ECE Fundamentals II

Fall 2017 Final Project: PCB Audio Visualizer

November 2017 - - December 13, 2017

Andrew Powell - awp6rd@virginia.edu

John Hilten - jph5fs@virginia.edu

Eddie Ryan - ejr8wh@virginia.edu

Pierce Faraone - pf4tj@virginia.edu

Individual Efforts:

While most of our efforts were collaborative, some of the project stages were completed by individual group members. Andrew and Pierce led the analytical calculations on Multisim to ensure that the peak detectors and filters worked correctly. Pierce, Jonathan, and Andrew made sure that the Multisim verifications translated correctly to the breadboard format. Further experimental verification was done by Pierce, Andrew, and Eddie to ensure that no mistakes would be made during soldering. Soldering was done by Andrew, Pierce, and Eddie. Overall, this group understood how to divvy up the project work and ensure that the work was done in a timely manner.

Background Information and Rationale:

In picking the song for our signal, we tried a variety of different options. At first we tried a fast paced EDM song but quickly realized the song was too chaotic to use. We then went through a variety of different songs until we decided to pick one in which there were clear bass notes as well as clear high frequency signals. Our resulting choice ended up being the intro to Common's album "Be". We chose this song because it starts off with a bassline from a stringed bass instrument, and then adds synthesizer and piano. All these elements allowed us to develop breakpoint frequencies that would result in the best output on our board.

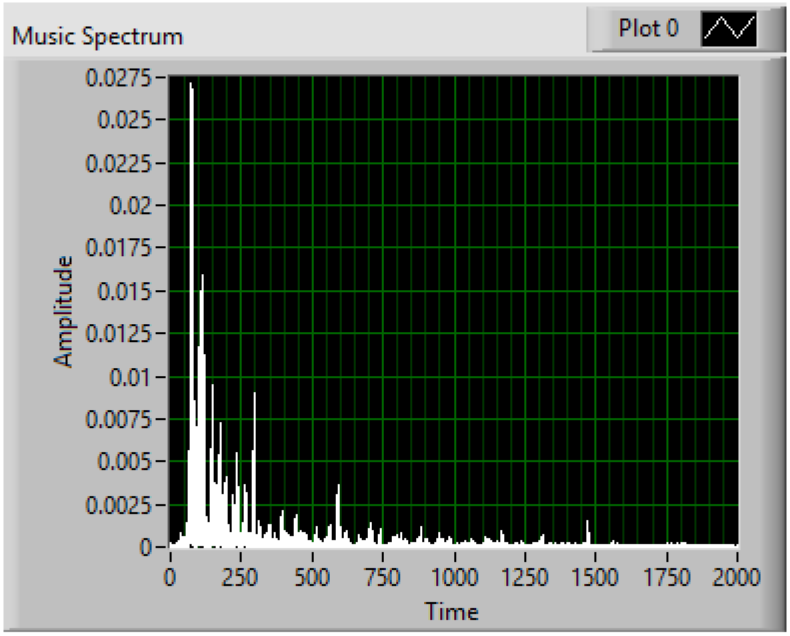


Figure 1 - Signal Spectrum Analysis

The figure above shows the spectrum of our selection from the provided LabView Sound Processor file. From this spectrum, we decided to set our low pass breakpoint to about 100 Hz and our high pass breakpoint to about 300 Hz. We chose these values because the spectrum shows significant peaks at the frequencies. Looking back it may have made more sense to set the breakpoints at frequencies closer together because then certain frequencies would not be completely left out. However we made the decision not to do this because we wanted our LEDs to be somewhat distinct and not have them both be on for similar frequencies.

Filter Calculations:

We chose the parts for both our low pass and high pass filter using spreadsheet 3.2 from the data sheet, “Analysis of the Sallen--Key Architecture”.

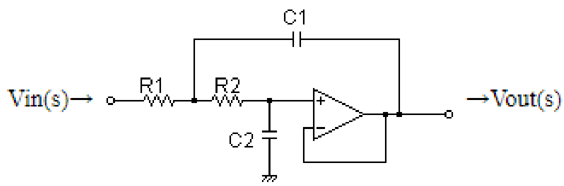


Figure 2 - Low Pass Sallen-Key Filter

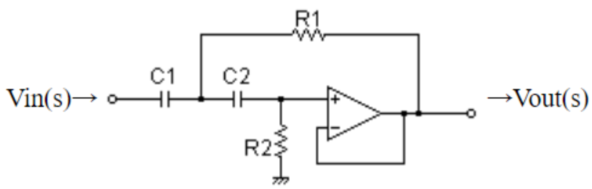


Figure 3 - High Pass Sallen-Key Filter

For our filter designs, we used the simplification of gain of the OpAmp being 1. Then we used the equations below to determine our values.

Table 1 - Sallen-Key Filter Simplifications

R1 = m*R	R2 = R	C1 = C	C2 = n*C

$$Q = \frac{\sqrt{mn}}{m+1} \qquad f_c = \frac{1}{2\pi\sqrt{mn}RC}$$

For the low pass filter we did our best to achieve a breakpoint as close to 100 Hz as possible using values present in our lab kits. On top of getting the breakpoint frequency close to a certain value, we also had to be careful in our ratio selection as it would affect the Q point greatly. In order to get a Q as close to Butterworth damping, Q = 0.707, we made the m ratio equal to 1 and tried to get the n ratio as close to 2 as possible.

Table 2 - Low Pass Sallen-Key Results

R1 = 47kΩ	R2 = 47kΩ	C1 = 0.022μF	C2 = 0.047μF

Q = 0.731 $\qquad f_c$ = 105.31 Hz

From the table, the values calculated got us very close to our desired Butterworth Q value and breakpoint frequency. The Q value we got is slightly less damped then Butterworth damping. The figure below shows a resultant Bode plot based on our values.

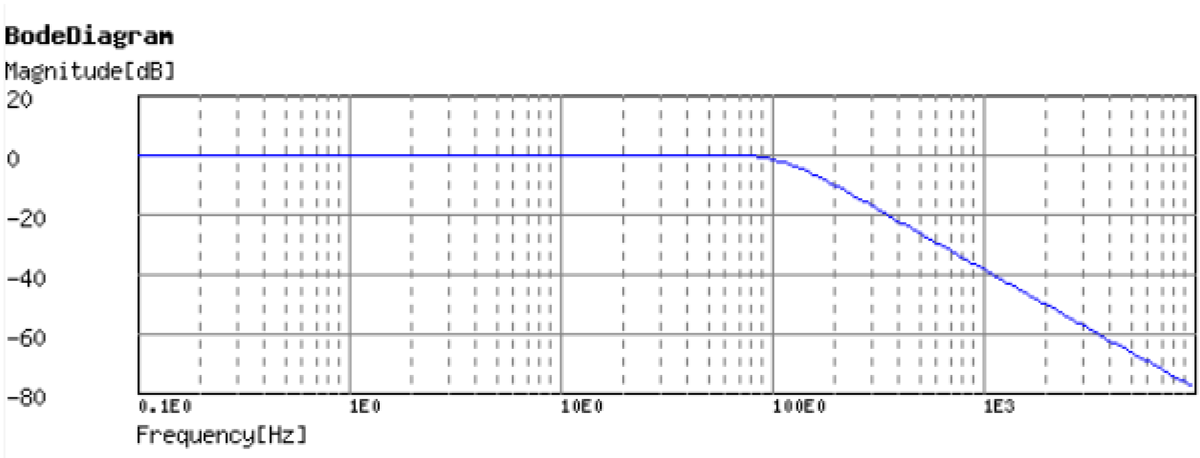


Figure 4 - Low Pass Sallen-Key Bode Plot

For the high pass filter we used a very similar approach to that of the low pass. We set our m to 1, making our resistors equal, and tried to get n to be as close to 2 as possible. We chose this approach because we determined it to be the simplest way to achieve Butterworth damping. Our calculated high pass values are in the table below.

Table 3 - High Pass Sallen-Key Results

R1 = 18kΩ	R2 = 18kΩ	C1 = 0.022μF	C2 = 0.047μF
Q = 0.731		f_c = 274.97 Hz	

We achieved the same damping as with the low pass filter and our breakpoint frequency was appropriate for what we wanted to achieve.

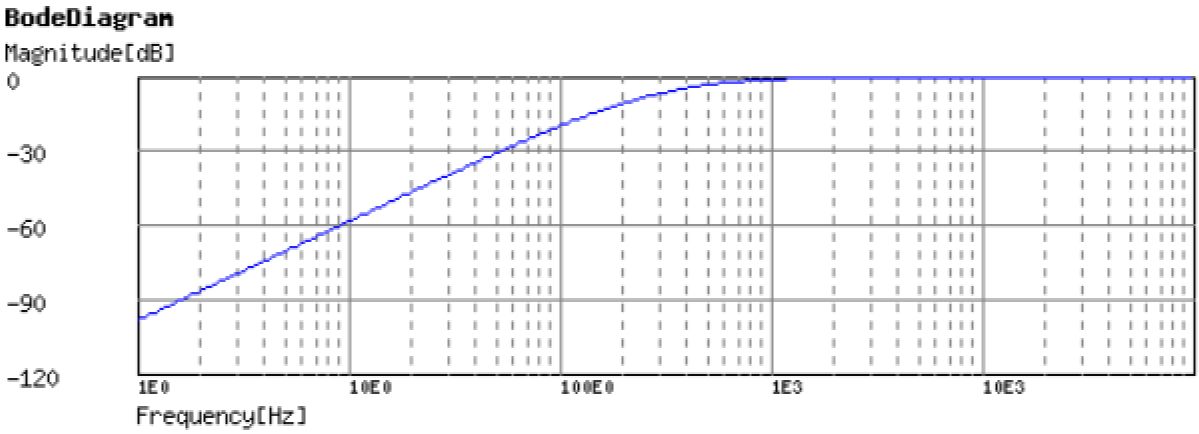


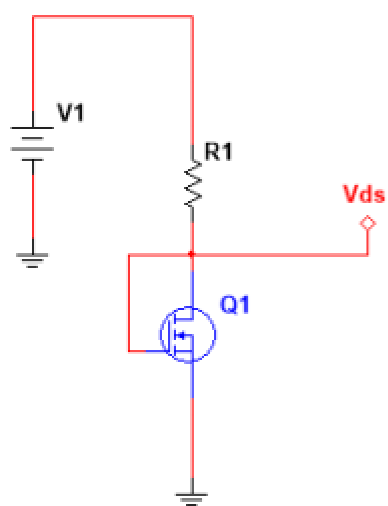
Figure 5 - High Pass Sallen-Key Bode Plot

Summing Amplifier Calculations:

For our summing amplifier design, we were looking for a gain of 1. In order to achieve this we set the resistors for both channel inputs and our feedback resistor to be equal. This means our input resistors R18 and R19 should equal the resistor R5 (placed across the inverting node and the output). R18, R19, and R5 were chosen to be 150kΩ because high resistor values will keep our current within an acceptable range. We chose fairly large capacitors before the summing node to ensure that the breakpoint frequency of that passive high pass filter was small enough that it only blocked any DC coming from the inputs.

Peak Detector Calculations:

The input of the peak detector op-amp was designed to have a DC offset equal to the measured threshold voltage of our ZVN3306 MOSFET. To experimentally determine the threshold voltages of our transistors, we built the circuit to the right. When R1 is very high resistance, Since Ids = Kn (Vgs − Vtn)^2 ≈ 0, then Vgs=VTN and Vgs=Vds. Measuring Vds therefore would give an accurate value for the threshold voltage, that was close to the MOSFETs' threshold voltage. (placed across the inverting node and the output).



Experimental Values:

Since the threshold values may vary among MOSFETs, our group experimentally determined our threshold voltage values to be 1.2 and 1.6. Using these values we determined what resistor values we would need to use in the voltage divider. Setting R14 to 270k ohms and R15 to 56k ohms gave us in a voltage of 1.54 at the positive terminal. At the other voltage divider, setting R16 to 220k ohms and R17 to 33k ohms gave us a voltage of just under 1.2 volts. This kept the incoming voltage just under the threshold voltage value of the MOSFET. The gain resistors (R18 and R19) were equal to keep the gain = 1. 150k resistors worked in simulation and in testing with the virtual bench. Another, trial and error determination was the size of the resistors R7 and R13 in the peak detector circuits. Too much resistance and the LED will not be bright enough. We settled on a value of 330 ohms and 560 ohms for R7 and R13, respectively.

P-Channel MOSFET:

P-channel MOSFETS are extremely useful electrical devices that can reduce power loss and avoid polarity mistakes to keep from damaging the circuit. A typical P-channel MOSFET turns on when the difference between the gate and the source Vgs is around -4 volts or more negative. In our circuit, the input voltage is positive, so the source voltage is positive. As mentioned above, the Vgs is negative and always greater than the positive threshold voltage. In this state the MOSFET is turned ON and VCC is very close to 9v (small voltage drop). If the battery was connected backwards, the voltage at the drain will be negative. If the MOSFET is turned on then the voltage at the source will be almost exactly the negative voltage at the drain. Therefore, Vgs is a positive voltage and the MOSFET is therefore not ON. What is the power loss in a circuit like this? Well, one parameter, the Rds (Static drain-source ON-resistance) describes the resistance between the drain and source specific to the MOSFET type. In this case, the power loss is given by I^2 * R. This is much, much less than the power loss if a diode was used instead. Of course, when designing a circuit some specific MOSFET characteristic must be taken into account. The MOSFET's drain-source voltage rating must be higher than the power supply voltage. Our P-channel MOSFET has a voltage rating of +/- 20 volts which is larger than our power supply. In addition, the Rds ON should be as little as possible to minimize power leakage. Our BSP170P MOSFET has approximately 300 mΩ . Maximum gate source voltage should have a large breakdown voltage range. Accounting for these necessary design specifications, a P-channel MOSFET can protect circuitry with little power loss.

Board Assembly

First, we built the circuit in Multisim

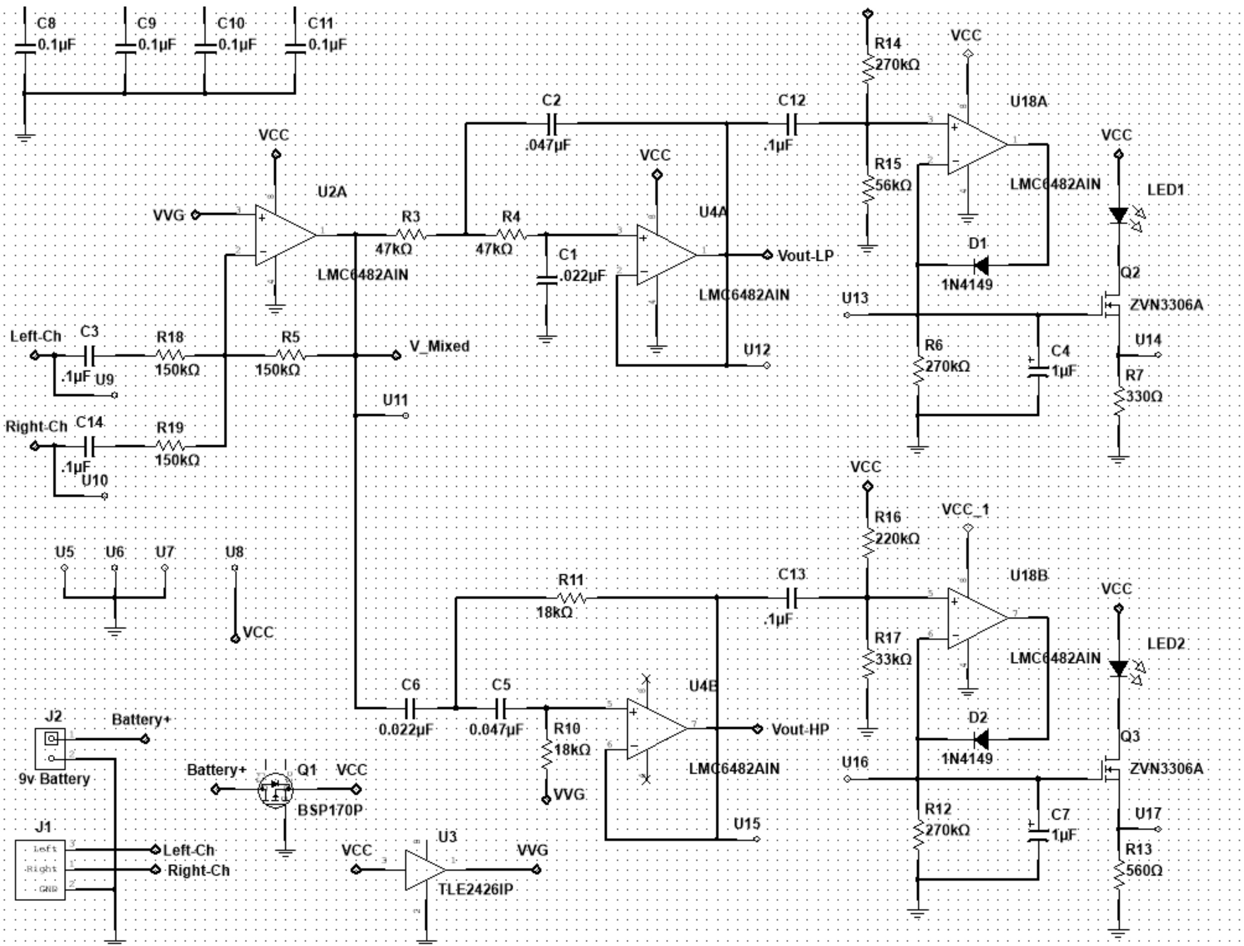


Figure 6 - Multisim schematic

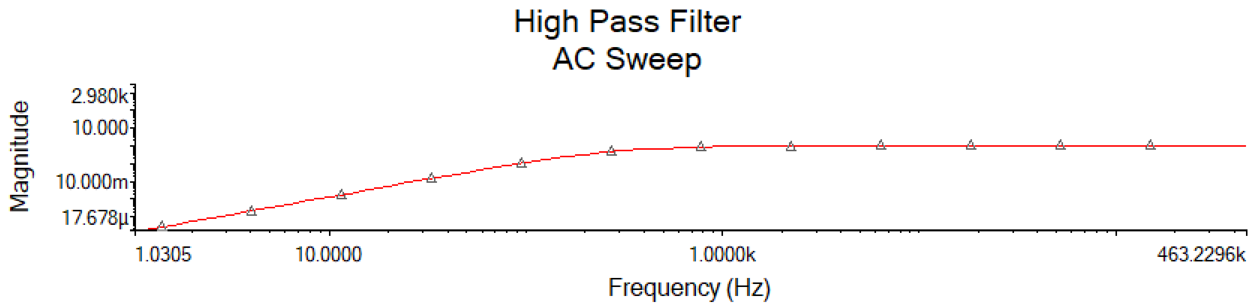


Figure 7 – AC Sweep of High Pass Filter

	Variable	Operating point value
1	V(1) | V(V_Vin)	9.00000
2	V(v_mixed)	4.56650
3	V(vcc)	8.99961
4	V(vgate)	1.54653
5	V(vvg)	4.56575

Figure 8 – DC Operating Point Analysis

Our DC operating point analysis showed us that the voltages were very close to expected values.

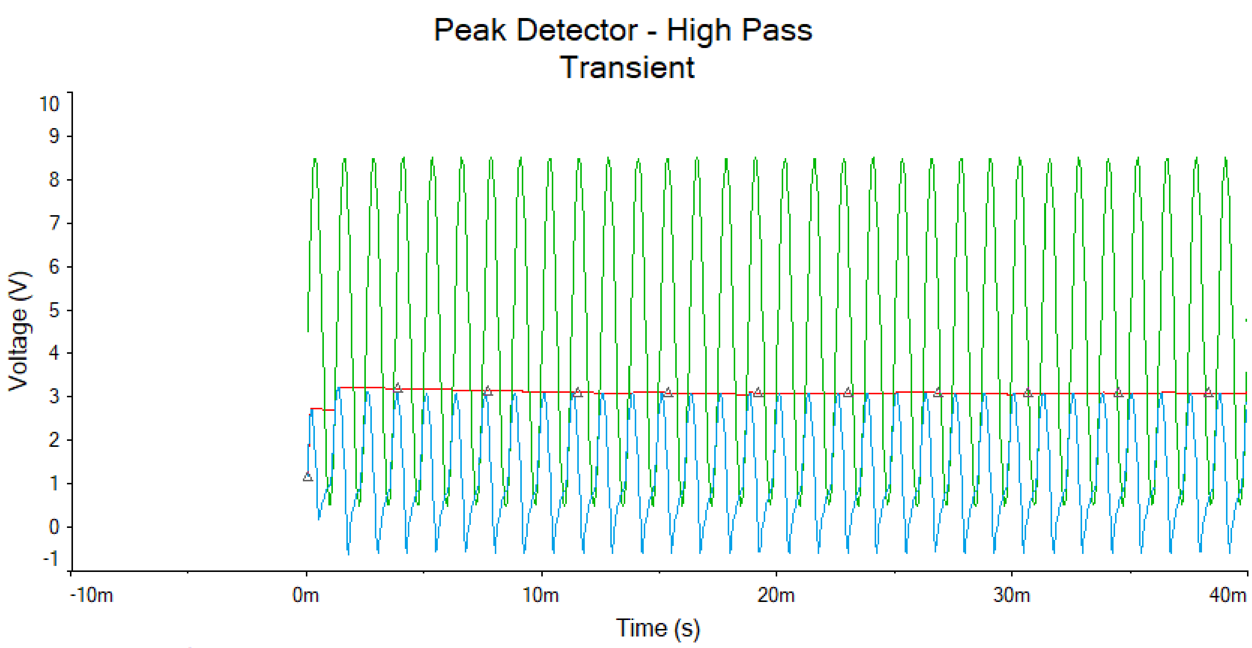


Figure 9 – Peak Detector after high pass filter transient analysis

Our transient response showed that a high frequency voltage with an offset (green line) would pass through the filter with a gain of 1 and without the DC offset (blue line). The peak detector would slowly discharge due to our small RC discharge time constant and stay close to the peak (red line).

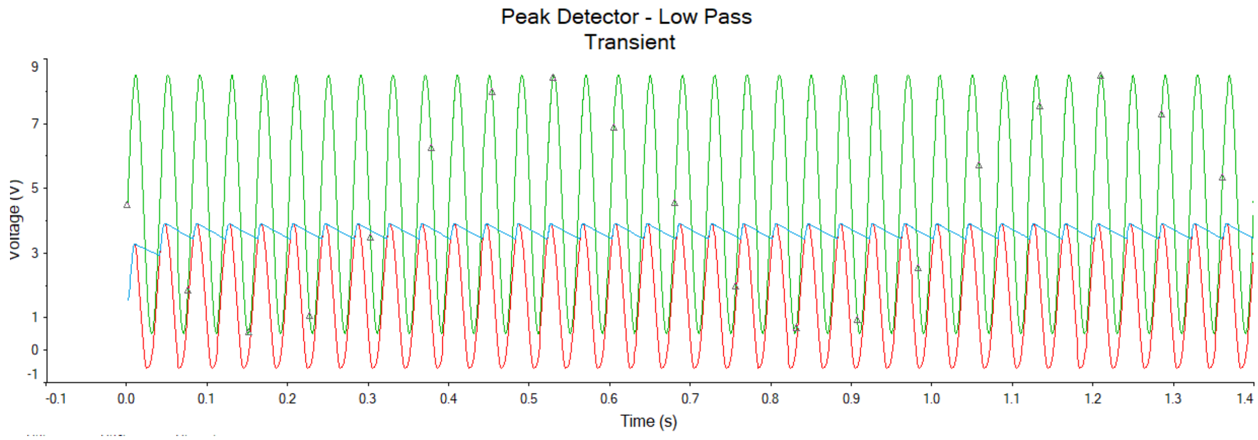


Figure 10 – Peak Detector after low pass filter transient analysis

Our transient response showed that a low frequency voltage with an offset (green line) would pass through the filter with a gain of 1 and without the DC offset (red line). The peak detector would slowly discharge due to our small RC discharge time constant and stay close to the peak (blue line).

Assembly Process

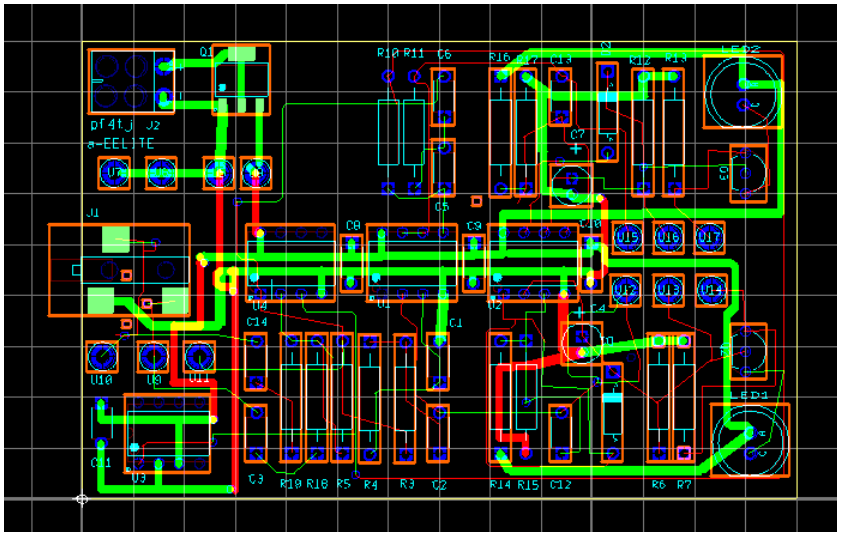


Figure 11 - EElite UltiBoard file

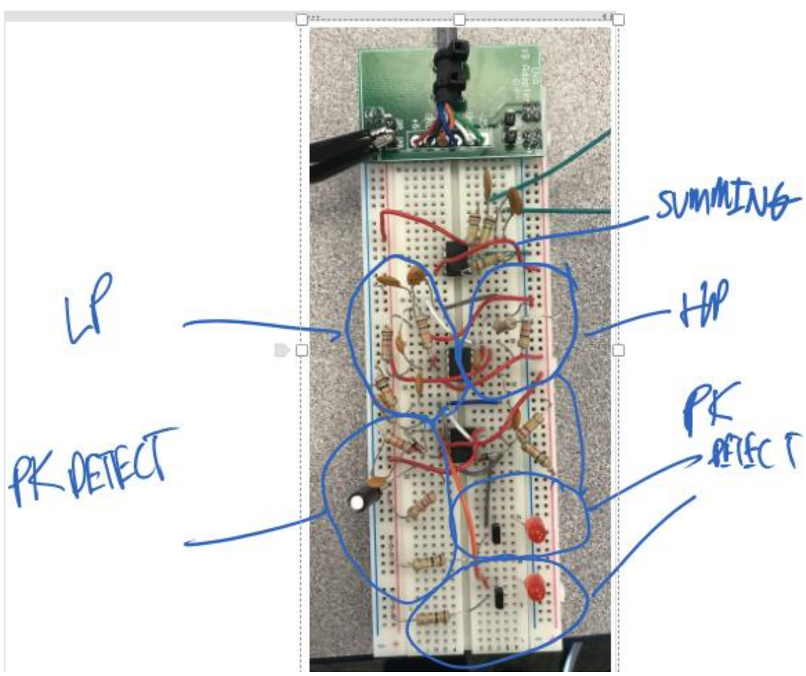


Figure 12 - Fully prototyped on a breadboard

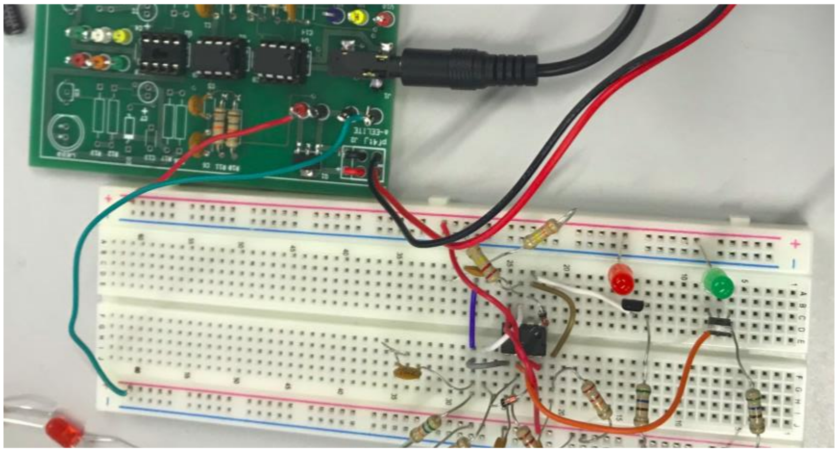


Figure 13 - Right/Left channel/Ground/Test points/ Summing MOSFETs soldered onto board

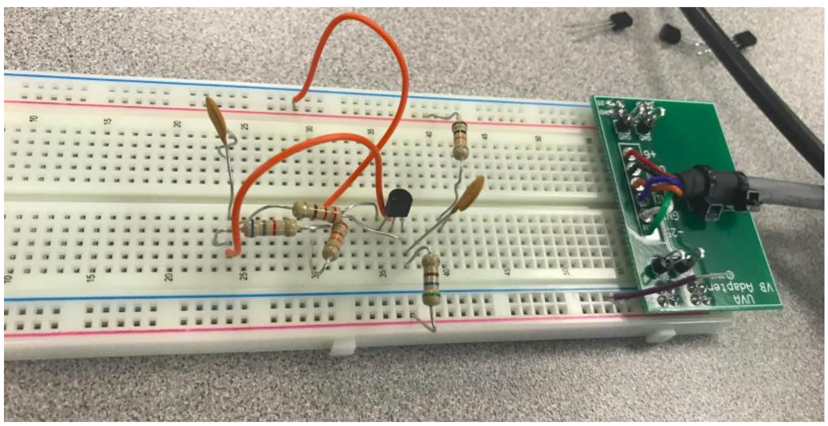


Figure 14 - Individual section testing. MOSFET gate

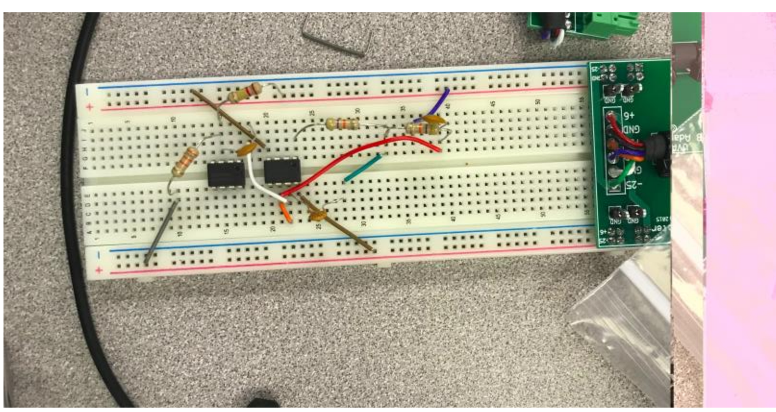


Figure 15 - Low pass filter and peak detector prototyping and testing on the breadboard

Test Point Color Codes

Color	Component
Black	Ground
Red	Right Channel VCC Vsource of MOSFET after high pass
Yellow	Left Channel LP Filter output
Green	HP Filter output Voltage source of LP
Blue	After summing opamp
White	Peak detector output after LP filter
Orange	Peak detector output after HP filter

Testing:

The testing process involved:

- Soldering required components for filter/detector etc. onto the board
- Cleaning up the soldered component connections with flux pen if necessary
- Using the DMM functionality of VirtualBench to test the connections and responses of the soldered components and whether they were sufficiently close to what we calculated analytically
    - For example, if the summed voltage from the summing amplifier is 5V and 8V is predicted analytically we needed to identify what caused the discrepancy and how to troubleshoot it.

After assembling the circuit up through the filters we tested their respective outputs using varying FGEN signal frequencies.

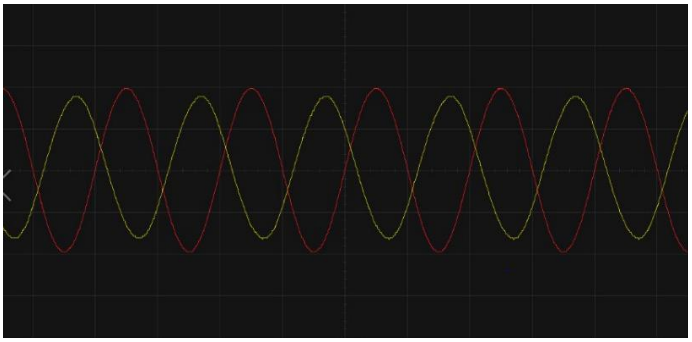


Figure 16 - Low pass filter output (yellow) when inputting signal of 50 Hz

Figure 17 - High pass filter output (yellow) when inputting signal of 50 Hz

Figures 16 and 17 above show our high and low pass filters working as desired for a frequency of 50 Hz. The phase shift in figure 16 is the result of the nature of second order filters and their effect on phase for certain frequencies.

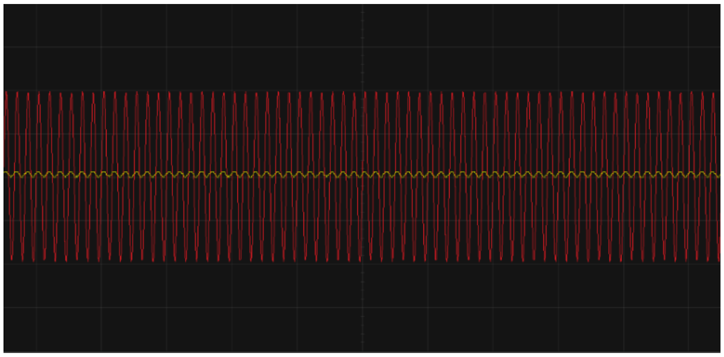


Figure 18 - Low pass filter output (yellow) when inputting signal of 600 Hz

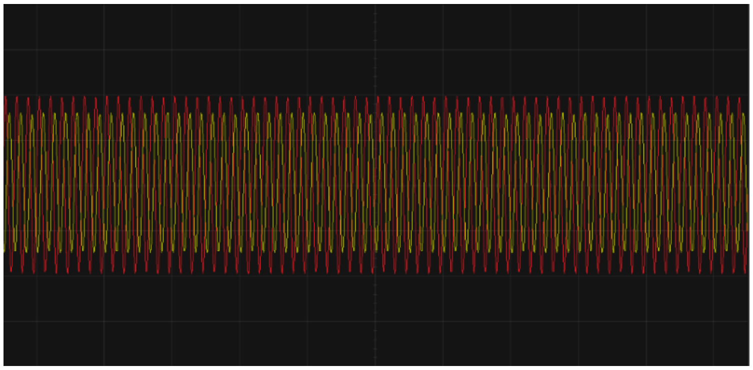


Figure 19 - High pass filter output (yellow) when inputting signal of 600 Hz

Figures 18 and 19 above show how the filters react when a signal of 600 Hz is inputted.

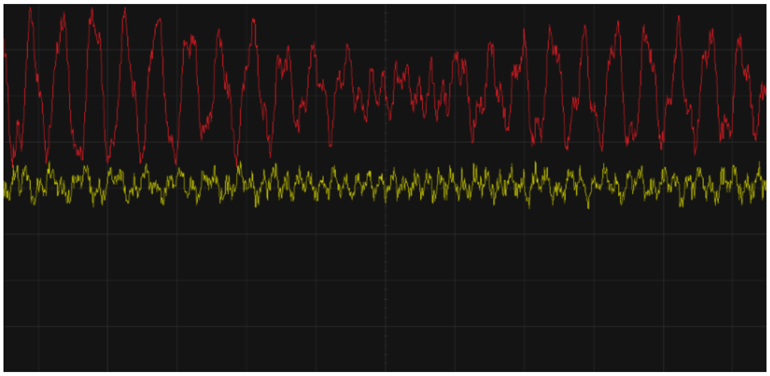


Figure 20 - Music signal in red. High pass output in yellow.

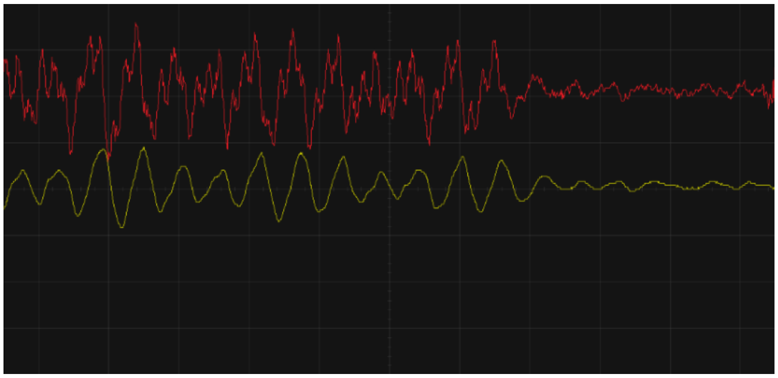


Figure 21 - Music signal in red. Low pass output in yellow.

Figures 20 and 21 above show how the filters reacted to our music signal.

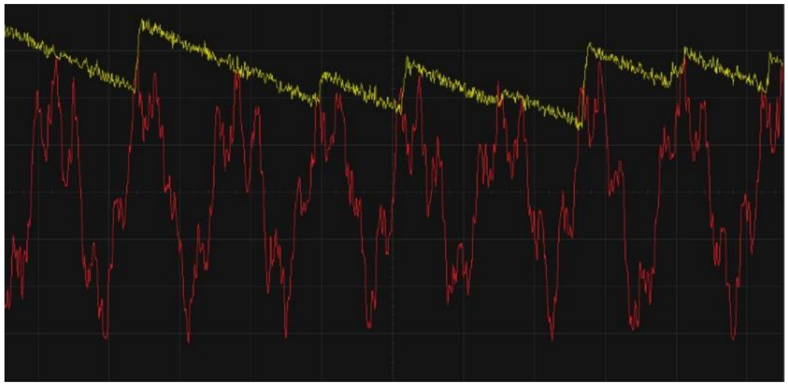


Figure 22 - Music signal in red. Peak detector output after low pass filter in yellow.

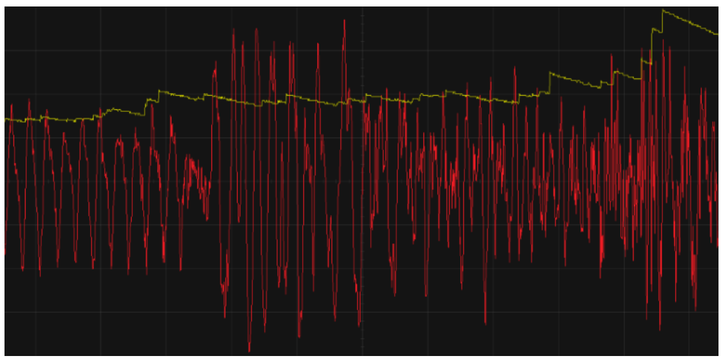


Figure 23 - Music signal in red. Peak detector output after high pass filter in yellow.

The Figures above show the resulting signal that then goes to the gate of the MOSFET to determine whether the LED should be on or not. The peak detector after the low pass filter ended up being much more noisy than we would have liked. We believe the reason for this is our time constant not being large enough.

For Future Classes:

One modification to the project that would be helpful is to add speakers to the board. Most groups had to rely on playing the music synchronously on another group member's laptop.

It would be cool to add three speakers to the board, each with their own on/off switch. One speaker that would play the music exactly as the board receives it, unmodified. One that could play the high frequencies only, and one that could play the low frequencies only. Also, could use of potentiometers or something like that allow the frequency cutoffs to be adjusted in real time? It would be really cool if we could hear the high or low frequencies playing through a speaker while we made adjustments to the cutoffs.