The box generating these sounds needs to be easy for a musician to carry from gig to gig. The ability to transport the device is extremely important to act as a live sound instrument. The device has a sturdy enclosure and standard musical inputs and outputs located out of the way of the user. The project enclosure and user interface recreates the feel of an electronic device from the 1950's Atomic Era. Large flashing lights are used as well as large knobs and bulky switches similar to 1950's radio or lab equipment. The current states of different subsystems are visible through flashing LEDs allowing visual user feedback. The user inputs provide tactile response giving the user a rugged interface to work with without having to worry about breaking the device. The interface is extremely simple allowing even someone inexperienced in making music to manipulate the sounds.

Design Requirements

The device takes in synchronization signals through a MIDI connector (See Appendix A for project specifications). Using MIDI sync signals allows the box to synchronize with external devices such as a laptop, drum machine or keyboard expanding the musical possibilities. MIDI signals allow the device to control notes or events on external devices. MIDI input allows the device to trigger from an external source. MIDI input and output allows the user to record and replay sequences. The ability to record MIDI sequences allows for repeatable performances. The MIDI input, output, and sync all connect through a MIDI DIN 5 standard MIDI connector. The device outputs audio signals at standard audio line level through a ¼" jack allowing use of an audio amplifier. The user powers the device using either a 9V battery or a standard 9V barrel power adapter.

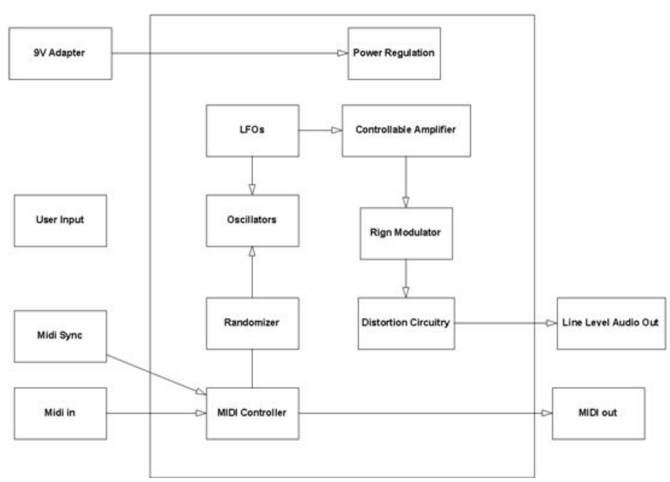

Figure 1: Project Block Diagram

Figure 1 shows the overall system connections. The sound synthesis and manipulation uses a voltage controlled oscillator, a randomizer, a MIDI controller, a controllable amplifier, a low frequency oscillator (LFO), a ring modulator, and a distortion circuit. The oscillator's center randomization frequencies are controlled using a center frequency knob. The center frequency knob controls if the sounds generated are centered on a lower pitch or centered on a higher pitch. The user selects the wave shape of each oscillator using a switch to select either a triangle or a square oscillator output. The user controls the portamento, or transition time between notes in the frequency domain, through a knob for each oscillator. A pseudo-random control voltage outputted by the randomizer controls the oscillator frequency. A master knob controls the tempo of the pseudo random note changes. If the system receives a MIDI sync signal the tempo knob is overridden. The randomizer provides a trigger causing the controllable amplifier to step through the amplifier's volume envelope. The user controls the attack and decay of the controllable envelope through a single knob for each individual oscillator. The system's LFO ties to the voltage-controlled oscillator's input of the first oscillator. The LFO is then used to frequency modulate the oscillator producing ringing sounds or warbles in frequency. The user switches between these settings using a selector switch. A knob controls the low frequency oscillators frequency and amplitude. A knob controls the ring modulator's mix between modulated and non-modulated signals.. The ring modulator only modulates between oscillator channels two and three. A knob controls the amount of overdrive in the fuzz section. The mixed oscillator signals are run through the fuzz section acting as a master distortion. The components in Figure 1 function together to synthesize the desired sounds allowing the user full control over the sounds generated.

Project Impact

Economic

Musicians often seek unique and different ways to create sounds often turning to boutique or limited release effects and sound generators. The Drum Buddy, a boutique instrument with unique aesthetic and tonality, sells for thousands of dollars. Currently no instruments are made for creating randomized bed noises so this fills a currently open niche in the boutique instruments and effects market. The project is made from easily obtainable low cost components (See Appendix B) and manufactured easily in a home electronics shop making it well suited as a limited run boutique instrument.

Environmental

Cradle-to-cradle design lessens the environmental impact of the project [2]. The cradle-to-cradle philosophy specifies a device should go back to soil safely or go into building other machines as efficiently as possible. The hardware enclosing the project and any wiring between components separate quickly and easily for reuse. RoHS compliant electronic components lower the amount of hazardous substances in the project. Local materials are used when possible to lower the amount of waste coming from shipping over long distances.

The Makers Bill of Rights [1] is another important design philosophy used throughout the project. The Maker's Bill of Rights states that the device should be easily repaired, disassembled or modified without requiring any special tools or information. This facilitates an easier time repairing, disassembling or re-purposing the device once it has reached the end of its life cycle rather than throwing the device away.

Sustainability

Cradle to Cradle states that sustainability is a minimum requirement and not an end goal[2]. The materials used in the project improve with each cycle. All of the materials used are made of recycled components or components that recycle as easily as possible. Any paints used on the project are easily removed to avoid contamination of the recycling process.

Manufacturability

The project is easily manufactured at a small home electronics workbench using a limited tool-set. The only tools needed to manufacture the device are a hand-held drill, a screwdriver, an Allen wrench, and a soldering iron. These simple tools make it so no outsourcing of parts is necessary. This keeps the manufacturing costs low for this project and makes assembly of the project a much quicker process. The circuit boards were etched using the iron on toner mask method [3] that can easily be done with a limited amount of supplies. This allows the project to be manufactured without outsourcing for a costly small circuit board run. The limited tool-set used to manufacture the project makes disassembly or modification at the end of its life-cycle much easier. The only tools needed to disassemble the project should be a screwdriver, Allen wrench, and diagonal cutters.

Ethical

The parts and materials used to build the project come from factories and supply chains that are as ethical as possible. This includes companies paying workers living wages and giving their employees ethical working conditions. The companies are ethical in regards to where the materials used to manufacture their products come from as well as the company's impact on the environment. Decision-making is based on this information wherever possible.

Health and safety

The health and safety of the user is very important. The project is a low voltage device and as such should be extremely safe for its user. The finished enclosure has no sharp edges or possible places for user injuries and is safe enough for a child to use. The safety of the person assembling and disassembling the project is very important. Lead free solder and RoHS compliant components prevent human exposure to toxic chemicals. No glues are used to put together the project lowering off gassing of fumes and lowing the toxins used. The project was soldered in an area lowering exposure to fumes. These measures make the manufacturing and use of this device as safe as possible.

Social

Music is a very important form of self-expression. The project encourages inexperienced people to make music because anyone can shape the sounds just by twisting knobs. By encouraging this self-expression from even non-musicians this allows people to come together and experiment with making sounds together. This improves the quality of life and helps people realize that it is possible for anyone to make art as a form of relaxation.

Political

Politics is the process by which a group makes a decision. The project required interaction between the Cal Poly Electrical Engineering Department, the project faculty advisor, and the the student working on the project. The Electrical Engineering has a structure and process for working on the senior project laid out in the Senior Project Handbook. The interaction between the faculty advisor was laid out on the faculty advisors web page. The need for politics was at a minimum because these processes were in place.

Design and Test

The design and testing of the system is organized with the system control sections first, signal generation sections second, and the signal manipulation sections last. Figure 1 illustrates the connections of the projects different subsections. Each subsection is broken down showing the current design first with a description of the circuit operation. The section then characterizes the section operation and states improvements and possible additional features to each subsection.

Randomizer

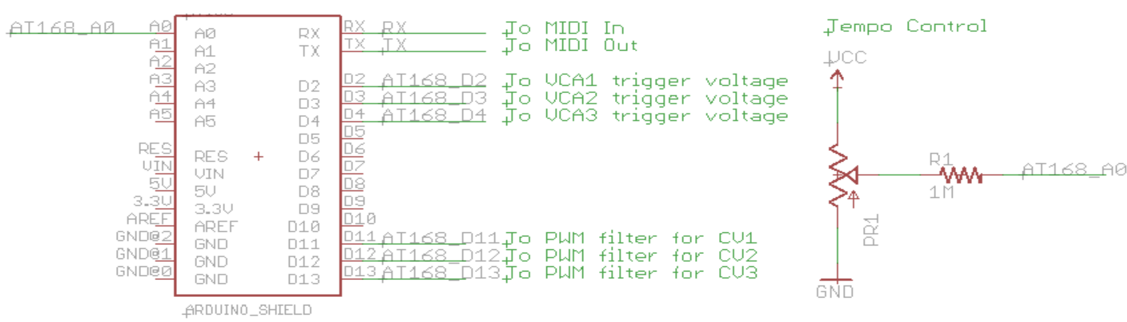

Figure 2: Atmega168 inputs and outputs

The randomizer requires 3 different 5V outputs to randomly trigger the controllable amplifier to gate the oscillators. Figure 2 shows an Atmega168 used to provide the random values for triggering the controllable amplifier (see Appendix D for micro-controller source code). The Arduino is a USB programming board for the Atmega 168 micro controller. The Arduino provides sufficient inputs and outputs to provide randomization and MIDI inputs and outputs using a single micro controller. The *random(int max)* function is used with a randomizing range of 2 making the only possibilities for output 0 or 1 for the controllable amplifier triggers. The output on pins 11,12 and 13 are either high or low depending on the random output. The randomization of trigger voltages creates a sequence that does not repeat from the users perspective.

A 1MΩ potentiometer provides user input setting the tempo of the random number changes. The potentiometer acts as a voltage divider on the analog input of the Arduino. The atmega168 analog pin provides an analog to digital converter. ADC output has a 10bit range allowing a potentiometer range from 0 to 1024. The input range is scaled providing the desired interval delay of 50ms to 30000ms. Multiplying the input by 30 and adding 50ms gives a range of 50ms to 30770ms.

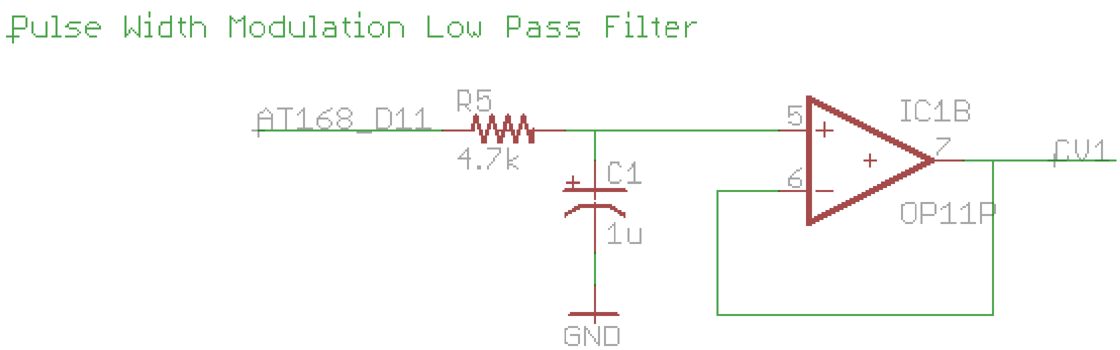

Figure 3: Pulse width modulation input through a low pass filter providing a DC control voltage

Pulse width modulation (PWM) provides a control voltage to the VCO. PWM minimizes the system's part count because an digital to analog conversion uses a simple low pass filter. The Atmega168 provides built in PWM on 6 different pins. Figure 3 shows the low pass filter used to transform from the PWM signal to a control voltage for the VCO. A passive LPF filters out any harmonics leaving only a DC signal suitable for a control voltage. A -3dB point at 30Hz filters the Atmega168s PWM frequency of 460Hz.

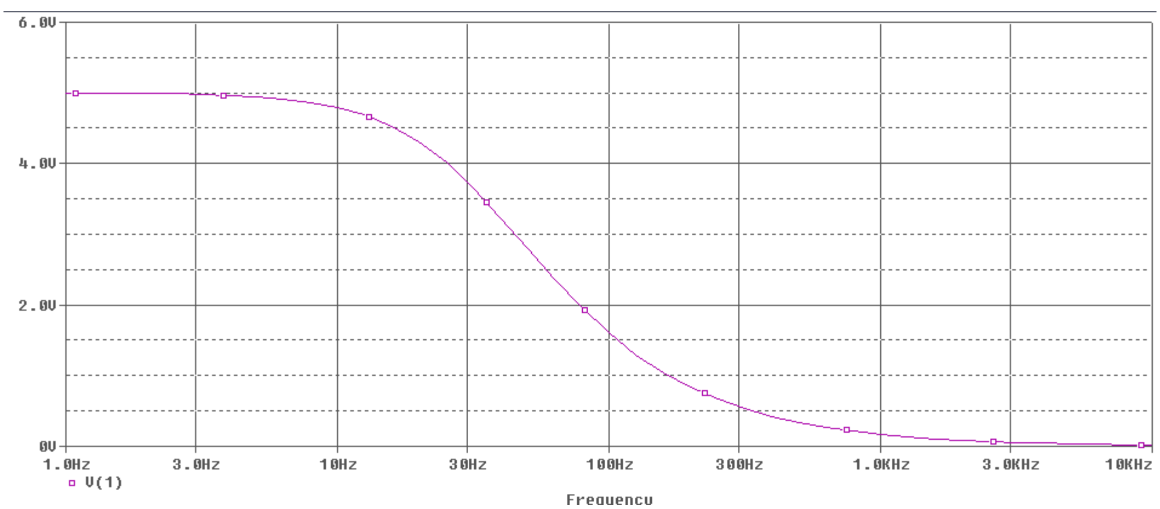

Figure 4: PSPICE simulation of PWM low pass filter providing a DC control voltage. V(1) is the transfer characteristic of the low pass filter with a 5V input at varying frequencies.

Figure 4 shows the low pass filter PSPICE simulation used to turn the PWM signal to a control voltage. The simulated output at the PWM frequency of 460Hz is –22.68dB. The 460Hz cutoff frequency filters out a large portion of the PWM signal leaving a steady control voltage.

Table 1: Pulse width modulation low pass filter spice simulation

```
* Jared Huntington
* PWM low pass filter
********

Vin 2 0 AC 5
R1 2 1 4.7k
C1 1 0 1u

.ac DEC 10 1 10k
.probe
.end
```

Table 1 shows the SPICE simulation code for the pulse width modulation low pass filter.

PWM software value	Output Voltage
0	0V
64	1.22V
127	2.43V
191	3.65V
255	4.88V

Table 2: Software Value Output Voltage

Table 2 shows the output voltages measured using a multi-meter to verify the desired output range. A PWM software value between 0 to 255 is written to the output pin on the Atmega168. Appendix D contains the Atmega168 source code used for this project.

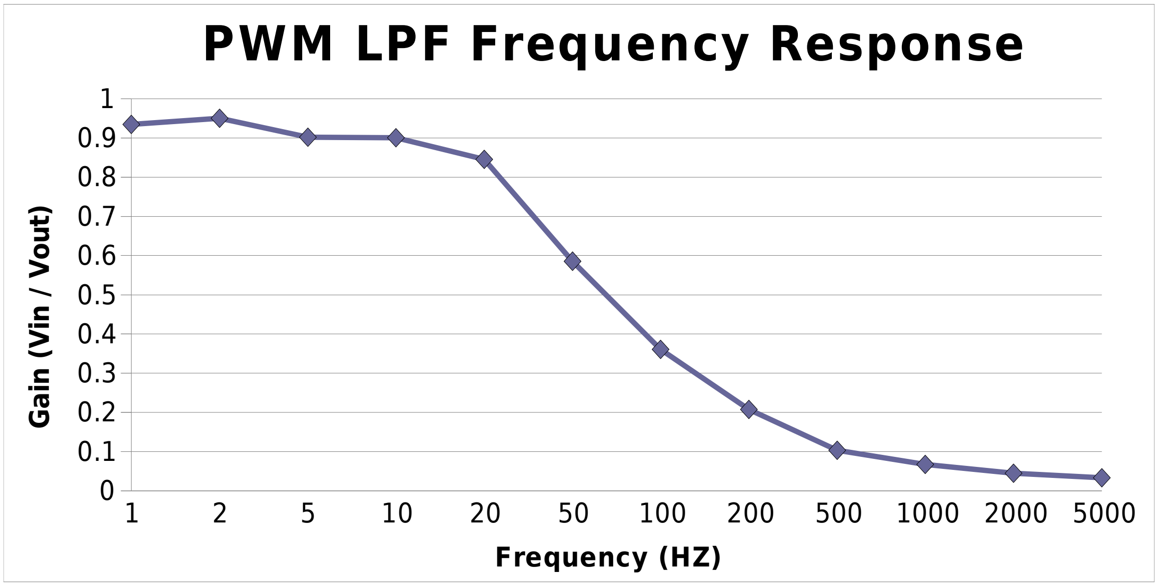

Figure 5: Measured PWM transfer characteristic

Figure 5 shows the measured pulse width modulation filter characteristics matching the simulated filter characteristics. An issue with using a PWM generated control voltage is that the filter does not create an entirely steady DC voltage. The filter lets through some PWM ripple. The ripple is audible on the output of the voltage controlled oscillator because even a few millivolt change translates to an audible change in pitch. Switching from a PWM based design to a resistive ladder digital to analog converter would remove the ripple issue. Changing to the resistive ladder design increases the projects part count but would provide an overall improvement in the precision of the control voltages.

MIDI Controller

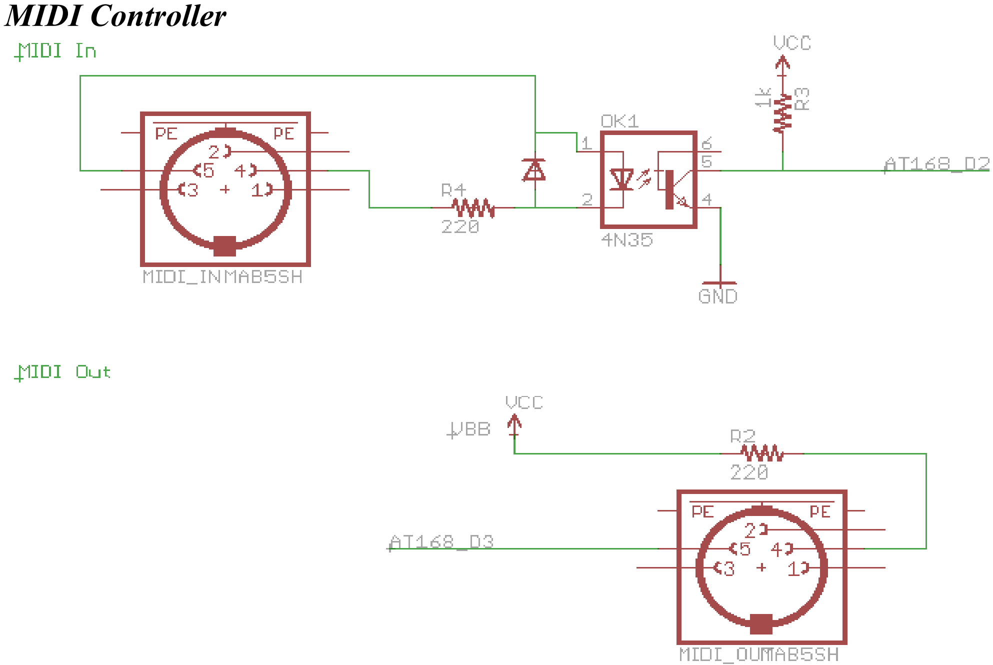

Figure 6: Atmega168 MIDI hardware interface

The MIDI in and out cables connect to the MIDI circuitry through a DIN5 connector. The MIDI protocol communicates serially through a current loop at a rate of 31250 baud. Figure 6 shows the MIDI input isolated through a 4N35 opto-coupler OK1 and connects to the pins 4 and 5 of the DIN5 connector. Diode D1 protects the opto-coupler's LED from reverse voltages across the input. The MIDI data after the opto-coupler has a 5-volt logic high value and a 2.5-volt logic low value. The input to the Atmega168 does not trigger a low data value at 2.5-volts so a CMOS buffer shifts the logic low value to a level that triggers the Atmega168. The MIDI output only requires a current limiting resistor.

The Atmega168 MIDI software loops waiting for MIDI data to come in on the serial data stack. The first nibble of the MIDI note sent on 0x9 states that the MIDI note on command has been sent. The second nibble is the MIDI channel triggered by the MIDI on command. 0x90 triggers MIDI channel 1. 0x91 triggers MIDI channel 2. MIDI channels start counting at channel 1. The MIDI note on command for channels one through three turns on the trigger voltages to the voltage-controlled amplifier. The MIDI note off commands 0x80 – 0x82 turn off the different voltage controlled amplifier trigger voltages. The MIDI clock command 0xF8 is sent 96 times per measure. When the MIDI section receives a clock command the software switches from the user input tempo to waiting for MIDI clock commands. The randomizer waits for 96 clock signals to randomize the VCO control voltages output and the VCA trigger voltage. The MIDI start command 0xFA is sent at the start of a MIDI stream. The start command resets the counter within the Atmega168. The start command keeps the MIDI counter synced with the MIDI controller source when a sequence has been reset.

The MIDI section meets all of the specified requirements and needs no improvements.

Voltage Controlled Oscillator

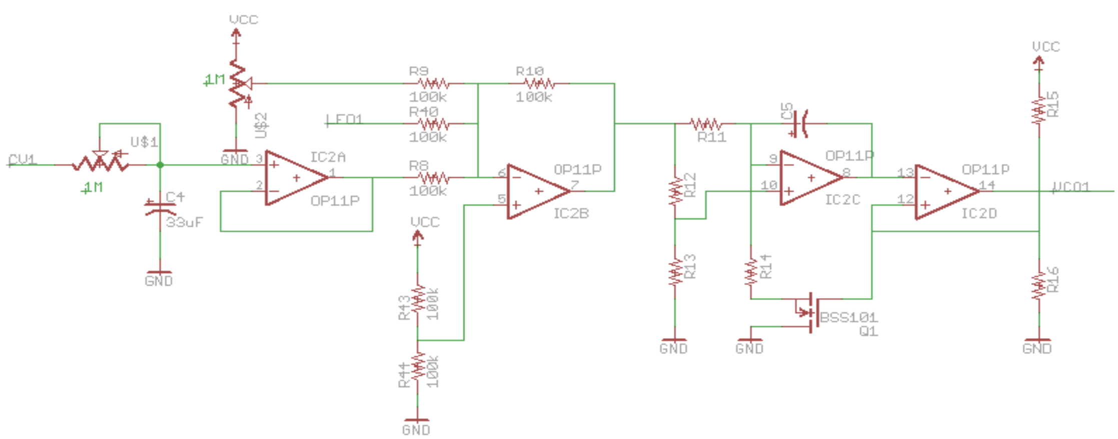

Figure 7: Voltage Controlled Oscillator Schematic

Figure 7 shows the system's portmanteau circuit. The low pass filter does not allow the control voltage to change quickly and instead causes the control voltage to slowly transition. The 1MΩ potentiometer U$1 adjusts the RC constant between U$1 and C4. A 30 second maximum RC constant is desired. The following equations calculate the portmanteau resistor and capacitor values.

$T = RC$

$30sec = 1M\Omega * C$

$C = 30uF$

A 33uF capacitor exceeds the necessary time constant. Opamp OP11P is configured as a voltage follower providing a high impedance barrier to the portamento. The control voltage from the output of the portmanteau circuit is summed with the output from the low frequency oscillator and a tuning resistor U$2. U$2 eliminates the voltage controlled oscillators voltage offset providing a linear response. The summed control voltage is fed to the input of the voltage-controlled oscillator. The oscillator consists of a current to voltage converter followed by a Schmitt trigger. The voltage controlled oscillator generates a square and a triangle wave[5].

Calculating the R and C values for the voltage controlled oscillator:
Fo = kvi

Vol ~~= 0V
Voh = Vcc / (1 + R2 / R1)
= 9v / (1 + 5k/10k)
= 6V

Fo = V1 / 8RC(VTH – Vtc)
Foh = 3/16 1/RC
R = 10kΩ
C = 37.5nF

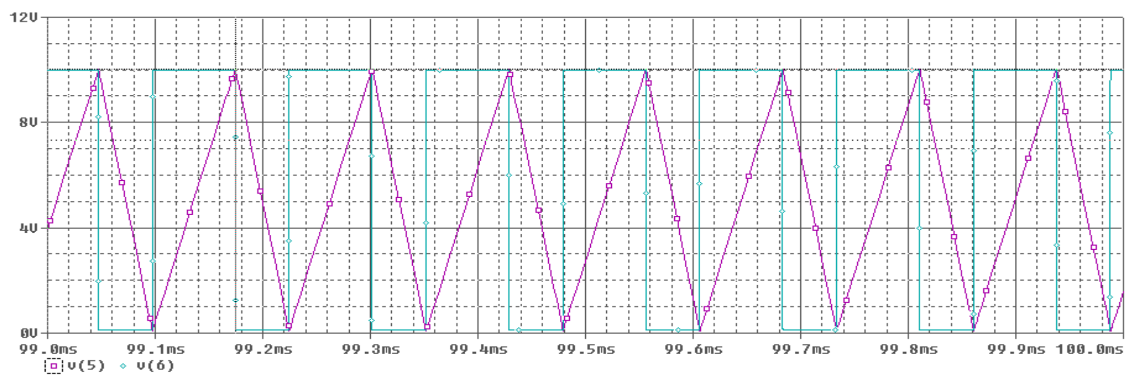

Figure 8: Voltage Controlled Oscillator PSPICE simulation. V(6) is the square wave output of the voltage controlled oscillator. V(5) the triangle output of the voltage controlled oscillator.

Figure 8 shows the voltage-controlled oscillator PSPICE simulation output.

Table 3: Voltage Controlled Oscillator SPICE Code

```
*Jared Huntington
*VCO
*June 17 2009
*****************
vi  1  0 9
vss 10 0 9

r1  1  2 10k
r2  2  0 10k
r3  3  1 20k
r4  3  4 10k
r5  10 6 3.3k
r6  6  0 6.6k
c1  3 5 1.8n

x1  2 3 10 0 5    uA741
x2  6 5 10 0 6    uA741
m1  0 6 4 0      nmosfet
.MODEL nmosfet NMOS(W=4u L=1u kp=50u VTO=1 lambda=0.01)
.MODEL CMOSN NMOS LEVEL=3 PHI=0.600000 TOX=2.1200E-08 XJ=0.200000U
+TPG=1 VTO=0.7860 DELTA=6.9670E-01 LD=1.6470E-07 KP=9.6379E-05
+UO=591.7 THETA=8.1220E-02 RSH=8.5450E+01 GAMMA=0.5863
+NSUB=2.7470E+16 NFS=1.98E+12 VMAX=1.7330E+05 ETA=4.3680E-02
+KAPPA=1.3960E-01 CGDO=4.0241E-10 CGSO=4.0241E-10
+CGBO=3.6144E-10 CJ=3.8541E-04 MJ=1.1854 CJSW=1.3940E-10
+MJSW=0.125195 PB=0.800000
.tran .1u 10m 0 10u
.probe
.end
```

Table 3 shows the PSPICE code used to simulate the VCO. The VCO successfully simulated a frequency range of 62Hz to 3907Hz using a single rail supply. The NJM2059 is used as a comparator and a voltage to current converter. The specified range is met using the calculated values but the oscillator sounds better using a lower frequency range.

R = 5kΩ
C = 20nF
Fmin = 16Hz
Fmax = 2500Hz

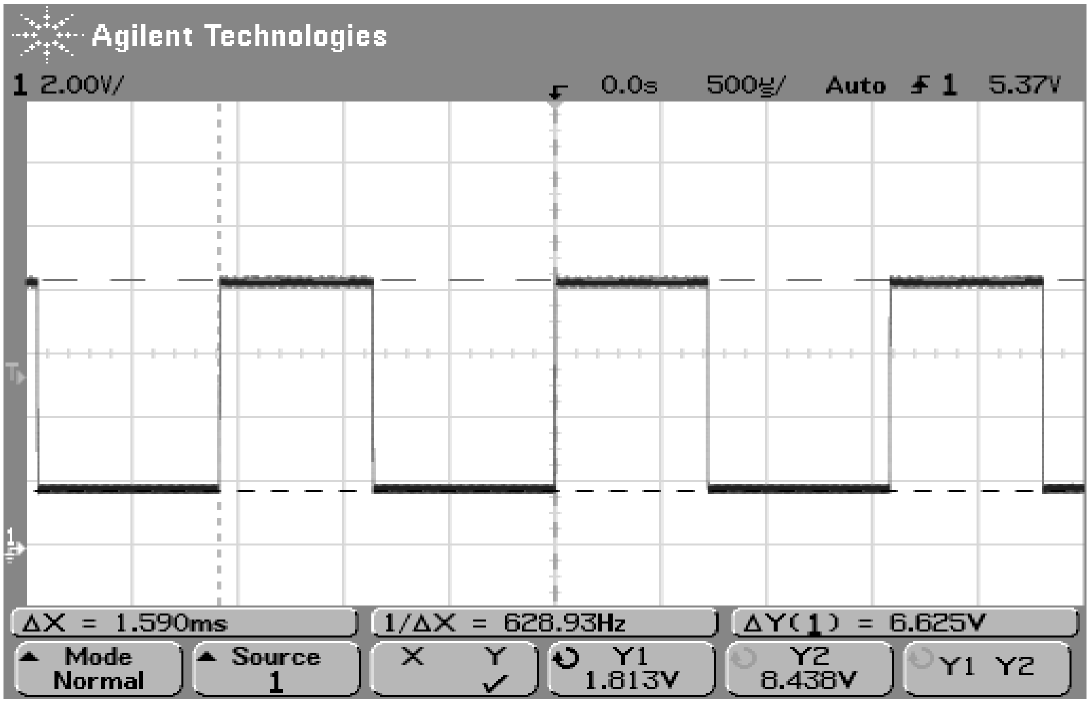

Figure 9: VCO Square Wave Output

Figure 9 shows voltage controlled oscillators square wave output. The square wave output results from the triangle oscillators triangle wave being run through the Schmitt trigger output stage.

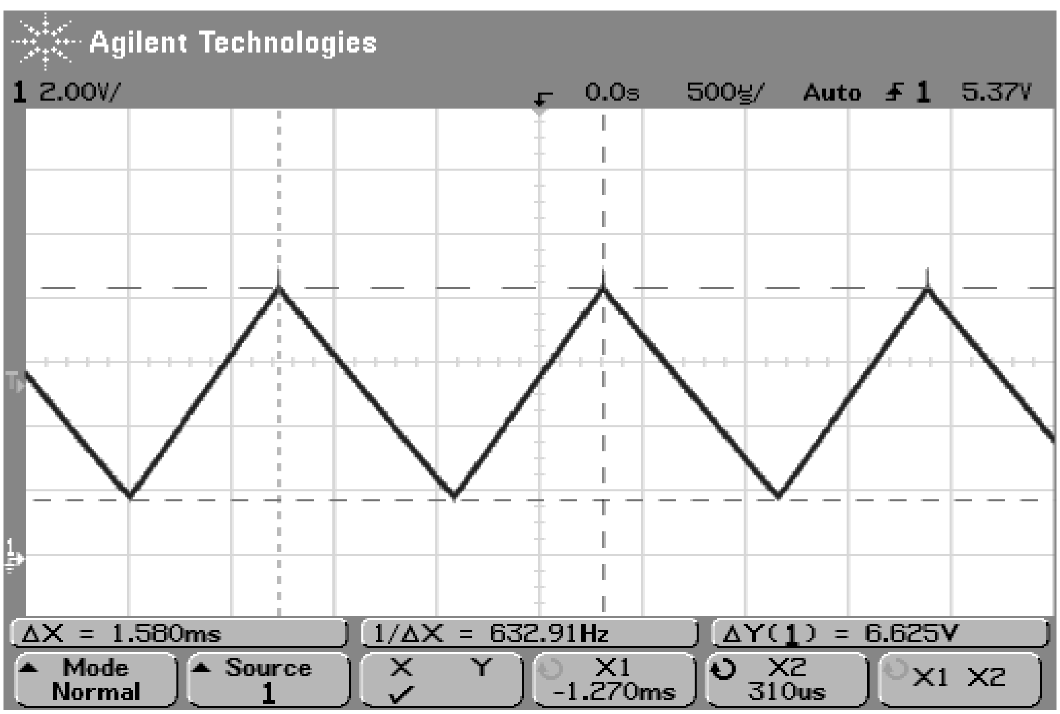

Figure 10: VCO Triangle Wave Output

Figure 10 shows the voltage controlled oscillators triangle wave output.

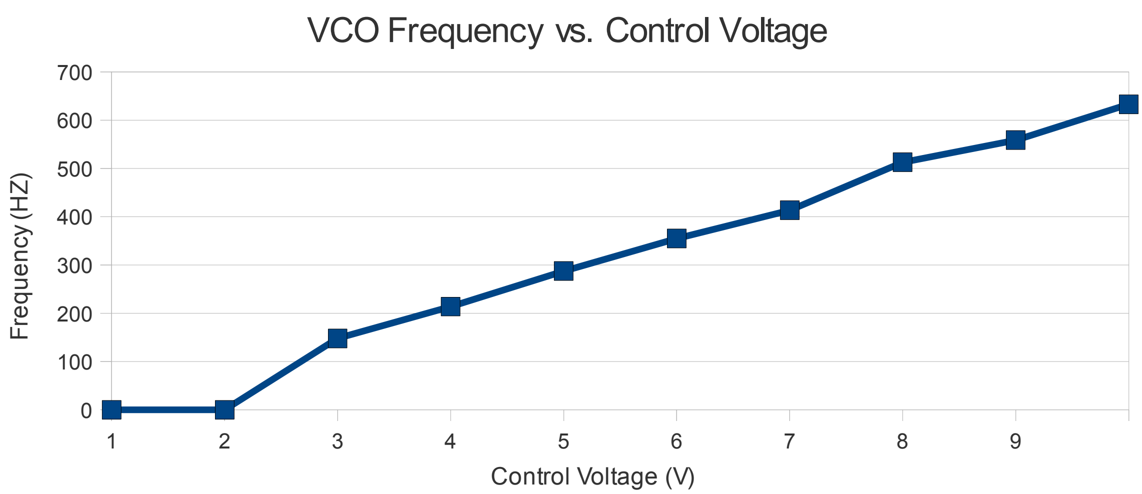

Figure 11: Frequency response of voltage-controlled oscillator to control voltage input

The voltage-controlled oscillator provides a linear control voltage after an initial turn on voltage. Figure 11 shows the voltage controlled oscillator's output frequency from the control voltage. The actual tuning range of the VCO is much smaller than the simulated tuning range. Switching the system from a single rail design to a dual rail design could improve the tuning range. A control voltage of 0V would be well above the bottom operational amplifier rail improving the VCO linearity. Including a ramp and sinusoidal wave-shaping section would increase the harmonic possibilities and improve the VCO.

Low Frequency Oscillator

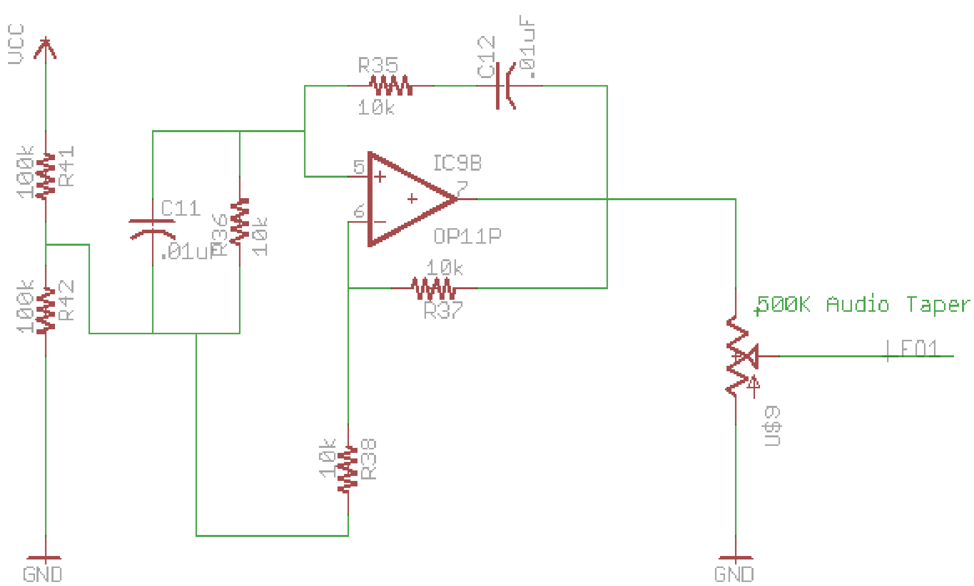

Figure 12: Low Frequency Oscillator Schematic

Figure 12 shows a simple wien-bridge oscillator used the low frequency oscillator [5]. R41 and R42 are necessary to provide a virtual ground for the operational amplifier. A 500k log taper potentiometer adjusts the depth of the LFO. The R36 and R35 need to be a dual potentiometer allowing the user to adjust the frequency of the LFO.

Table 4: Low Frequency Oscillator SPICE code

```
*Jared Huntington
*VCO
*August 10 2009
****************


*Wien-Bridge Oscillator:
Cp 3  8  1nf IC=0V
Rp 3  8  158k
Cs 3  36 1nF IC=0V
Rs 36 6  158k
R1 2  8  10k
R2 2  6  22.1k
R3 2  26 100k
R4 7  8  100k
R5 8  0  100k
XOA 3 2 7 0 6 uA741
VCC 7 0 dc 9

.tran 50us 15ms 0ms 50us
.probe
.end
```

Table 4 shows the SPICE code used to simulate the low frequency oscillator.

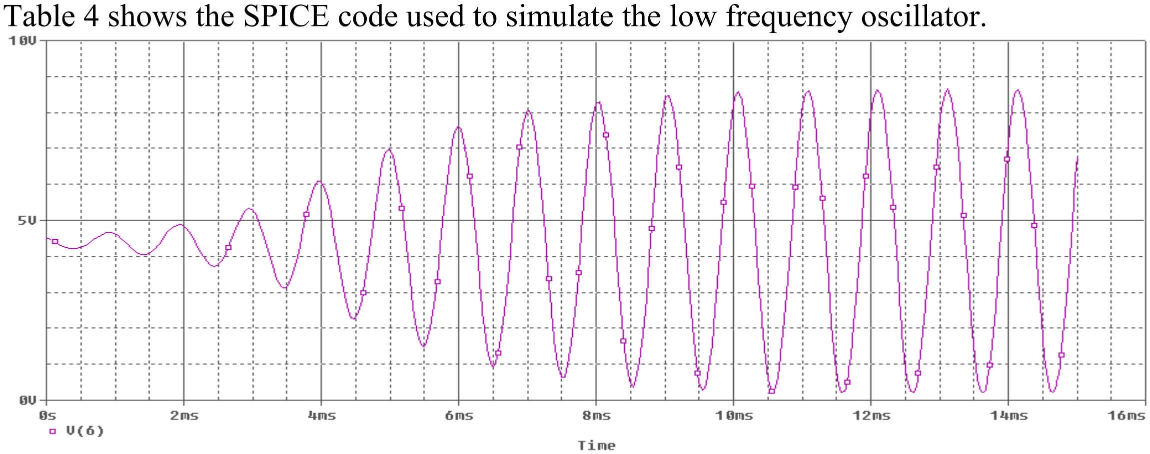

Figure 13: Low Frequency Oscillator PSPICE Simulation. V(6) is the output of the LFO.

Figure 13 shows the low frequency oscillator's SPICE simulation results.

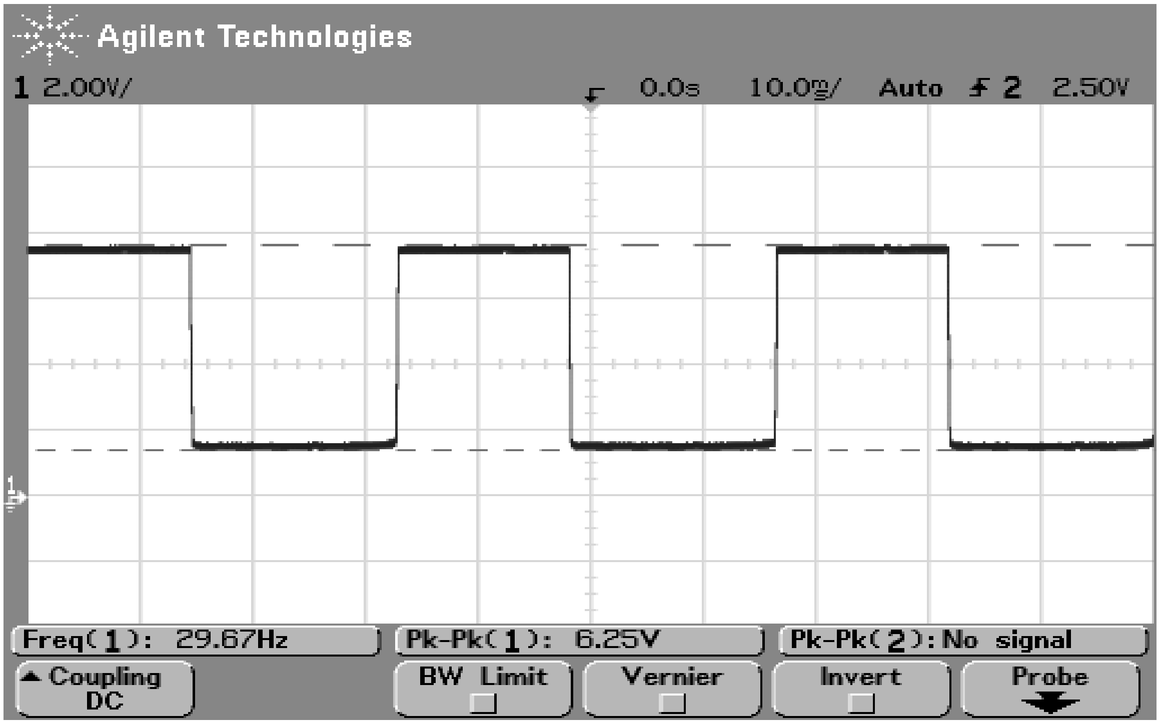

Figure 14: Low frequency oscillator

Figure 14 shows the SPICE simulation of the low frequency oscillator output. Figure 13 shows the actual low frequency oscillator output. The actual low frequency oscillator output matches the simulated response of the low frequency oscillator. R41 and R42 could be removed from the circuit with a dual rail design simplifying the low frequency oscillator. Changing the design to include more waveform shapes would improve the sonic possibilities of the low frequency oscillator.

Voltage Controlled Amplifier

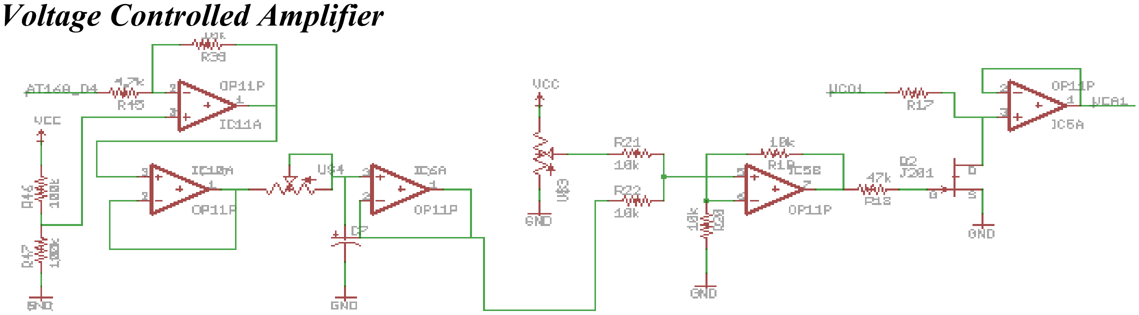

Figure 15: Voltage Controlled Amplifier Schematic

Figure 15 shows the voltage controlled amplifier's trigger output from the Atmega168 fed into a non-inverting amplifier. The amplifier shifts the trigger voltage from 5 volts to 9 volts. The trigger is fed to the adjustable RC combination of C7 and U$4 with voltage followers on either side providing high impedance buffers. High impedance buffers prevent the circuitry on either side from being affected. The RC circuit allows the user to adjust the potentiometer U$4 to change the RC time constant. The RC time constant acts

Table 5: Voltage Controlled Amplifier SPICE simulation code

```
*Jared Huntington
*VCA
*August 10 2009
****************

.model pjfet PJF ( Beta=500u Betatce=-.5 Rd=1 Rs=1
Lambda=10m Vto=-3
+             Vtotc=-2.5m Cgd=2.5p M=.3333 Pb=1 Fc=.5
Cgs=4p Is=90p
+             Isr=800p N=1 Nr=2 Xti=3)
.model njfet NJF (IS=1N VTO=-4 BETA=0.5M
+ Lambda=2.40E-3 CGD=5.85PF CGD=3.49PF)

vin  1  0 9v
vc   7  0 9v
vpp  5  0 9v
vnn  0  6 9v
r1   1  3 100k
r2   7  2 47k
x1   3 4 5 0 4    uA741
j1   3 2 0     pjfet
* analysis
.dc vc 0v 9v .1v
* results
.plot    dc       V(1) V(7) V(4)
.probe
.end
```

as an attack and release volume envelope. The output from the attack and release volume envelope is fed into a non-inverting summing amplifier along with a voltage dividing trimmer potentiometer. The user can adjust the trimmer potentiometer to further linearize the JFET response. After the summing amplifier the attack and decay voltage is fed into the J201 JFET. The VCA is based on a JFET operating in the linear range acting as a voltage controlled resistor. A p-channel JFET allows for a positive control voltage. Table 5 shows the SPICE simulation testing the voltage controlled amplifier design.

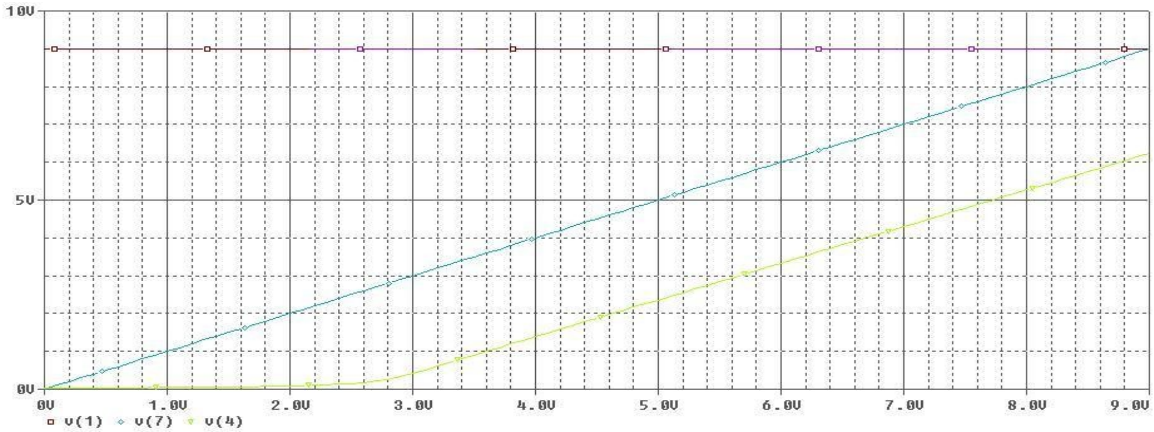

Figure 16: Voltage Controlled Amplifier PSPICE Simulation. V(1) the input to the VCA. v(7) is the control voltage varying linearly from 0 to 9V. V(4) is the VCA's output voltage.

Figure 16 shows the voltage-controlled amplifier simulated in PSPICE with the input tied to 9V and the control voltage swept from 0 to 9 volts. A non-linear period occurs when the control voltage is below 2.5 volts. A tuning resistor eliminates the non-linearity by summing with the control voltage. Switching to a dual rail design could improve the voltage-controlled amplifier. A single rail design would decrease the required voltage divider resistors and improve the power consumption of the circuit.

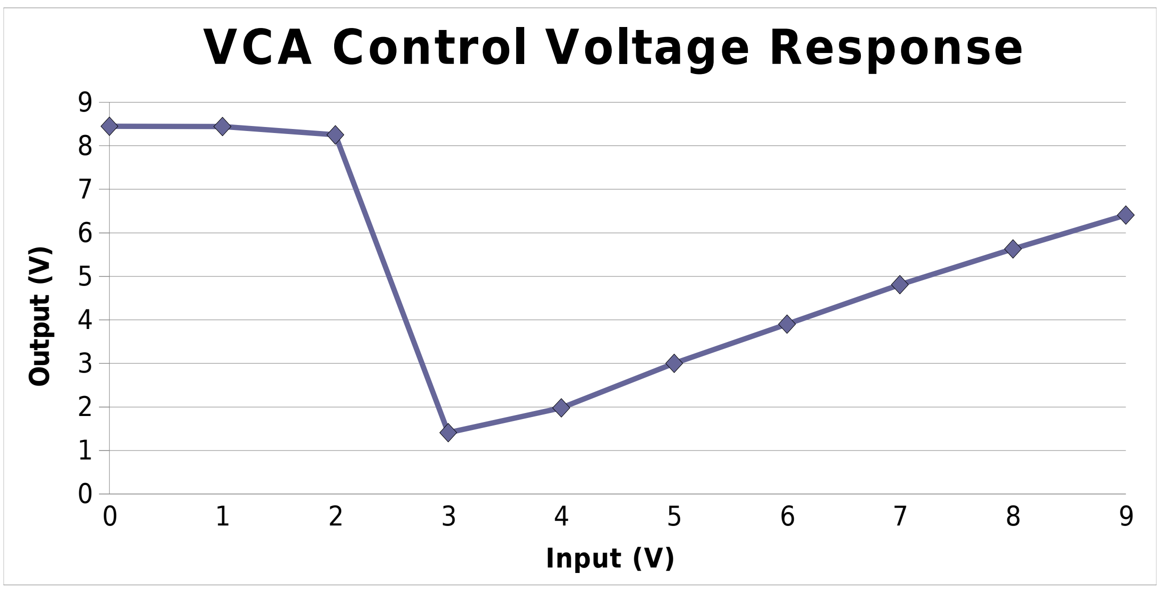

Figure 17: Voltage Controlled Amplifier Measured response

Figure 17 shows the voltage controlled amplifiers measured response matching the expected simulation values once the control voltage reached the linear region. The simulation area between 0 and 2 volts resulted in a 0 volt output. The P-channel JFET is off and has not reached its linear region so the simulation did not match the results. The biasing of the JFET needs to be corrected to linearize the control voltage response and fix the non-linearity.

Ring Modulator

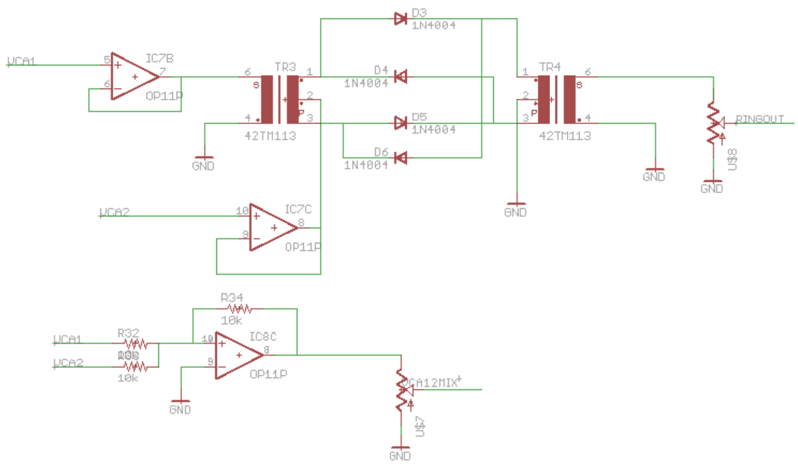

Figure 18: Ring Modulator Schematic

The ring modulator in Figure 18 modulates the signals from channel 1 and channel 2. These signals are mixed using IC8C to output un-modulated signals. The modulated and non-modulated signals are mixed using a dual potentiometer U$7 and U$8. The connections from U$7 and U$8 are connected inversely so the fully modulated signal is allowed through completely the un-modulated signal is almost eliminated from the mix. This allows the user to use the dual potentiometer to adjust the amount of ring modulation between channel 1 and channel 2. The mixer from channels 1 and channel 2 use 4.7k and 10kΩ feedback resistors to cut the voltages in half to avoid clipping. Two 100kΩ resistors create a virtual ground from the single supply voltage.

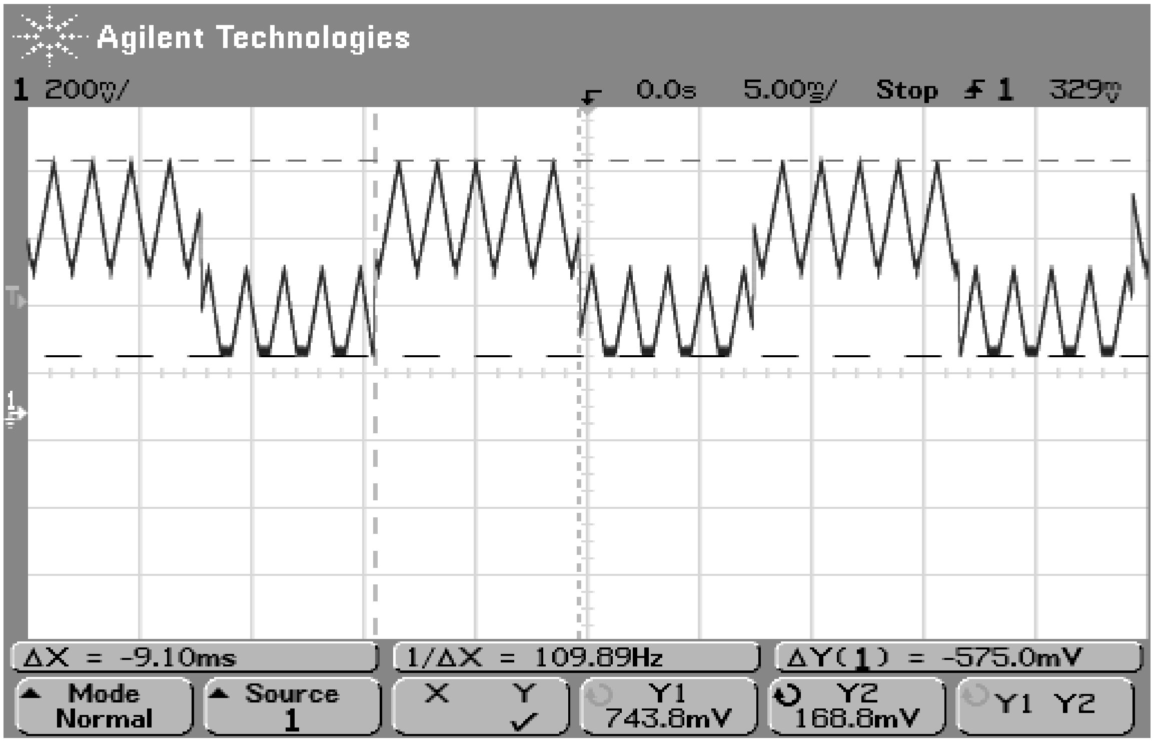

Figure 19: Mixed inputs from VCA1 and VCA2

Figure 19 shows the outputs of VCA 1 and VCA 2 mixed. Two signals are mixed with equal attenuation.

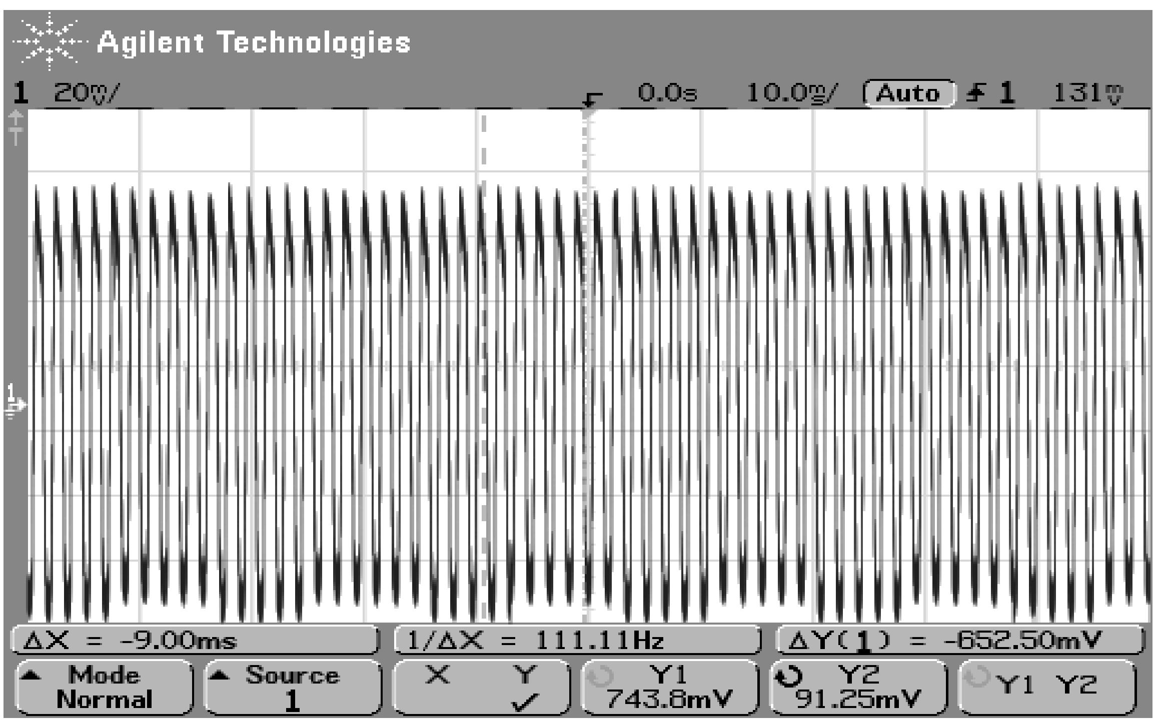

Figure 20: Ring Modulated inputs from VCA1 and VCA2

Figure 20 shows the ring modulator section successfully multiplying the two input signals. The ring modulator mixer has a low impedance input. Increasing the impedance would increase the amount of modulation between the two input signals.

Fuzz Circuit

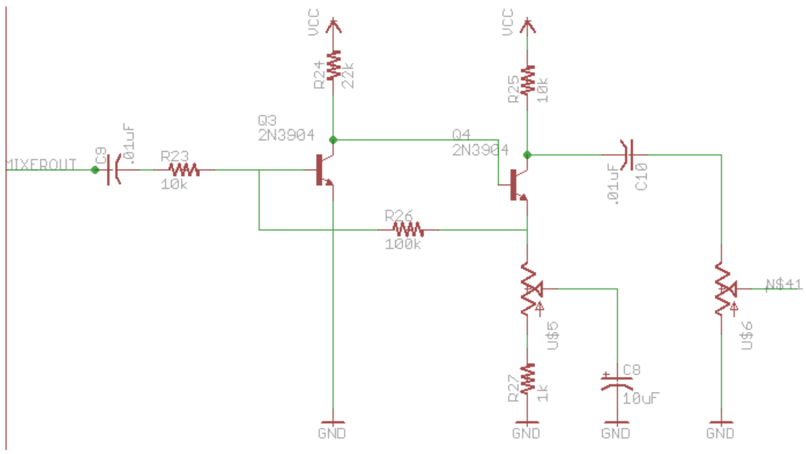

Figure 21: Fuzz Circuitry and Output Volume Schematic

The fuzz circuit in Figure 21 is based on the classic Fuzz Face circuit [4]. C9 removes the DC offset of the final mixer centering the signal on ground. The first transistor Q3 amplifies the input signal to clip the input of the second transistor Q4. The potentiometer U$5 allows the user to adjust the clipping of transistor Q4. C10 acts as an AC bypass removing the Q3 and Q4 DC biasing. The potentiometer U$6 allows the user to adjust the final output volume.

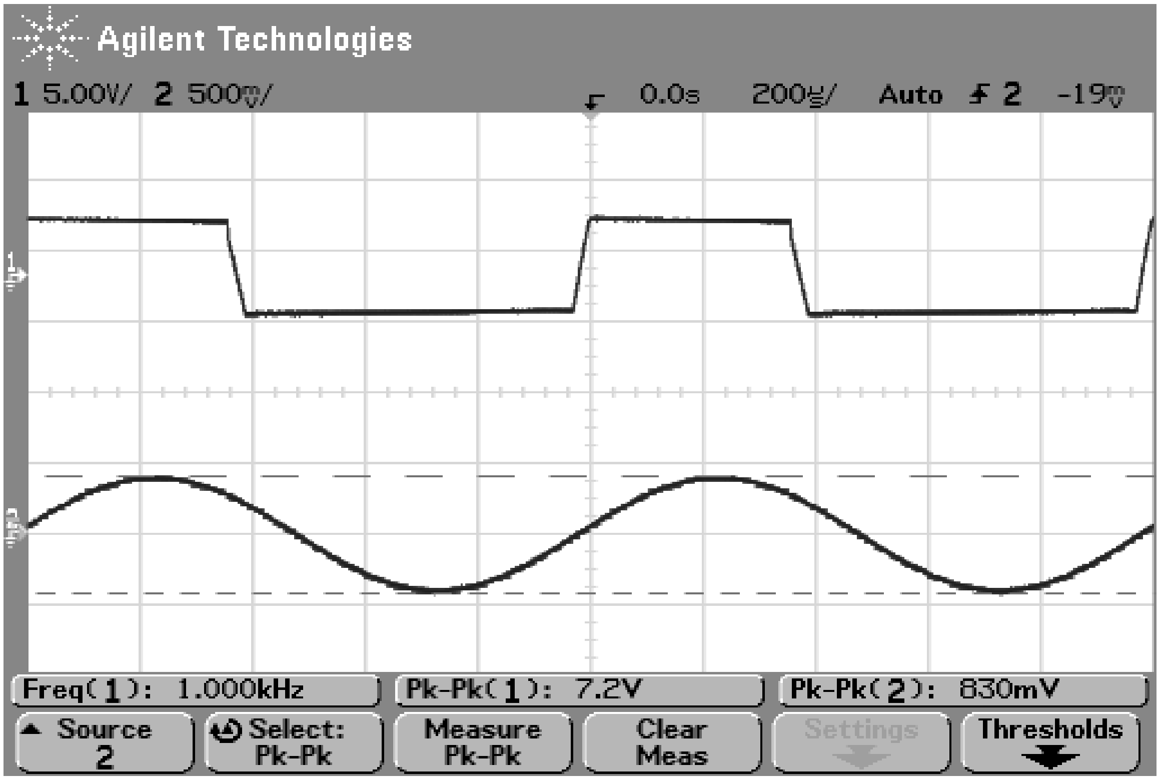

Figure 22: Fuzz circuit final output

Figure 22 shows the fuzz section clipping a sinusoidal input. Switching from the fuzz face circuit to an op-amp based distortion circuit could reduce the part count of the synthesizer. Adding a filter to the fuzz section would improve the design.

Power Supply

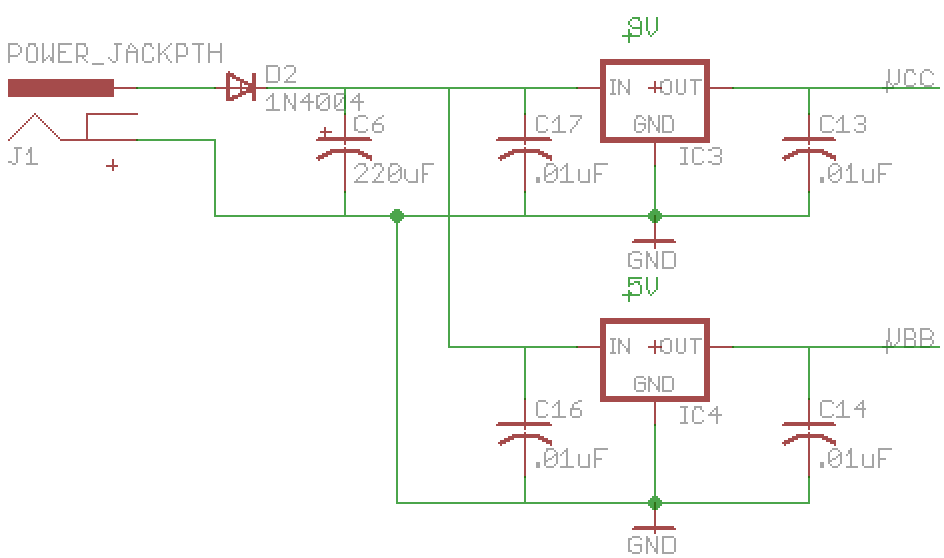

Figure 23: Power Supply Schematic

The power supply circuit in Figure 23 is based around the LM75XX family of power chips. The LM75XX provide 1A of current and +/- 5% voltage stability. The capacitor C6 220µF polarized capacitor acts to reduce the amount of ripple voltage coming from the DC wall wart power supply. The LM7509 power IC provides power for the VCO, VCA, Mixers, LFO, and Distortion Circuitry. The LM7505 provides the power to the Randomizer and the MIDI interfacing circuitry. The .01µF capacitors C13, C14, C16, C17 on the inputs and the outputs of the LM75XX act to decouple noise from the power supply. The power supply meets all requirements and does not need any improvements to the design.

Lessons Learned

Using a dual rail design instead of a single rail design would improve the design. A single rail design requires less external biasing resistors which reduces the part count and lowers the power consumption of the final design. Implementing the design on a breadboard is a time consuming process. It would be better to test different subsections and ideas on a breadboard while implementing the different system components on a printed circuit board. A PCB would reduce the amount of time spent debugging loose wires. The time spent debugging wires could be spent improving the overall system design. Further tests on the way the specifications sound would help the system's sound. A broader definition of the projects technical specifications would have allowed for more flexibility in the design. The resulting design works and sounds good. There are many improvements that could be made to the design.

References

1. Mister Jalopy, "The Makers Bill of Rights," Makezine.com, Original Source MAKE Magazine Vol. 4 Pg. 154, Date of internet publication unknown. Available: http://makezine.com/04/ownyourown/. [Accessed: 12/1/2009].
2. W. McDonough, M. Braungart , Cradle to Cradle, 1st Edition. Place of publication: North Point Press, 2002.
3. A. Ricci, "How to make PCB's at home," Riccibitti.com, Date of internet publication unknown. . Available: http://www.riccibitti.com/pcb/pcb.htm. [Accessed: 12/1/2009].
4. J. Orman, "NPN Fuzzface" Muzique.com, Date of internet publication unknown. Available: http://www.muzique.com/schem/fuzzface3.gif. [Accessed: 12/1/2009]
5. S. Franco, Design With Operational Amplifiers And Analog Integrated Circuits, 3rd Edition, New York, New York: McGraw Hill, Year, 454 - 474.

Appendices

Appendix A: Specifications

Randomizer

- A maximum of 50ms to at least 30 seconds between pseudo-random note changes in the time domain
- Note density of once for every 4 measures to once every eight note
- Three 0-5V outputs for the VCOs control voltage input
- Three 5V discrete outputs to trigger the controllable amplifier envelope
- Operate using 9v and 0V rails

MIDI Controller

- Operate using 9v and ground rails
- Baud rate of 31250 baud for serial input and output
- `0x90` *Note-on* MIDI signal turn on 5V envelope trigger from randomizer to controllable amp
- `0x80` *Note-off* MIDI signal turn off 5V envelope trigger from randomizer to controllable amp
- *0xF8 Timing Clock* MIDI signal should override the user input master tempo
- MIDI input is optically isolated from input signal

Voltage Controlled Oscillator

- 50Hz-5kHz output frequency range
- Square and triangle waveforms
- Square wave form has a maximum rise time of 100us
- Maximum of1ms to at least 30s frequency domain slide between notes (portamento).
- Control voltage input from 0-9V
- Output 9V waveforms peak to peak
- Operate using 9v and 0V rails

Low Frequency Oscillator

- .1-50Hz frequency range
- Square output waveform
- 0-2V peak to peak output
- Operate using 9v and 0V rails
- Square wave form has a maximum rise time of 100us

Voltage Controlled Amplifier

- Attack and decay ranging from a maximum of 50ms to at least 30 seconds
- Envelope should be triggered by 5V input signal
- 0-9V input and output
- Operate using 9v and 0V rails
- 19dB to 1dB

Ring Modulator

- 0-9V peak to peak input
- A maximum allowable attenuation of –19dB

Fuzz Circuit

- 0-9V peak to peak input and output

- Operate using 9v and 0V rails
- At least 2V of clipping on input waveform

Power Supply

- 9V+/- .25V regulated power
- Minimum 1A of power.

Appendix B: Parts List and Costs

Estimated Costs:

Part	**Cost ($)**
Knobs	10
Hardware	5
Enclosure	20
Components	20
Circuit Boards	10
Total:	105

Actual Costs:

Part	**Cost ($)**
Knobs	0
Hardware	0
Enclosure	0
Components	40
Circuit Boards	10
Total:	50

Appendix C: Schedule and Time Estimates

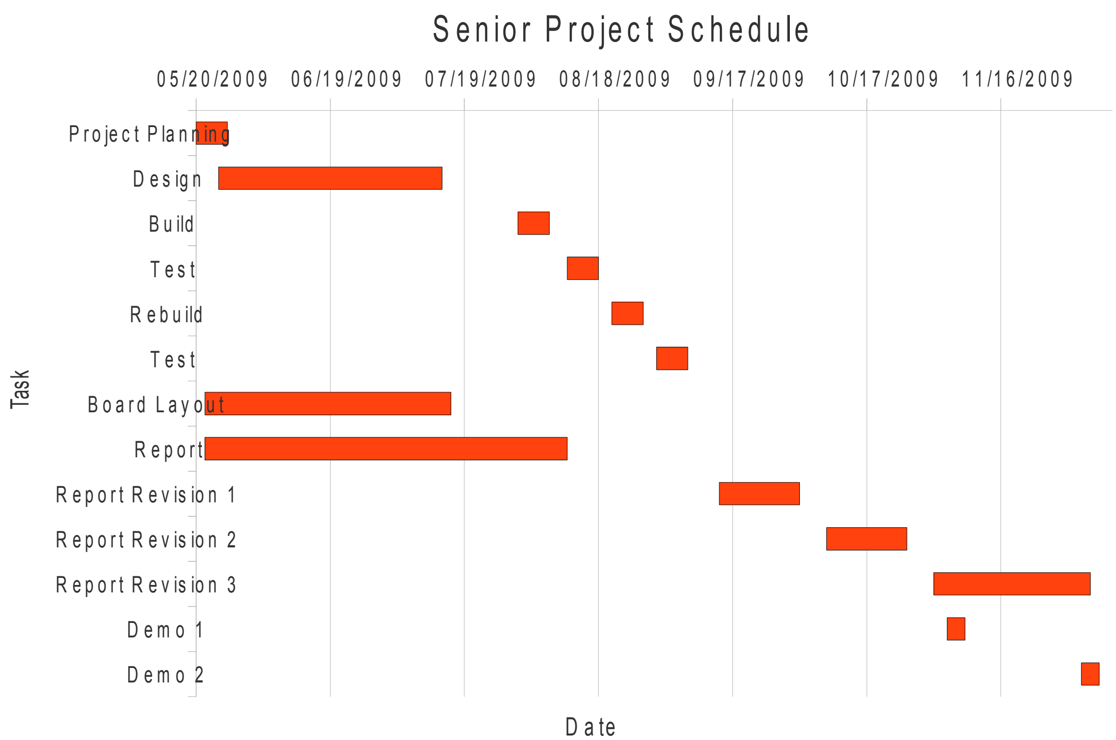

Figure 24: Project Gantt Chart

Appendix D: Atmega168 Source Code

```
/*
 * Jared Huntington October 18 2009
 * Atmega168 (Arduino) source code
 */
//input output pin definitions
int tempo_LED = 13;  // select the pin for the LED
int vco1 =  9;  // pins for vco CVs
int vco2 = 10;
int vco3 = 11;
int vca1 =  6;  // pins for vca triggers
int vca2 =  7;
int vca3 =  8;

int tempo_pin = 0;  //AtoD pin for tempo
//arduino constants
#define PWM_MAX 255
//flags
#define INTERN_CLK 0
#define EXTERN_CLK 1
#define INTERN_NOTES 0
#define EXTERN_NOTES 1
```

```
int clk_location = INTERN_CLK;
int notes_location = INTERN_NOTES;
int midi_clk_count = 0;
//MIDI constants
#define CHANNEL1 0x00
#define CHANNEL2 0x01
#define CHANNEL3 0x02
#define MIDI_CLK 0xF8
#define MIDI_START 0xFA
#define MIDI_STOP 0xFC
#define MIDI_ON  0x90
#define MIDI_OFF 0x80
#define MIDI_BAUD_RATE 31250
#define MIDI_CLK_PER_MEASURE 96
//setup: declaring inputs and outputs and initialize midi serial
void setup() {
  pinMode(tempo_LED,OUTPUT);   // declare the LED's pin as output
  pinMode(vca1, OUTPUT);      // sets the vca pins as output
  pinMode(vca2, OUTPUT);
  pinMode(vca3, OUTPUT);
  pinMode(vco1, OUTPUT);      // sets the vco pins as output
  pinMode(vco2, OUTPUT);
  pinMode(vco3, OUTPUT);
  Serial.begin(MIDI_BAUD_RATE); //initialize tx/rx to midi baud
}
//loop: wait for serial data, and interpret the message
void loop () {
  int idelay;  //variable to hod the delay read from tempopin
    if(Serial.available() > 0) //if serial stack has recieved anything
      process_midi();  //process midi
    if(clk_location == INTERN_CLK)  //internal based clk and gating
    {                               //until a midi clk is heard
      idelay = analogRead(tempo_pin);  //read tempopin
      randomize_notes();
      tempo_update();
      randomize_gating();
      delay(idelay*30+50);  //set the minimum tempo to .05sec and scale input to
    }                       //make a 30sec maximum
}
void process_midi()
{
  int input = Serial.read();  //read byte off the serial stack
  if((input & 0xF0) == MIDI_ON)  //wait for a midi on signal to set extern note
  {                              //dependency
    notes_location = EXTERN_NOTES;
    digitalWrite(vca1, LOW);     //set all gatings to zero once midi has been
    digitalWrite(vca2, LOW);     //switched to external note source
    digitalWrite(vca3, LOW);
  }
  if(notes_location == EXTERN_NOTES)  //only process when external notes
  {
    //Channel 1 Code
    if(input == (MIDI_ON | CHANNEL1))
      digitalWrite(vca1, HIGH);
```

```
    if(input == (MIDI_OFF | CHANNEL1))
      digitalWrite(vca1, LOW);
    //Channel 2 Code
    if(input == (MIDI_ON | CHANNEL2))
      digitalWrite(vca2, HIGH);
    if(input == (MIDI_OFF | CHANNEL2))
      digitalWrite(vca2, LOW);
    //Channel 3 Code
    if(input == (MIDI_ON | CHANNEL3))
      digitalWrite(vca3, HIGH);
    if(input == (MIDI_OFF | CHANNEL3))
      digitalWrite(vca3, LOW);
  }
  if(input == MIDI_START)  //reset midi counter when midi
    midi_clk_count = 0;    //start recieved. keeps in sync
  if(input == MIDI_CLK && notes_location == INTERN_NOTES)
  {
      midi_clk_count++;  //increase counter
      clk_location = EXTERN_CLK;  //switch to extern clock shutting off intern generation
      //randomize notes and gating once a midi measure
      //has been completed.  reset the midi counter.
      if(midi_clk_count == MIDI_CLK_PER_MEASURE)
      {
        midi_clk_count = 0;
        tempo_update();
        randomize_notes();
        randomize_gating();
      }
  }
}
//toggle tempo LED indicator on/off with each function call
int tempo_led_status = 0; //global status variable
void tempo_update(){
  tempo_led_status = ~tempo_led_status; //invert current value
  tempo_led_status ? digitalWrite(tempo_LED, HIGH): //write high or low
                     digitalWrite(tempo_LED, LOW);  //depending on the current value
}
//randomize the PWM output to generate a control voltage for VCO
void randomize_notes(){
  analogWrite(vco1,random(PWM_MAX); //generate random PWM value 0 - 255
  analogWrite(vco2,random(PWM_MAX)); //generate random PWM value 0 - 255
  analogWrite(vco3,random(PWM_MAX)); //generate random PWM value 0 - 255

}
//randomize the trigger voltages of channel 1, 2, and 3
void randomize_gating(){
  random(2) ? digitalWrite(vca1,HIGH) : digitalWrite(vca1,LOW); //randomize a high or low
value
  random(2) ? digitalWrite(vca2,HIGH) : digitalWrite(vca2,LOW); //randomize a high or low
value
  random(2) ? digitalWrite(vca3,HIGH) : digitalWrite(vca3,LOW); //randomize a high or low
value
}
```