# ECE 2660 Final Project Report

Jack Basinet, John Berberian, and Liam Timmins

*Abstract*—**This report describes the process of designing, simulating, physically realizing, and testing an audio ”frequency splitter” - a device that blinks two LEDs depending on the frequencies input into the audio jack. The green LED should blink when there are high frequencies present, and the red LED should blink when there are low frequencies present. DC should be blocked entirely. The project required work with second-order Sallen-Key filters, diode-based peak detectors, and trans-conductance amplifiers. Furthermore, this project is an opportunity for ECE2660 students to put education to practice and work as a team in production.**

CONTENTS



This is the final project report for ECE 2660 Fundamentals 2, Spring 2023 at the University of Virginia

## I. Background Information and Rationale

The purpose of this project is to create a circuit which takes in an audio input, ideally a predetermined song, and drives current through two LED’s depending on the frequency content of the audio input. One LED is for high frequencies, and another is for low frequencies. To separate the frequency content of the input, the left and right channels of the input were run through a summing amplifier. This was then fed into a low-pass and high-pass filter, creating two signals which reflect the high frequency and low frequency content of the initial input. The corner frequencies used to separate the signal were 850Hz for the high-pass filter, and 250Hz for the low-pass filter. These values were chosen based both on the human range of distinguishing different frequencies, and on the frequency content of the song “Teeth in your Neck” by Kali Uchis, included in Fig. 1. We chose this song because of its syncopated low frequency base lines and high frequency and snare beats, which should clearly display the out of sequence rhythms with the blinking lights.

Following the high and low-pass filters, peak detectors, fed off a designed threshold DC voltage, rectify the filtered signals. Finally those rectified signals feed into MOSFETS which drive current through either LED respectively in time with the frequency extremes.

## II. Design and Analysis

### A. Summing Amplifier

The purpose of the summing amplifier/input filter block is twofold: to sum the left and right channels, and to filter out DC offsets in the signal. We were given the schematic (Fig. 2), and just had to fill in the values.

*1) Input RC Filters:* The requirements on the high-pass RC input filters were:

- Resistor must be greater than $10\text{k}\Omega$ (and be in our lab kit)
- Cutoff frequency must be around 20Hz
- The capacitor must be a ceramic capacitor in our lab kit

The left and right channel filters are identical, so we’ll only do the analysis once. By the voltage divider rule, the transfer function of the left-channel RC filter is:

$$H(s) = \frac{R_{18}}{R_{18} + \frac{1}{sC_3}} = \frac{sR_{18}C_3}{1 + sR_{18}C_3} = \frac{s}{s + \frac{1}{R_{18}C_3}}$$

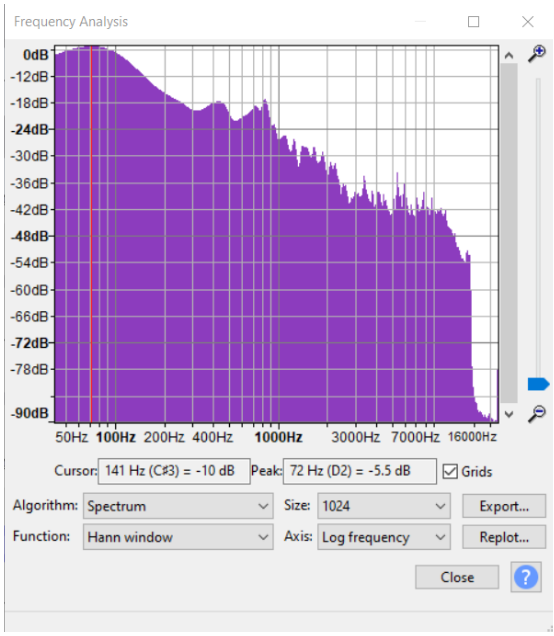



Fig. 1. Frequency content of chosen song. Note the large bump approximately bounded by 250Hz, and the bump after 850Hz.

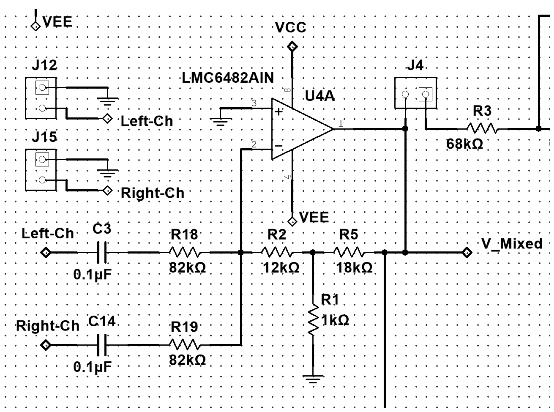



Fig. 2. Summing Amplifier Schematic

The corner frequency is:

$$\omega_c = \frac{1}{R_{18}C_3}$$

$$f_c = \frac{1}{2\pi R_{18}C_3}$$

We guessed $0.1\mu$F for the ceramic capacitor. There were limited options, and it seemed plausible. If we find that this corresponds to a resistor value above $10\text{k}\Omega$, we will accept this as the solution. Evaluating the corner frequency requirement:

$$f_c = \frac{1}{2\pi R_{18}C_3} = 20\text{Hz}$$

$$R_{18} = \frac{1}{2\pi \cdot 20\text{Hz} \cdot 0.1\mu\text{F}} \approx 79577\Omega$$

The closest resistor in our lab kit is $82\text{k}\Omega$. This corresponds to a corner frequency of:

$$\frac{1}{2\pi \cdot 82\text{k}\Omega \cdot 0.1\mu\text{F}} \approx 19.4\text{Hz}$$

This is close enough for our purposes. Note that the current through the resistor is what gives us our voltage above ground, and (as the current is forced through the summing amplifier) the RC filter's resistor is also the summing amplifier's weight resistor.

2) *Summing Op-Amp:* We now consider the part of the circuit responsible for summing the left and right channels. To create a high feedback impedance, a T feedback network was used. Within the passband of the input filters, we can model the capacitors as shorts. This will simplify our analysis. We will call the right and left input voltages $V_R$ and $V_L$, respectively. Because the filters are the same, we will use $R_{\text{in}}$ for $R_{18}$ and $R_{19}$.

The op-amp is in negative feedback, so we can assume that $V_+ = V_- = 0$V. The inverting terminal of the op-amp is a virtual ground. We will call the voltage in the middle of the T-network $V_m$. We perform node analysis:

$$\frac{0 - V_R}{R_{\text{in}}} + \frac{0 - V_L}{R_{\text{in}}} + \frac{0 - V_m}{R_2} = 0 \tag{1}$$

$$\frac{V_m - 0}{R_2} + \frac{V_m - 0}{R_1} + \frac{V_m - V_{\text{Mix}}}{R_5} = 0 \tag{2}$$

Rearranging Equation 1:

$$\frac{V_R + V_L}{R_{\text{in}}} + \frac{V_m}{R_2} = 0$$

$$V_m = -\frac{R_2}{R_{\text{in}}}(V_R + V_L) \tag{3}$$

Rearranging Equation 2:

$$V_m\left(\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_5}\right) = \frac{V_{\text{Mix}}}{R_5}$$

$$V_{\text{Mix}} = R_5 \cdot V_m\left(\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_5}\right)$$

Substituting in Equation 3:

$$\begin{aligned} V_{\text{Mix}} &= -\frac{R_2 R_5}{R_{\text{in}}}\left(\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_5}\right)(V_R + V_L) \\ &= -\frac{1}{R_{\text{in}}}\left(\frac{R_2 R_5}{R_1} + R_5 + R_2\right)(V_R + V_L) \end{aligned}$$

The linear gain of the summing amplifier is required to be $\pm 3$.

$$|H(\omega)| = \frac{1}{R_{\text{in}}}\left(\frac{R_2 R_5}{R_1} + R_5 + R_2\right) = 3$$

$$\frac{R_2 R_5}{R_1} + R_5 + R_2 = 3R_{\text{in}} = 246\text{k}\Omega$$

We wrote programs to brute-force resistor combinations that satisfy this. The code is in Appendix A1. We found that $R_1 = 1\text{k}\Omega$, $R_2 = 12\text{k}\Omega$, and $R_5 = 18\text{k}\Omega$ is a solution.

3) *Results:* Here follows a quick summary of our results.

- Input Filters
  - $R_{18} = R_{19} = 82\text{k}\Omega$

- $C_3 = C_4 = 0.1\mu$F
- $f_c = 19.4$Hz
- Passband gain $= 1$

- Summing Op-Amp
  - $R_1 = 1$k$\Omega$
  - $R_2 = 12$k$\Omega$
  - $R_5 = 18$k$\Omega$
  - Gain $= -3$
  - Feedback Impedance $= 246$k$\Omega$

### B. High-Pass Filter (HPF)

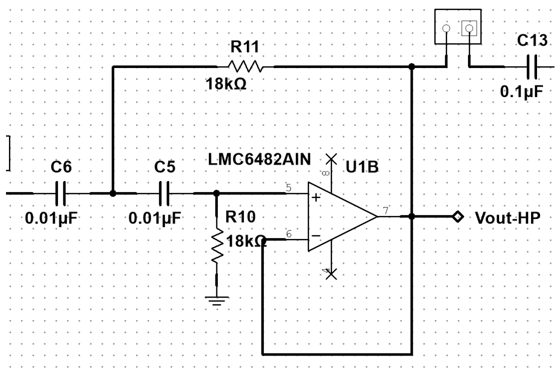



Fig. 3. Sallen-Key High-Pass Filter Schematic

To efficiently attenuate the low-frequency signals of our song, we used a second-order high-pass filter. The generic second-order high-pass transfer function is:

$$K \cdot \frac{s^2}{s^2 + s\frac{\omega_0}{Q} + \omega_0^2}$$

Given how bulky and expensive inductors are, an active Sallen-Key filter seemed appropriate. The high-pass version of this filter is shown in Fig. 3 The values for the capacitors and resistors were calculated with python code, located in Appendix A3. This code calculated what combination of components would have a gain of within 10% of $-6$dB at 850Hz, and have a quality factor, $Q$, between 0.5 and 0.707. The $Q$-factor, which describes the frequency and step response behavior, was derived in lecture 6.2b as:

$$Q = \frac{\sqrt{R_1 R_2 C_1 C_2}}{R_1 C_1 + R_2 C_1 + R_1 C_2 (1 - K)}$$

A quality factor of 0.5 would be critically damped meaning that the unit time response increases as quickly as possible without exceeding the input. The corner frequency, $\omega_0$, was derived in lecture 6.2b as:

$$\omega_0 = \frac{1}{\sqrt{R_1 R_2 C_1 C_2}}$$

The corner frequency is the point at which the high-pass filter's zeroes exist. It's a rough measure of the lowest frequency that the filter will allow to pass through. Given that all components must come from our lab kit, there existed two potential combinations of values that would produce the desired result. The chosen set of values would have both resistors be 18k$\Omega$ and both capacitors would be $0.01\mu$F. Of the two, the chosen set of values had a lower Q, 0.5, but a more accurate corner frequency, 884.19Hz, making it better suited to our purposes.

#### 1) Results:

- $R_{10} = R_{11} = 18$k$\Omega$
- $C_5 = C_6 = 0.01\mu$F
- $f_c = 884.19$Hz

### C. Low-Pass Filter (LPF)

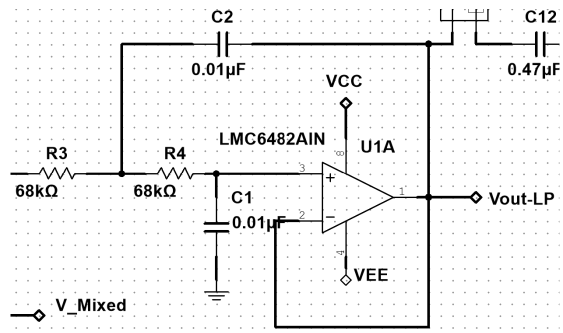



Fig. 4. Sallen-Key Low-Pass Filter Schematic

We likewise required a low-pass filter to block the higher-frequency components of our song. We had similar constraints: a compact filter with second-order roll-off, $0.5 \leq Q \leq 0.707$, a passband gain of 0dB, and a gain of $-6$dB near our low-pass break frequency of 250Hz. The generic second-order low-pass transfer function is:

$$K \cdot \frac{\omega_0^2}{s^2 + s\frac{\omega_0}{Q} + \omega_0^2}$$

We again used a Sallen-Key filter, this time in the low-pass configuration (Fig. 4). Using the same code as the previous section, we found a resistor and capacitor combination that satisfies the conditions: both resistors are 68k$\Omega$ and both capacitors are $0.01\mu$F. This yields a $-6$dB frequency of 234.05Hz with $Q = 0.5$.

#### 1) Results:

- $R_3 = R_4 = 68$k$\Omega$
- $C_1 = C_2 = 0.01\mu$F
- $f_c = 234.05$Hz

### D. Q-Point

At steady-state, the peak detector outputs the voltage on the inverting input of its op-amp. Because this is connected directly to the MOSFET's gate, this is the Q-point gate voltage for the MOSFET. We measured characteristics for the high- and low-pass MOSFETs:

- $V_{t,\text{lo}} = 2.06032$V
- $K_{n,\text{lo}} = 0.09311$A/V$^2$
- $V_{t,\text{hi}} = 2.04315$V
- $K_{n,\text{hi}} = 0.09583$A/V$^2$

We need to set the Q-point voltage below the threshold voltage for the transistor, so that it will be in the cutoff region (so the LED will be off) when there's no signal. We also want it to be close - just slightly below - so that a small signal can push it over the threshold to light up the LED. From this, we calculated resistors for the voltage dividers. We were unfortunately rather limited in our choice of resistors, and the nearest voltage divider we could build for either of the threshold voltages would be (modulo a resistor decade):

$$\frac{33\text{k}\Omega}{120\text{k}\Omega + 33\text{k}\Omega} \cdot 9\text{V} \approx 1.9411\text{V}$$

Unsure if this was sufficiently close, we asked Prof. Crockett, who recommended that we test it out. After some quick tests, we noticed that the green LED was behaving correctly, but the red LED was slightly on when there was no signal. We were somewhat confused by this - surely, given that we were below the threshold voltage, there should be no current? When we lowered $R_{15}$ to 27k$\Omega$, it exhibited the correct behavior.

Next, we had to decide the decade of the resistors and the sizing of the blocking capacitors. Unfortunately, $C_{12}$ and $C_{13}$ are not quite as big or friendly as we might like to treat them. Instead, it's important to remember that they form an RC high-pass in combination with the Q-point voltage divider resistors. We must ensure that the RC filter doesn't block any of our desired signal frequencies for a given channel.

For the low-frequency channel (red LED), the frequencies in our signal range between 19.4Hz (input high-pass filters) and 250Hz (Sallen-Key low-pass filter), so we should try to set the RC high-pass corner frequency at or below 19.4Hz. After some guesswork, we settled on $C_{12} = 0.47\mu$F, $R_{14} = 120$k$\Omega$, and $R_{15} = 27$k$\Omega$. This gives us a corner frequency of:

$$f_c = \frac{1}{2\pi \left(R_{15} || R_{14}\right) C_{12}} \approx 15.36\text{Hz}$$

Similarly, the high-frequency channel (green LED) carries a signal that contains frequencies above 850Hz (Sallen-Key high-pass filter), so we should set our high-pass corner frequency below 850Hz. Again, a bit of guesswork later, we found that $C_{13} = 0.1\mu$F, $R_{16} = 12$k$\Omega$, and $R_{17} = 3.3$k$\Omega$ gives us a corner frequency in range:

$$f_c = \frac{1}{2\pi \left(R_{17} || R_{16}\right) C_{13}} \approx 614.92\text{Hz}$$

*1) Results:*

- Low-Frequency
  - $R_{14} = 27\text{k}\Omega$
  - $R_{15} = 120\text{k}\Omega$
  - $C_{12} = 0.47\mu\text{F}$
  - $f_c = 15.36$
- High-Frequency
  - $R_{16} = 3.3\text{k}\Omega$
  - $R_{17} = 12\text{k}\Omega$
  - $C_{13} = 0.1\mu\text{F}$
  - $f_c = 614.92$

### *E. Peak Detector*

The peak detector (or rather, envelope detector) is intended to both rectify the signal and buffer the peaks so that the LEDs don't rapidly blink at the frequency of the input - we want them to light up pretty solidly, not blink. The peak detector is essentially a super-diode connected to an RC buffer circuit (Fig. 5). To properly buffer the signal, the time constant of the buffer should be significantly above the period of the expected signal.

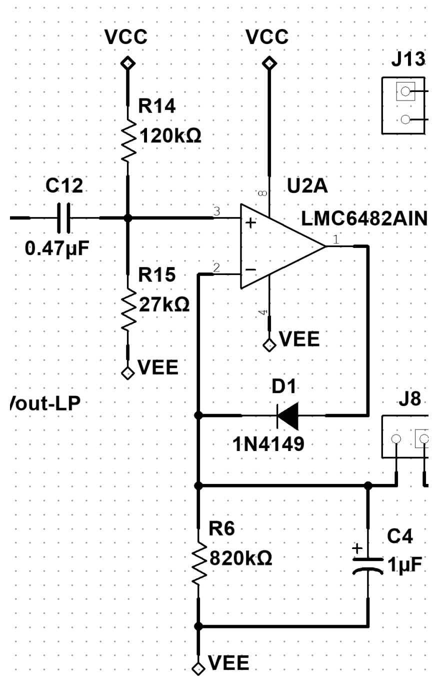



Fig. 5. Low-Frequency Peak Detector Schematic. The high-frequency counterpart is the same except for component values.

For the low-frequency channel, the lowest-frequency (highest-period) signal is expected to be near 20Hz, with a period near 50ms. The time constant is $\tau = R_6 C_4$. We calculated that $R_6 = 82\text{k}\Omega$ and $C_4 = 10\mu$F ($\tau = 820$ms) should work well. We thought thought it was a nice balance between buffering out the wavefronts and staying on too long. Unfortunately, we mistakenly soldered on a $1\mu$F capacitor, so we had to change $R_6$ to $820\text{k}\Omega$ to leave the time constant unchanged.

For the high-frequency channel, the lowest-frequency (highest-period) signal is expected to be near 850Hz, with a period near 1.18ms. We again decided to make the time constant significantly higher than the period: $C_7 = 10\mu$F and $R_{12} = 6.8\text{k}\Omega$ gives us a time constant of $\tau = 68$ms.

Both of these values were tested visually on a breadboard, and we decided that they would work fine.

*1) Results:*

- Low-Frequency Channel
  - $R_6 = 820\text{k}\Omega$
  - $C_4 = 1\mu\text{F}$
  - $\tau = 820\text{ms}$
- High-Frequency Channel
  - $R_6 = 6.8\text{k}\Omega$
  - $C_4 = 10\mu\text{F}$
  - $\tau = 68\text{ms}$

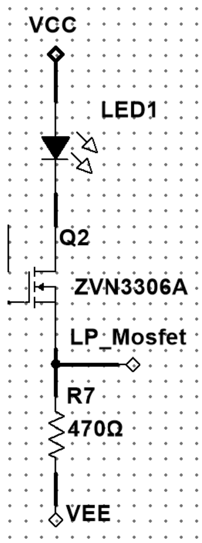



Fig. 6. Low-Frequency LED Driver Schematic. The high-frequency counterpart is the same except for component values.

*F. LED Driver*

To drive the LED itself, the output voltage from the peak detector is input into the gate of a MOSFET - just below the threshold between cutoff and saturation - which forces current through the LED. To ensure that the current doesn't get too high, we use a biasing resistor as negative feedback: more current increases the source voltage, which decreases the gate-source voltage, which decreases the current. The whole system is shown in Fig. 6.

To determine the value for the biasing resistor, we recorded the maximum voltages (relative to the negative rail) output by our peak detectors while playing the song we were targeting: $V_{G\text{max,hi}} = 3\text{V}$, $V_{G\text{max,lo}} = 4\text{V}$. Next, we used the LED datasheets to estimate a reasonable maximum-volume current ($I_{DS\text{max}}$) for each LED. Finally, we analyzed the system to find a value for the resistor that satisfied the equation:

$$I_{DS} = \frac{K_n}{2}\left(V_{GS} - V_t\right)^2$$
$$I_{DS\text{max}} = \frac{K_n}{2}\left(V_{G\text{max}} - I_{DS\text{max}} R_S - V_t\right)^2$$

For the green LED (high-frequency channel), the datasheet listed a maximum current of 30mA [1, p. 2], but we decided on a much safer maximum current of 15mA. Substituting values into our equation:

$$30\text{mA} = \frac{0.09583}{2}\left(3\text{V} - 30\text{mA} \cdot R_S - 2.04315\text{V}\right)^2$$

This yields two solutions, $101.091\Omega$ and $26.4892\Omega$. Wishing to be conservative with our estimation, we chose the larger of the two solutions, $101.091\Omega$. The closest resistor in our lab kit to that is $R_{13} = 100\Omega$.

The red LED (low-frequency channel) datasheet listed a maximum current of 7mA [2, p. 3], but we chose to use a safer value, $I_{DS\text{max}} = 5.5\text{mA}$. Again, we substituted into the equation:

$$5.5\text{mA} = \frac{0.09311}{2}\left(4\text{V} - 5.5\text{mA} \cdot R_S - 2.06032\text{V}\right)^2$$

This again yielded two solutions: $415.163\Omega$ and $290.175\Omega$, of which we chose the larger. When deciding on which of our lab kit resistors to use, we didn't want to risk decreasing $R_S$, as the margin of safety on this LED is much lower. Instead, we increased it to the nearest value above, $R_S = 470\Omega$.

*1) Results:*

- $R_7 = 470\Omega$
- $R_{13} = 100\Omega$

## III. Simulations

To verify our analytical results, we simulated the components in Multisim.

*A. Summing Amplifier*

See the required results to include in homework 2 and 3. You should write enough to give the simulations some context and explain what the reader should take away from them. Fig. 7

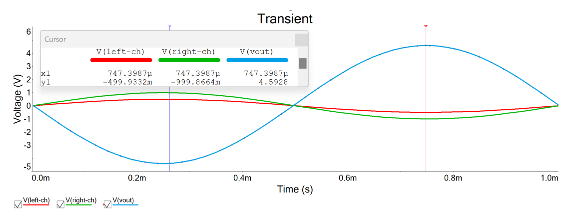



Fig. 7. Summing Amplifier Numerical Simulation: verification of summing in the passband. The left and right channels are in-phase 1kHz sines with amplitudes of 0.5V and 1V, respectively. The output is, as expected, a 1kHz sine, 180° out of phase with the inputs, with an amplitude of $(1 + 0.5) \cdot 3 = 4.5\text{V}$.

is a transient simulation of a 1V and 0.5V amplitude wave inputs and an output with a wave of the same period and an amplitude of -4.5. Considering that the gain of the summing amplifier is -3, the amplifier is working as expected in this case.

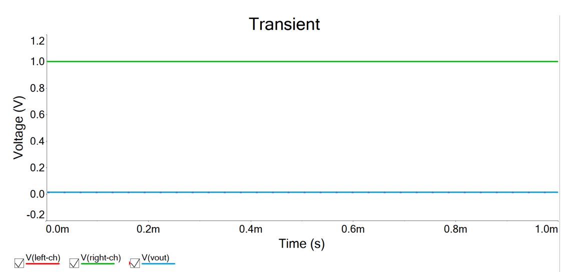



Fig. 8. Summing Amplifier Numerical Simulation: verification of the DC-blocking behavior. The left and right channels are both at 1V DC. The output is 0V, so the DC blocking should work as designed.

Fig. 8 is a transient simulation with two DC inputs with a 1V offset, which produces a DC output of 0V. Considering that this circuit removes any DC offset, this further supports that the design will work as expected.

Figures 9 and 10 show the left and right channel inputs of the summing amp producing a Bode plot of the output.

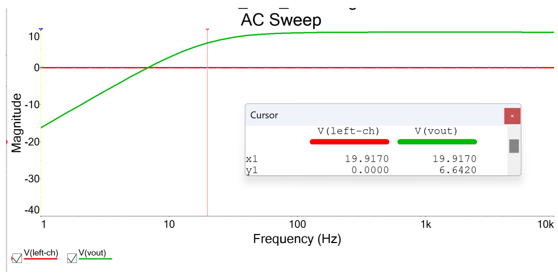



Fig. 9. Summing Amplifier Numerical Simulation: left-channel frequency response. The right channel is at 0V, the left channel did a frequency sweep. This is a plot of the gain. Note that the gain close to 20Hz is near 6.5dB, as designed. The first-order high-pass characteristic is also correct.

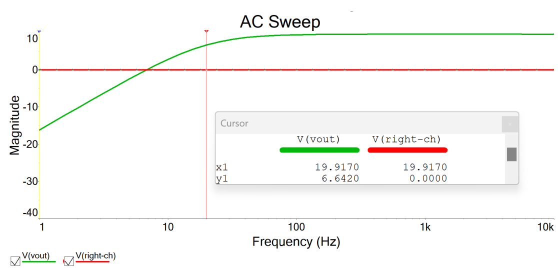



Fig. 10. Summing Amplifier Numerical Simulation: right-channel frequency response. The left channel is at 0V, the right channel did a frequency sweep. This is a plot of the gain. Note that the gain close to 20Hz is near 6.5dB, as designed. The first-order high-pass characteristic is also correct.

The corner frequency of 19.917Hz is highlighted on both, demonstrating that the corner frequency is approximately -3dB, further supporting that the summing amplifier is working as expected.

## B. High-Pass Filter

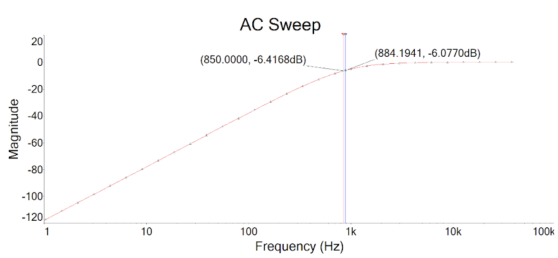



Fig. 11. Sallen-Key High-Pass Numerical Simulation: frequency response. Note the second-order high-pass characteristic and gain of $-6.077$dB at 884Hz.

Fig. 11 shows the Bode plot of the Sallen-Key high-pass filter, with both the initial planned corner frequency, 850Hz, and the real calculated corner frequency, 884.1941Hz. Considering that the magnitude of the bode plot at 884.1941Hz is -6.077dB, which can be estimated as -6dB, and that the bode plot is increasing by 40dB per decade, the analytical Sallen-Key filter operating as expected. This is significant as it means signals with greater frequency than the corner frequency will remain largely unaffected, while signals with lower frequency will become blocked.

Fig. 12 shows the DC bias point simulation for the high-pass filter, which completely blocks the 1V input. Considering that

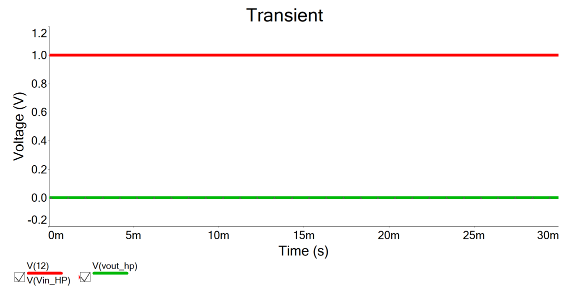



Fig. 12. Sallen-Key High-Pass Numerical Simulation: DC response. The input is 1V DC, and the output is 0V.

a DC input has a frequency of 0Hz, the filter is working as expected.

## C. Low-Pass Filter

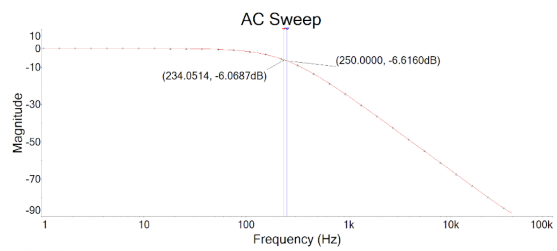



Fig. 13. Sallen-Key Low-Pass Numerical Simulation: frequency response. Note the second-order low-pass characteristic and gain of $-6.0687$dB at 234Hz.

Fig. 13 shows the Bode plot of the Sallen-Key low-pass filter, with both the initial planned corner frequency, 250Hz, and the real calculated corner frequency, 234.0514Hz. The magnitude of the bode plot at 234.0514Hz is -6.069dB, which can be estimated as -6dB, and the bode plot is decreasing by 40dB per decade following this point, supporting that the analytical Sallen-Key filter is operating as expected. This is significant as it means signals with a smaller frequency than the corner frequency will remain largely unaffected, while signals with greater frequency will become blocked.

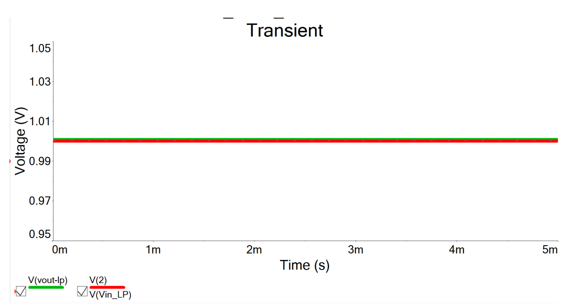



Fig. 14. Sallen-Key High-Pass Numerical Simulation: DC response. The input is 1V DC, and the output is close to 1V.

Fig. 14 shows the DC bias point simulation for the low-pass filter, which allows it to pass completely. This result is also in support of the filter working as expected, as the 0Hz frequency passes with minimal changes.

## D. Peak Detector

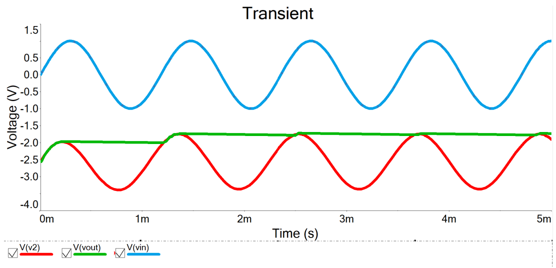



Fig. 15. Numerical High-Frequency Peak Detector Simulation. The blue input trace is a 1V amplitude 850Hz wave.

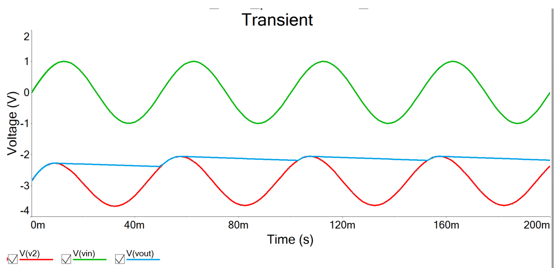



Fig. 16. Numerical Low-Frequency Peak Detector Simulation. The green input trace is a 1V amplitude 20Hz wave.

Figures 15 and 16 show a transient analysis of the peak detectors following the high-pass and low-pass filters respectively. The high-pass peak detector is a 20Hz wave, while the low-pass peak detector is a 850Hz wave. Considering that these frequencies are the lowest expected value for both peak detectors, and that there is a sufficiently slow voltage decay for both these figures, both peak detectors are expected to be functioning properly.

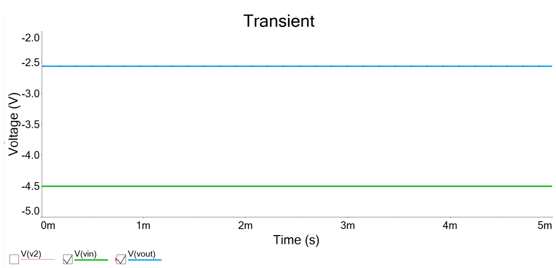



Fig. 17. Numerical High-Frequency Peak Detector. The input is DC at -4.5V, while the output is the Q-point, relative to ground.

Figures 17 and 18 show a DC bias point analysis of both peak detectors. Both peak detectors remove the DC bias and shift the voltage to approximately -2.5V due to the voltage dividers in the circuit diagram, which in this case is the Q-point of the system relative to the negative rail. Considering that both peak detectors satisfy expectations for the no input signal example, the peak detectors functioning as expected can be further supported.

## E. LED Driver

Fig. 19 and Fig. 20 show a 1kHz and 20Hz frequency input wave through the peak detector to simulate an expected wave

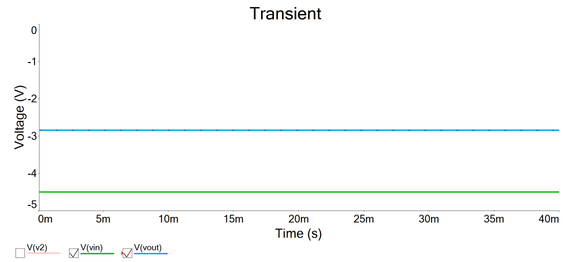



Fig. 18. Numerical Low-Frequency Peak Detector. The input is DC at -4.5V, while the output is the Q-point, relative to ground.

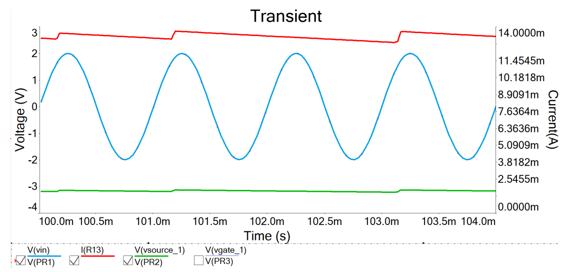



Fig. 19. Numerical LED Driver: high-frequency channel typical signal test. The input is 2V sine wave at 1kHz. The green output trace is greater than -4.5V, the negative rail, so there is current flowing through the LED, supported by the red trace.

form as an input to the LED driver. In both figures, the green trace shows the voltage across the respective biasing resistor being greater than 0V, demonstrating that for a voltage input greater than -4.5V, there will exist some current flow through the LED. The current trace in both figures being non-zero also supports this.

Fig. 21 and Fig. 22 show the LED drivers following the high-pass and low-pass blocks respectively being simulated with no input signal. Considering that for this, there exists no voltage across the biasing resistor, there must not exist current flow through the LED.

Considering both these cases for the two LED drivers, we can expect them to function as intended with these component values for the final product.

## F. System Simulations

To verify the system simulation was operating as expected, 3 sample voltage waves were input to the system to represent

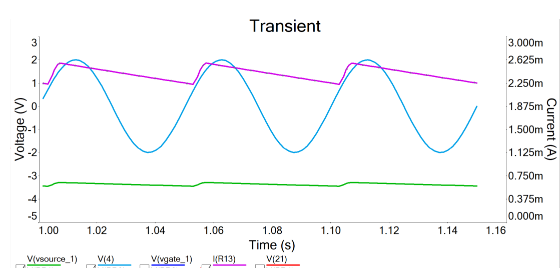



Fig. 20. Numerical LED Driver: low-frequency channel typical signal test. The input is 2V sine wave at 20Hz. The green output trace is greater than -4.5V, the negative rail, so there is current flowing through the LED, supported by the magenta trace.

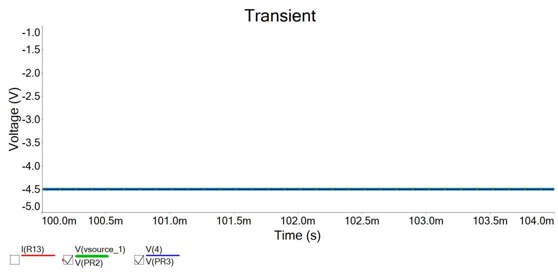



Fig. 21. Numerical LED Driver Simulation: high-frequency channel no signal test. Both the input and the output are -4.5V, so there is no current flowing through the LED.

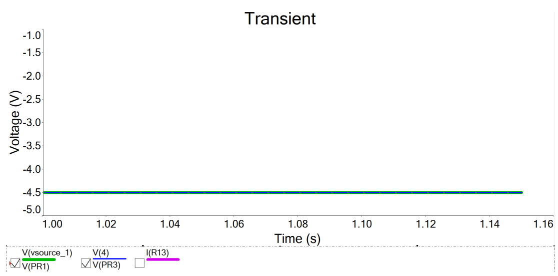



Fig. 22. Numerical LED Driver Simulation: low-frequency channel no signal test. Both the input and the output are -4.5V, so there is no current flowing through the LED.

our song: A 1V DC in Fig. 23, a 2V amplitude 20Hz frequency wave in Fig. 24, and a 2V amplitude 1kHz frequency wave in Fig. 25. For the DC case, neither LED lit up, as the voltage across the biasing resistor was 0, meaning no current was flowing to power the LED. For the 20Hz case, the red LED following the low-pass block lit, while the green LED remained dark, as the voltage across the low-frequency biasing resistor was greater than 0V, but not for the high-frequency biasing resistor. Finally, for the 1kHz case, the green LED lit, while the red remained dark because the green LED's biasing resistor had a voltage greater than 0V and the red did not.
These results indicate that the entire system is working as expected, as there is a consistent current flow for the red LED for low frequency inputs and the green LED for high frequency inputs.

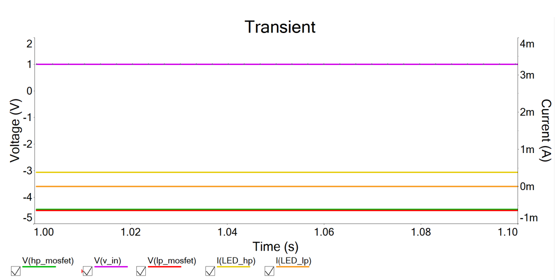



Fig. 23. System-Level Numerical Simulation: DC response. The input is 1V DC on the right audio channel, represented by the magenta trace. The green and red traces represent the voltage across the biasing resistors for the green and red LED respectively, while the yellow and orange traces represent the current across the green and red LED's, respectively. Note that both are biasing resistor voltages are -4.5V.

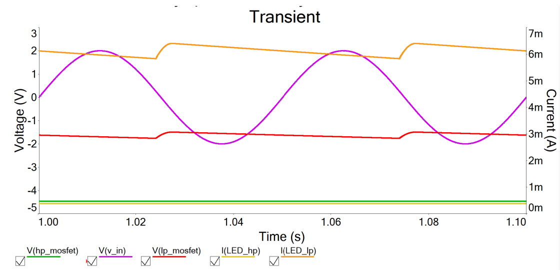



Fig. 24. System-Level Numerical Simulation: 20Hz wave response. The input is 2V amplitude 20 Hz wave on the right audio channel, represented by the magenta trace. The green and red traces represent the voltage across the biasing resistors for the green and red LED respectively, while the yellow and orange traces represent the current across the green and red LED's, respectively. Note that the green trace is -4.5V, while the red is always greater.

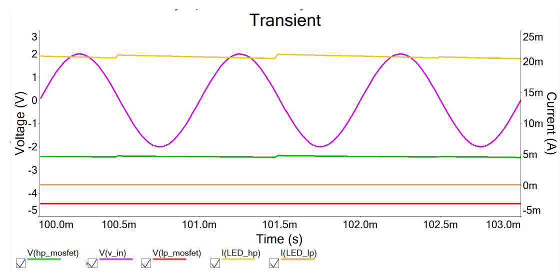



Fig. 25. System-Level Numerical Simulation: 1kHz wave response. The input is 2V amplitude 1 kHz wave on the right audio channel, represented by the magenta trace. The green and red traces represent the voltage across the biasing resistors for the green and red LED respectively, while the yellow and orange traces represent the current across the green and red LED's, respectively. Note that the red trace is -4.5V, while the green is always greater.

## IV. Experimental Results

### A. Summing Amplifier Experimental Results

The summing amplifier block has three tasks:

- To block DC inputs, and attenuate very low frequencies.
- To sum the left and right audio channels.
- To amplify the sum by a factor of 3.

We devised tests to confirm these three properties. Note that all tests were performed between J12 (left) or J15 (right) and J15 (output), with no shunts on the board. We tested these properties somewhat out of order. First, we made a bode from the perspective of each input. We expect the plots to be identical, because the circuit should be treating both inputs identically. The passband gain should be near $20\log_{10}3 \approx 9.5$dB, and the gain should be close to $9.5 - 3 = 6.5$ near our designed corner frequency of 20Hz.

The left channel bode plot is shown in Fig. 26, and the right channel is shown in Fig. 27. As expected, they display a first-order high-pass characteristic with a passband gain near 9.5dB. The gain near 20Hz is around 6.5dB. This confirms the first property.

Next, we confirmed the gain in the time domain. We input a 1kHz signal to each channel in turn and observed the output. As 1kHz is solidly in the passband of the input filter, we'd expect a linear gain of $-3$ with no attenuation. Both the right (Fig. 29) and left (Fig. 28) channels demonstrate the expected behavior for an inverting amplifier with a gain of 3 (that is, $-3$). This confirms the third property. To confirm that the

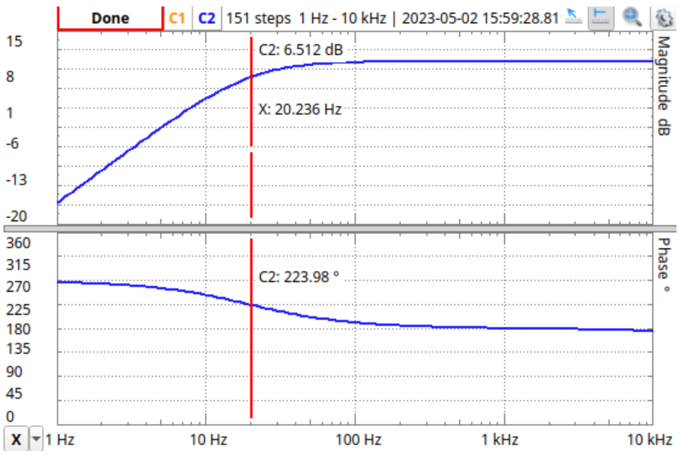



Fig. 26. Experimental Summing Amp: Left Channel Bode Plot. Note the the first-order high-pass characteristic and the gain at 20Hz : 6.5dB.

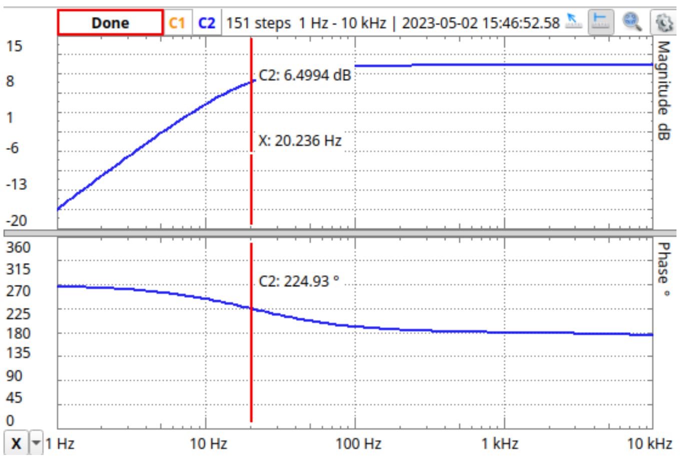



Fig. 27. Experimental Summing Amp: Right Channel Bode Plot. Note the the first-order high-pass characteristic and the gain at 20Hz : 6.5dB.

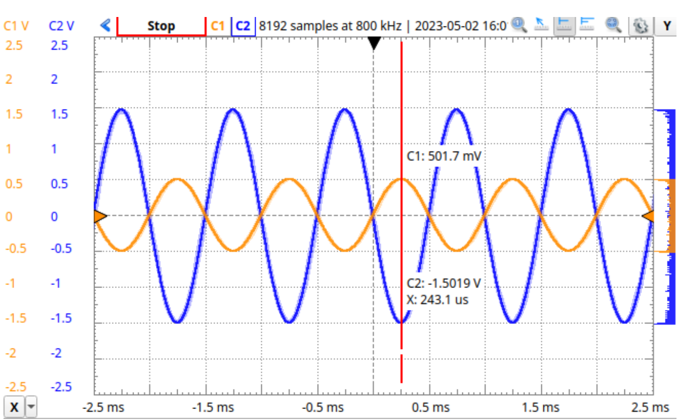



Fig. 28. Experimental Summing Amp: Left Channel Time-domain Response. The input signal was a 0.5V sine with a frequency of 1kHz. The input is on C1, the output on C2. Note the linear gain factor of $-3$.

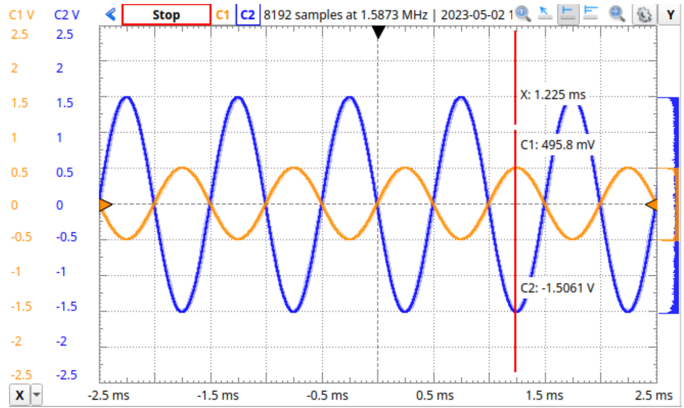



Fig. 29. Experimental Summing Amp: Right Channel Time-domain Response. The input signal was a 0.5V sine with a frequency of 1kHz. The input is on C1, the output on C2. Note the linear gain factor of $-3$.

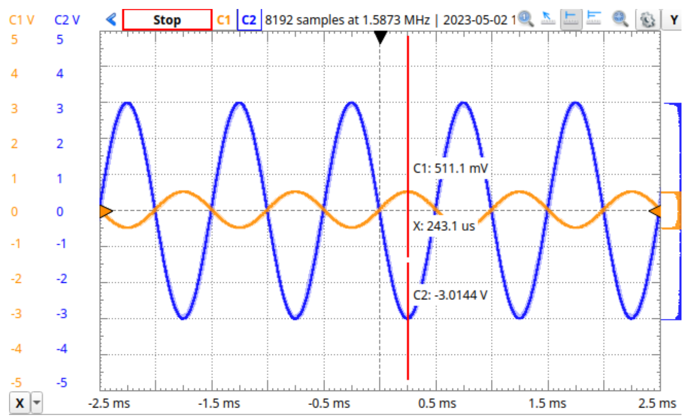



Fig. 30. Experimental Summing Amp: Both Channels Time-domain Response. Both channels have an identical 0.5V sine with a frequency of 1kHz. The input (to both channels) is on C1, the output on C2. Note the linear gain factor of $-6$.

summing was working properly, we connected both the left and right channels to the same 0.5V 1kHz sine wavegen. If the summing is correct, we'd expect to see an effective gain of $-6$ (two gains of $-3$ added together). The results are shown in Fig. 30. As expected, we see that the output for both channels is double that of one channel - that is, the linear gain factor is $-6$. This confirms the second property.

### *B. High-Pass Filter Experimental Results*

The high-pass filter is intended to attenuate frequencies below the designed corner frequency of 884Hz. To confirm that it attenuates the desired frequencies, we measured the frequency response of the filter. To isolate the filter, we removed all shunts from the board and measured between J5 (input) and J7 (output). We would expect a second-order high-pass characteristic with a gain near $-6$dB at 884Hz (ideally, it would look like Fig. 11). The bode plot for the magnitude is shown in Fig. 31. The gain near 884Hz is almost right - it looks around 1dB too low - but the second-order characteristic is correct. The discrepancy in the corner frequency gain is within the tolerance on the resistors (5%) and capacitors (20%).

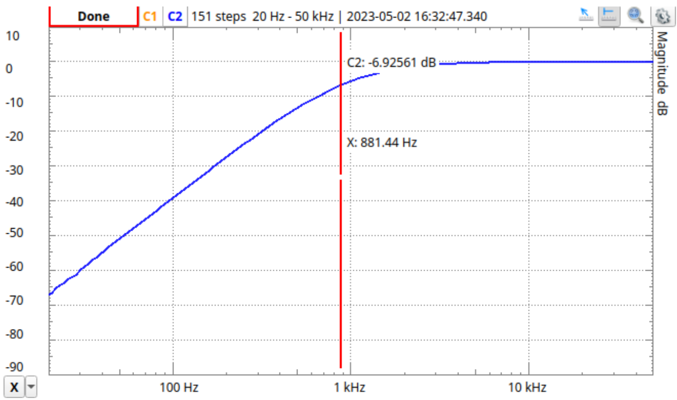



Fig. 31. Experimental Sallen-Key High-Pass Bode Plot. Note the measured gain of $-6.92$dB at 881Hz and the second-order slope of 40dB/dec.

### *C. Low-Pass Filter Experimental Results*

Similarly, the low=pass filter is intended to attenuate frequencies below the designed corner frequency of 234Hz. We again removed all shunts on the board to isolate the Sallen-Key low-pass, and generated a bode magnitude plot. Because we're measuring the low-pass filter, we needed to measure between J6 (input) and J8 (output). We expect a second-order low-pass characteristic with a gain near $-6$dB at 234Hz (should look something like Fig. 13). The bode plot is shown in Fig. 32. The gain near 234Hz is again close to right, but this time the error is in the other direction - it's around 0.7dB too high. Again, the errors are within component tolerances, and the overall characteristic of the frequency response is right.

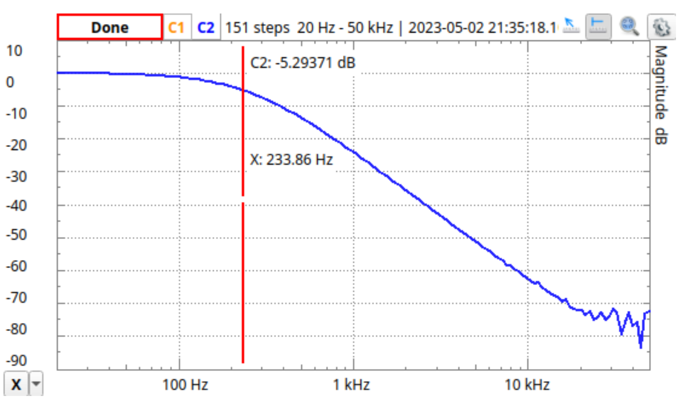



Fig. 32. Experimental Sallen-Key Low-Pass Bode Plot. Note the measured gain of $-5.29$dB at 233.8Hz and the second-order slope of $-40$dB/dec.

### *D. Peak Detector Experimental Results*

The peak detectors should buffer the signals such that there's no noticeable variation in the light brightness at the lowest expected frequency (20Hz for the low-frequency channel, 850Hz for the high-frequency channel). To verify this, we input those frequencies to the peak detector and recorded the results for both the high (Fig. 33) and low (Fig. 35) frequency channels. These approximately match Fig. 15 and Fig. 16. We also verified the behavior of the peak detectors at DC - we'd expect to see just the Q-point of the circuit here - in Fig. 34 and Fig. 36. These also match their simulated counterparts, Fig. 17 and Fig. 18.

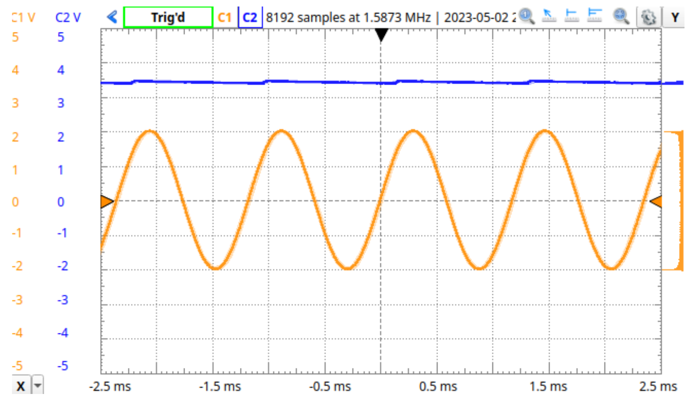



Fig. 33. Experimental High-Frequency Peak Detector. The input is a 2V sine at 850Hz. The output is measured relative to the negative rail, and is increased by the addition of the Q-point.

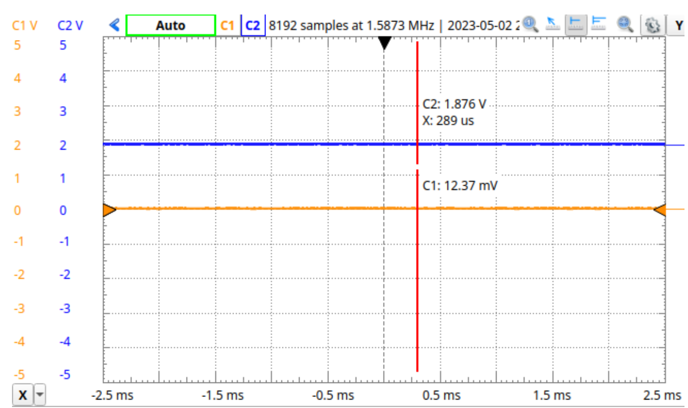



Fig. 34. Experimental High-Frequency Peak Detector. The input is DC 0V. The output is measured relative to the negative rail - in this case, it's just the Q-point.

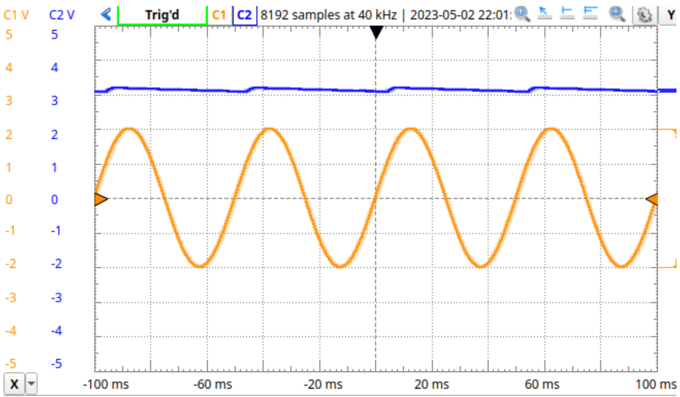



Fig. 35. Experimental Low-Frequency Peak Detector. The input is a 2V sine at 20Hz. The output is measured relative to the negative rail, and is increased by the addition of the Q-point.

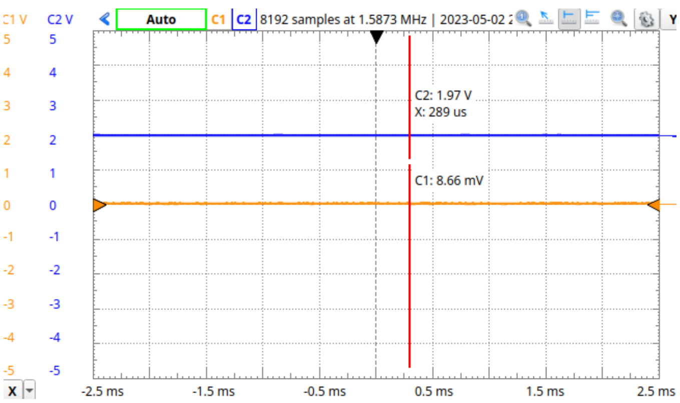



Fig. 36. Experimental Low-Frequency Peak Detector. The input is DC 0V. The output is measured relative to the negative rail - in this case, it's just the Q-point.

## *E. LED Driver Experimental Results*

We want to verify that the current through the LED is 0 when there's no signal into the peak detector, and positive when there is a signal. This will confirm that our Q-point is neither too high (no current for no signal) nor too low (a typical signal is able to push the MOSFET from cutoff into saturation).

We can't directly measure the current through the LED, but we can measure the voltage across $R_7$ and $R_{13}$ - that is, the voltage at J13 and J16 relative to the $-4.5$V rail. If the voltage is nonzero, there must be current through the LED.

After shunting J8 and J9, we input typical signals into J6 and J7. The output was recorded on the non-ground pins of J13 and J16. Both input and output are shown in Fig. 37 and Fig. 38. Notably, the voltage across the biasing resistors is nonzero, so there is current flowing through the LED when a signal is put in. This confirms that our Q-points aren't too low. Conversely, Fig. 39 and Fig. 40 show that there is no voltage across the biasing resistor (and hence no current through the LED) when there is no signal into the system. This confirms that our Q-points aren't too high - when there's no signal, the LEDs are off.

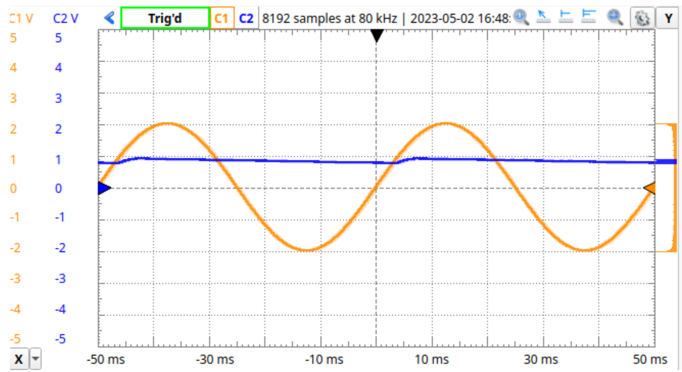



Fig. 37. Experimental LED Driver: low-frequency channel typical signal test. The input is a 2V sine at 20Hz. The output (measured relative to the negative voltage rail) is nonzero, so there is current through the LED.

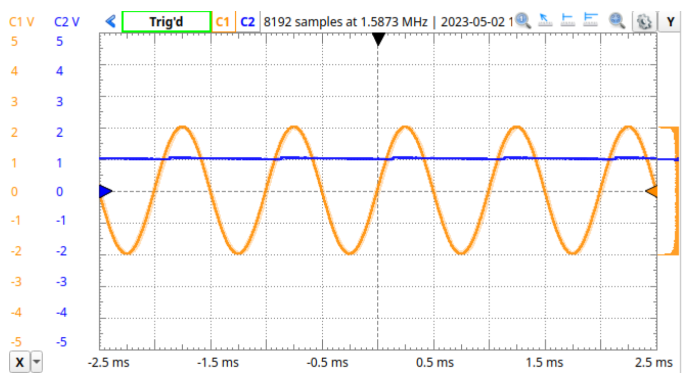



Fig. 38. Experimental LED Driver: high-frequency channel typical signal test. The input is a 2V sine at 1kHz. The output (measured relative to the negative voltage rail) is nonzero, so there is current through the LED.

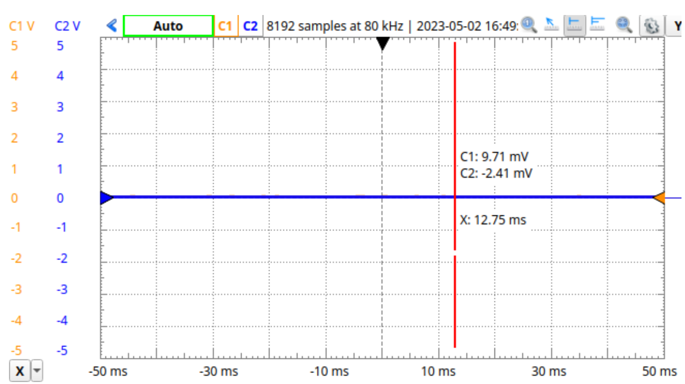



Fig. 39. Experimental LED Driver: low-frequency channel no signal test. Both the input and output are zero, so there's no current through the LED. Note that the output is measured relative to the negative voltage rail.

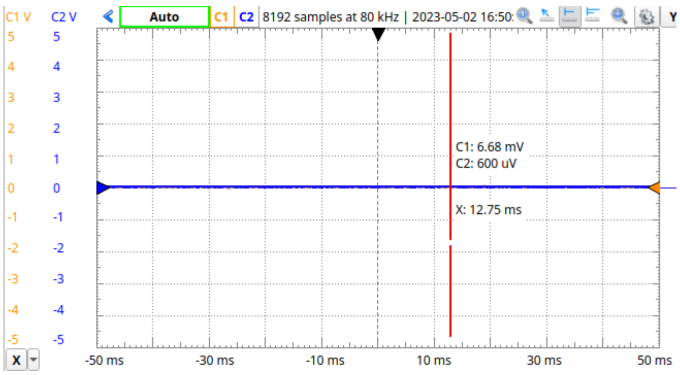



Fig. 40. Experimental LED Driver: high-frequency channel no signal test. Both the input and output are zero, so there's no current through the LED. Note that the output is measured relative to the negative voltage rail.

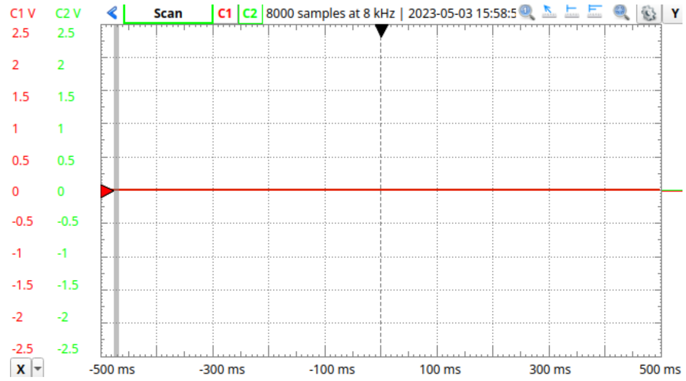



Fig. 41. Experimental System-Level: DC response. The input is 5V DC on the right audio channel. The red and green traces correspond to the voltages across the biasing resistors for the red and green LEDs. Note that both are zero.

### F. System Level Experimental Results

To verify the whole system, we played our song into the audio jack and observed the behavior of the LEDs. As a quick sanity check, we also input DC, low frequency, and high frequency signals to confirm that the correct LEDs turned on. Measurements were taken between J15 and J13 or J16, with shunts on J4, J5, J6, J7, J8, and J9.

When the song was played, the red LED noticeably blinked when the bass dropped, and the treble blinked on the snare drum and hi-hat hits. This confirmed that the LEDs were acting as expected - the snare and hi-hats have greater high-frequency components, and the bass had greater low-frequency components.

We also found the voltages across the biasing resistors when specific frequencies are input into the system: DC (Fig. 41), 20Hz (Fig. 42), and 1kHz ((Fig. 43). The voltage across the biasing resistors corresponds linearly to the current through the LEDs - if it's zero, the LEDs are off. As expected, DC was blocked - there was no current through either LED - a 20Hz signal only activated the low-frequency channel - there was only current through the red LED - and a 1kHz signal only activated the high-frequency channel.

From these results, we can confidently state that the whole system is operating as designed. The red LED turns on for low frequencies, the green LED turns on for high frequencies, and DC signals are blocked by the input filters.

## V. Challenges and Workarounds

C1: Calculations and MOSFET measurements for the low-frequency channel MOSFET's Q-point resulted in the red LED being on when there was no signal.

W1: We adjusted the Q-point down by 0.288V. This kept the red LED off when there was no signal.

C2: The high-pass MOSFET was not operating as expected.

W2: After comparing it to the low-pass MOSFET and checking that no other issues were present, the high-pass MOSFET was desoldered and found to be defective. We replaced it with a new one.

C3: The green LED was initially too dim when we set its maximum current to 5.5mA.

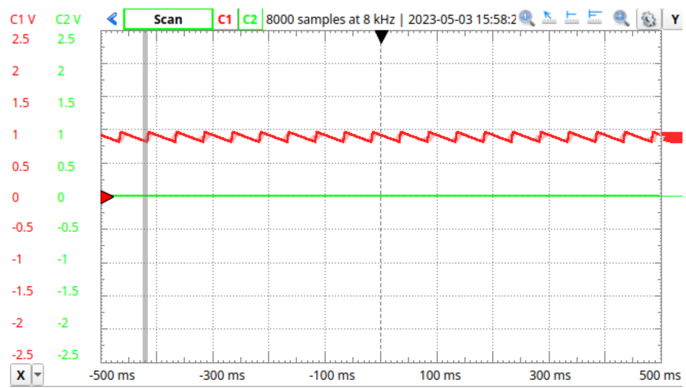



Fig. 42. Experimental System-Level: low-frequency response. The input is 1V sine at 20Hz on the right audio channel. The red and green traces correspond to the voltages across the biasing resistors for the red and green LEDs. Note that the green is zero and the red is positive.

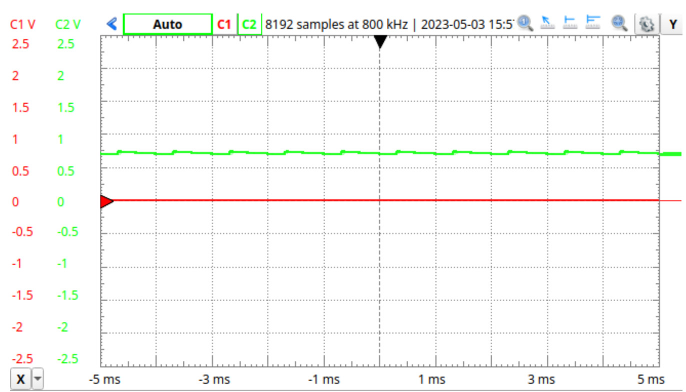



Fig. 43. Experimental System-Level: high-frequency response. The input is 1V sine at 1kHz on the right audio channel. The red and green traces correspond to the voltages across the biasing resistors for the red and green LEDs. Note that the green is positive and the red is zero.

W3: We increased the maximum current for the green LED to 15mA. This increased the brightness to be approximately equal (determined visually) with the red LED.
C4: The low-pass peak detector was not operating as expected - it wasn't visibly buffering low-frequency wavefronts.
W4: The RC time constant was too low, resulting in the output looking identical to the input - the RC buffer would decay almost instantly to the same level as the input signal. Adjusting the resistor and capacitor values to increase the time constant fixed this issue.

## VI. Conclusions and Thoughts for Future Classes

All in all this team project was a great success. The group built a well tuned and working product. Additionally, every exercise was educational and gave us opportunities to apply the techniques we've learned in each module of this course. We learned how to utilize op-amp configurations, tune circuits to desired outcomes, process and manipulate different kinds of signals, design voltage controlled current sources (VC-CS), apply component and circuit models to certain constraints and conditions, both design/build and analyze/trouble shoot, and importantly to tackle large tasks with cooperation.

For future projects and courses, we plan to use many of the strategies that worked well on this project. Delegation, open communication, and confidence in each other kept us on top of tasks, allowed us to solve tough problems, and made sure everyone was involved and present. Some improvements for group work might be more structured verification, screenshot, and testing standards so that we could stay more organized with resources, and secondly consistent task recording and checklists could make our group goals more clear. Finally, we believe that the formulation of the project was well thought out and intuitive; it followed the learning flow of the course. We'd even recommend other project based courses use this class as a guideline for combining subject and discovery.

## VII. Collaboration Statement

**John Berberian:** Wrote the summing amplifier and input filter sections, wrote the associated code for the summing amplifier. Wrote the analysis for the Q-point, the peak detector, and the LED driver. Also wrote most of the captions for figures, wrote the whole experimental report, and helped with general formatting and editing.

**Liam Timmins:** This student wrote the background information and rationale sections, as well as the parts of the report pertaining to the high-pass and low-pass filters. They additionally wrote much of the numerical simulation sections.

**Jack Basinet:** Completed the summing amplifier and high pass and low pass filter verifications. Additionally helped on the rationale sections and wrote the conclusions and abstraction.

## References

[1] *High efficiency led in ϕ 3 mm tinted diffused package*, TLHG4400-MS12, Vishay Semiconductors, 2022. [Online]. Available: https://www.digikey.com/en/products/detail/vishay-semiconductor-opto-division/TLHG4400-MS12/6594865.

[2] *Hlmp-4700, hlmp-4719, hlmp-4740 hlmp-1700, hlmp-1719, hlmp-1790*, HLMP-1700-B0002, Broadcom Inc., 2021. [Online]. Available: https://www.digikey.com/en/products/detail/broadcom-limited/HLMP-1700-B0002/1234784.

| Tasks | Jack Basinet | John Berberian | Liam Timmins |
|---|---|---|---|
| Soldering non-design components | Group 3 | Group 2 | Group 1 |
| Board beep check | | | X |
| Input filter and summer: design/-analysis | | X | |
| Input filter and summer: simulations | X | | |
| Input filter and summer: soldering | X | | |
| Input filter and summer: board testing | | | X |
| LPF: design and analysis | | | X |
| LPF: simulations | | | X |
| LPF: soldering | | X | |
| LPF: board testing | X | | |
| HPF: design and analysis | X | | |
| HPF: simulations | X | | |
| HPF: soldering | | | X |
| HPF: board testing | | X | |
| Peak detector: design and analysis | | X | |
| Peak detector: simulations | | X | |
| Peak detector: soldering | X | | |
| Peak detector: board testing | | | X |
| MOSFET+LED: design and analysis | | | X |
| MOSFET+LED: simulations | | | X |
| MOSFET+LED: soldering | X | | |
| MOSFET+LED: board testing | | X | |
| Final project report tasks | Waveform testing, some analytical derivations, simulations. Graph and screenshot making | Analytical work, high-precision simulations. Overall formatting, LaTeX help. | Helping format, providing numerical plots, and deriving analytical equations. |
| Final project demo video tasks | Editing, recording. | Demonstrating functionality. | Writing script/outline |

TABLE I
TEAM TASKS

# APPENDIX

## A. Code

This appendix contains all the scripts and code used in our project. Most of these were too large to place in the text directly.

*1) Summing Amplifier:* This is the script that brute-force guesses the resistor values for the summing amplifier. It was invoked with `python3 part2_rescalc.py 246000`. It requires another file, `resistors.lst`, that contains all the resistors in the lab kit, to be in the same directory. The contents of `resistors.lst` are given in Appendix A2.

```python
#!/usr/bin/env python3
import sys

RESISTORS_LIST="resistors.lst"
TARGET_TOLERANCE=0.05
ROW_LIMIT=50

# Gets the numeric resistor value from the encoded name.
def resistor_value(enc_name):
    multiplier = 1
    # Empty resistor names are zero ohms.
    if len(enc_name) == 0:
        return 0
    # If there's a letter at the end (k, K, M),
    # change the multiplier.
    match enc_name[-1]:
        case 'k':
            multiplier=1_000
        case 'K':
            multiplier=1_000
        case 'M':
            multiplier=1_000_000
        case _:
            pass
    if multiplier > 1:
        # The resistor can't be just a multiplier.
        if len(enc_name) == 1:
            return 0
        # Convert the rest of the name to a float
        # and multiply it by the detected multiplier.
        return float(enc_name[0:-1])*multiplier
    # If there is no multiplier, treat the whole name as a float.
    return float(enc_name)

# Read all the resistors from the file RESISTORS_LIST.
def read_resistors(fname):
    resistors = []
    # Read the file first.
    with open(fname, 'r') as f:
        lines = f.readlines()
    # Iterate through the lines, and convert each one to a value.
    for l in lines:
        resistors.append(resistor_value(l.strip()))
    return resistors

# A small helper function that calculates the percent
# error by which the arrow is off from the target.
def percent_error(target, arrow):
    return abs(target-arrow)/target
```

```
# A helper so that we can sort a list of tuples
# by the second term.
def second_term(e):
    return e[1]

# Brute-force the resistor combinations to get a specific resistance for the
# high-impedance feedback T network. Results are returned sorted by precision,
# and combinations that aren't within the specified percent of the target
# are eliminated from the list.
def three_resistor_combinations(resistors, target_value, within_percent):
    # The range that a combination's resistance has to be in.
    target_high = (1+within_percent)*target_value
    target_low  = (1-within_percent)*target_value
    answers=[]
    # Iterate through all resistor permutations with replacement.
    for r1 in resistors:
        for r2 in resistors:
            for r5 in resistors:
                # The resistance of the T network.
                comb = r2+r5+r2*r5/r1
                # If it's within range, append it.
                if target_low <= comb <= target_high:
                    # Make a tuple: a formatted string with the values, the percent error,
                    # and the combined resistance.
                    answers.append(("{0:10f}\t,\t{1:10f}\t,\t{2:10f}".format(r1, r2, r5),
                                   percent_error(target_value, comb), comb))
    # Sort the list by percent error.
    answers.sort(key=second_term)
    return answers

if __name__ == '__main__':
    # A bit of input sanitization.
    if len(sys.argv) != 2:
        print("Illegal number of arguments!")
        exit(1)
    # The only parameter is the desired resistance.
    desired_value = float(sys.argv[1])
    # Read in the list of resistors.
    resistors = read_resistors(RESISTORS_LIST)
    i=0
    # Print out a table header.
    print("{0:10}\t,\t{1:10}\t,\t{2:10}".format("r1", "r2", "r5"))
    # Get all three-resistor combinations and print them out. If we exceed the
    # row limit, stop.
    for c in three_resistor_combinations(resistors, desired_value, TARGET_TOLERANCE):
        if i > ROW_LIMIT:
            break
        print(c[0]+"\t\t{:.4%}\t\t{:f}".format(c[1], c[2]))
        i+=1
```

2) *Resistor List:* This is a file required by some of the code we used. It is a list of all the resistors in the lab kit.

```
10
12
15
18
22
```

```
27
33
39
47
56
68
82
100
120
150
180
220
270
330
390
470
560
680
820
1k
1.2k
1.5k
1.8k
2.2k
2.7k
3.3k
3.9k
4.7k
5.6k
6.8k
8.2k
10k
12k
15k
18k
22k
27k
33k
39k
47k
56k
68k
82k
100k
120k
150k
180k
220k
270k
330k
390k
470k
560k
680k
820k
1M
```

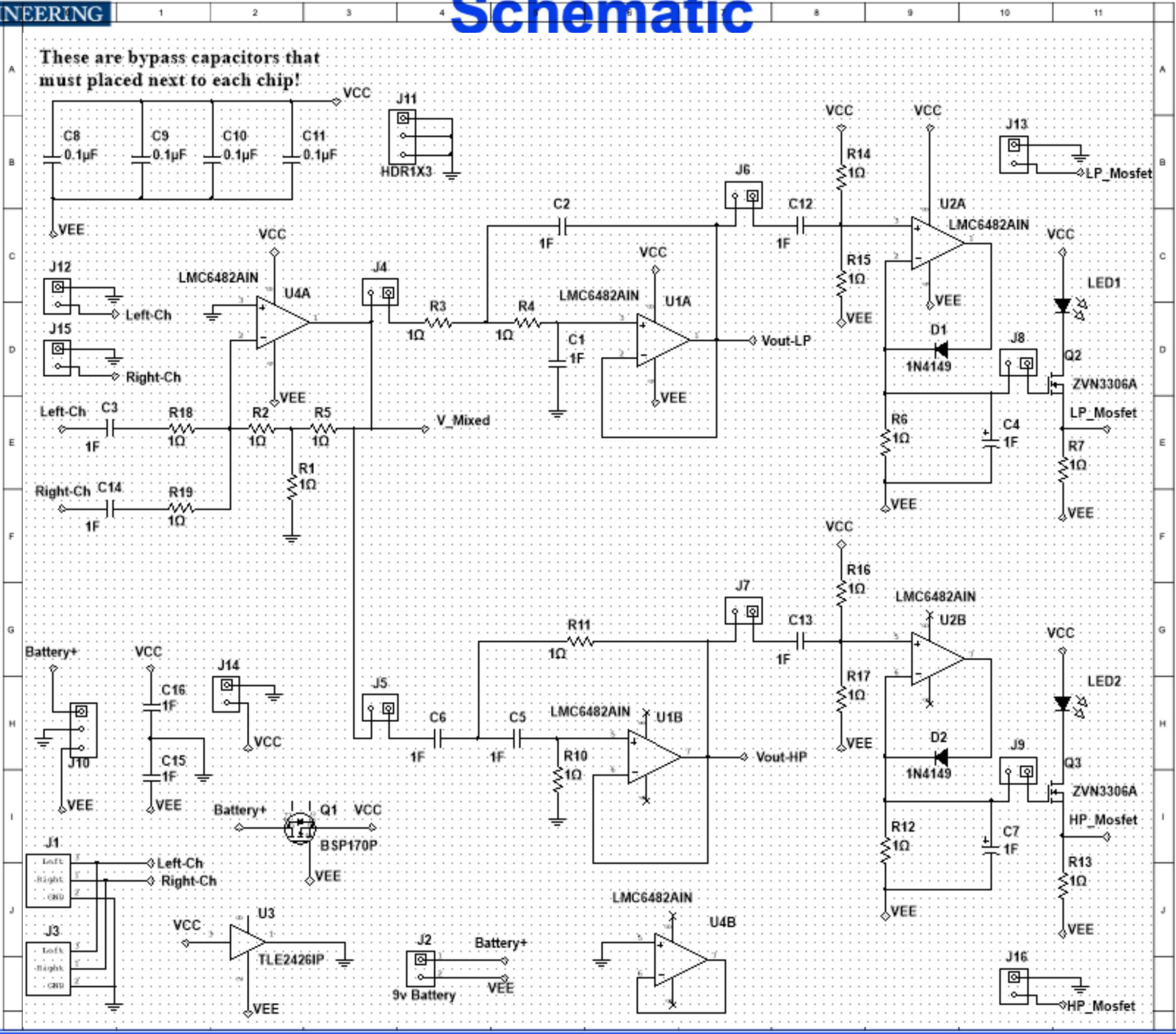
*3) High and Low-Pass Filters:* This script brute-force guesses the resistor and capacitor values for the high-pass and low-pass filters. This script already includes all the relevant resistor and capacitor values, so no other files are needed.

```
import math
RESISTORS = [10000, 12000, 13000, 18000, 22000, 27000, 33000, 39000, 47000, 56000, 68000,
82000, 100000, 120000, 150000, 180000, 220000, 270000, 330000, 390000, 470000, 560000,
680000, 820000, 1000000]
#Create list of available resistors
CAPACITORS = [0.01*10**(-6), 0.1*10**(-6), 0.47*10**(-6), 10**(-6), 4.7*10**(-6), 10*10**(-6)]
#Create list of available capacitors
CUTOFF_FREQUENCY_HP = 850 * 2 * math.pi
CUTOFF_FREQUENCY_LP = 250 * 2 * math.pi
#Define high-pass and low-pass cutoff frequency in rad/s
print("High-pass:")
print("--------")
for r10 in RESISTORS:
   for r11 in RESISTORS:
       for c6 in CAPACITORS:
           for c5 in CAPACITORS:
               #nested for loop for all resistor and capacitors in the high-pass block
               K = 1
               Q = ((r10*r11*c6*c5)**(1/2))/(r11*c5+r11*c6+r10*c5*(1-K))
               omega0 = (r10*r11*c6*c5)**(-1/2)
               #Using the equations for Q and omega0 calculated in section II-B, every possible
               #combination of resistors and capacitors are run through to determine which
               #meets the specified requirements
               if (Q >= 0.5) and (Q <= 0.707):
                   if (0.9*CUTOFF_FREQUENCY_HP<=omega0) and (omega0<=1.1*CUTOFF_FREQUENCY_HP):
                       print("R10 = "+str(r10))
                       print("R11 = "+str(r11))
                       print("C6 = "+str(c6))
                       print("C5 = "+str(c5))
                       print("Q = "+str(Q))
                       print("corner frequency = "+str(omega0/(2*math.pi)))
                       #convert from rad/s to Hz
                       print("----------")
print("Low-pass:")
print("--------")
for r3 in RESISTORS:
   for r4 in RESISTORS:
       for c1 in CAPACITORS:
           for c2 in CAPACITORS:
               #nested for loop for all resistor and capacitors in the low-pass block
               K = 1
               Q = ((r3*r4*c1*c2)**(1/2))/(r3*c1+r4*c1+r3*c2*(1-K))
               omega0 = (r3*r4*c1*c2)**(-1/2)
               #The process for the highpass filter is repeated here, with different expected
               #omega0
               if (Q >= 0.5) and (Q <= 0.707):
                   if (0.9*CUTOFF_FREQUENCY_LP<=omega0) and (omega0<=1.1*CUTOFF_FREQUENCY_LP):
                       print("R3 = "+str(r3))
                       print("R4 = "+str(r4))
                       print("C1 = "+str(c1))
                       print("C2 = "+str(c2))
                       print("Q = "+str(Q))
                       print("corner frequency = "+str(omega0/(2*math.pi)))
                       #convert from rad/s to Hz
```

```
                        print("----------")
```

### B. Full Project Schematic

# Schematic


These are bypass capacitors that must placed next to each chip!
C8 0.1μF
C9 0.1μF
C10 0.1μF
C11 0.1μF
VCC
VEE
J11
HDR1X3
J12
Left-Ch
J15
Right-Ch
Left-Ch C3 1F
R18 1Ω
Right-Ch C14 1F
R19 1Ω
LMC6482AIN
U4A
R2 1Ω
R5 1Ω
R1 1Ω
J4
V_Mixed
R3 1Ω
R4 1Ω
C1 1F
C2 1F
LMC6482AIN
U1A
J6
Vout-LP
C12 1F
R14 1Ω
R15 1Ω
U2A
LMC6482AIN
D1
1N4149
J8
Q2
ZVN3306A
LED1
LP_Mosfet
R6 1Ω
C4 1F
R7 1Ω
J13
LP_Mosfet
J7
R11 1Ω
C13 1F
R16 1Ω
R17 1Ω
LMC6482AIN
U2B
LED2
D2
1N4149
J9
Q3
ZVN3306A
HP_Mosfet
R12 1Ω
C7 1F
R13 1Ω
J5
C6 1F
C5 1F
LMC6482AIN
U1B
R10 1Ω
Vout-HP
Battery+
J10
C16 1F
C15 1F
J14
Battery+
Q1
BSP170P
J1
J3
Left-Ch
Right-Ch
U3
TLE2426IP
J2
9v Battery
LMC6482AIN
U4B
J16
HP_Mosfet