Mustafa Melih Can

Custom 1-Channel EEG Device: From PCB to Brainwaves

Abstract

This project presents the design and implementation of a custom single-channel electroencephalography (EEG) device developed from printed circuit board (PCB) fabrication to real-time brainwave visualization. The system integrates an analog front-end for amplification, filtering, and common-mode noise rejection with digital signal processing on an STM32 microcontroller. Core algorithms include biquad filters and Welch's method for spectral estimation, enabling reliable suppression of 50 Hz interference and white noise. The device was validated through the detection of alpha activity during eye closure, achieving a signal-to-noise ratio of approximately 6 dB and an alpha power increase of 5.2 times compared to the eyes-open condition. Despite limitations of its single-channel architecture and sensitivity to electrode placement, the results demonstrate that a low-cost DIY EEG can capture meaningful neural activity. The project provides a foundation for future improvements such as multi-channel expansion, wireless communication, and advanced artifact rejection techniques. It allows for a Digital Signal Processing (DSP) based EEG device that can be used in research / commercial / DIY use.

Introduction

Electroencephalography (EEG) is a non-invasive method of measuring brain activity by detecting voltage fluctuations generated by neural oscillations. It plays a crucial role in clinical diagnostics, neuroscience research, brain–computer interface (BCI) applications, and consumer neurofeedback devices. However, most commercial EEG systems are costly, complex, and designed for multi-channel use, making them less accessible for students, researchers, and hobbyists who seek to explore neural signals at a smaller scale. At the same time, do-it-yourself (DIY) EEG projects often struggle with fundamental challenges such as low signal amplitude, environmental noise, and electrode stability. To address these limitations, this project set out to design and implement a custom single-channel EEG device that is affordable yet capable of reliably detecting meaningful brainwave activity. The system integrates an analog front-end with amplification and filtering, digital signal processing on a microcontroller, and real-time visualization of EEG band powers. By focusing on detecting alpha activity during eye closure as a proof of concept, the project demonstrates how a streamlined design can bridge the gap between academic prototyping and practical EEG experimentation. The project aims to create a reliable system to detect brain signals.

System Overview

The EEG consists of three main parts in this project. The system is divided into three stages: Analog, Digital, Display. Because the main purpose of this study is to make a single Channel device and mixed signal analysis Analog and Digital parts of the project got more attention. The Analog side design flow progressed as *LTspice simulation*, *Breadboard Filter Prototyping and testing*, to *PCB design, soldering and testing*. You can see the full project pipeline on Figure 1.

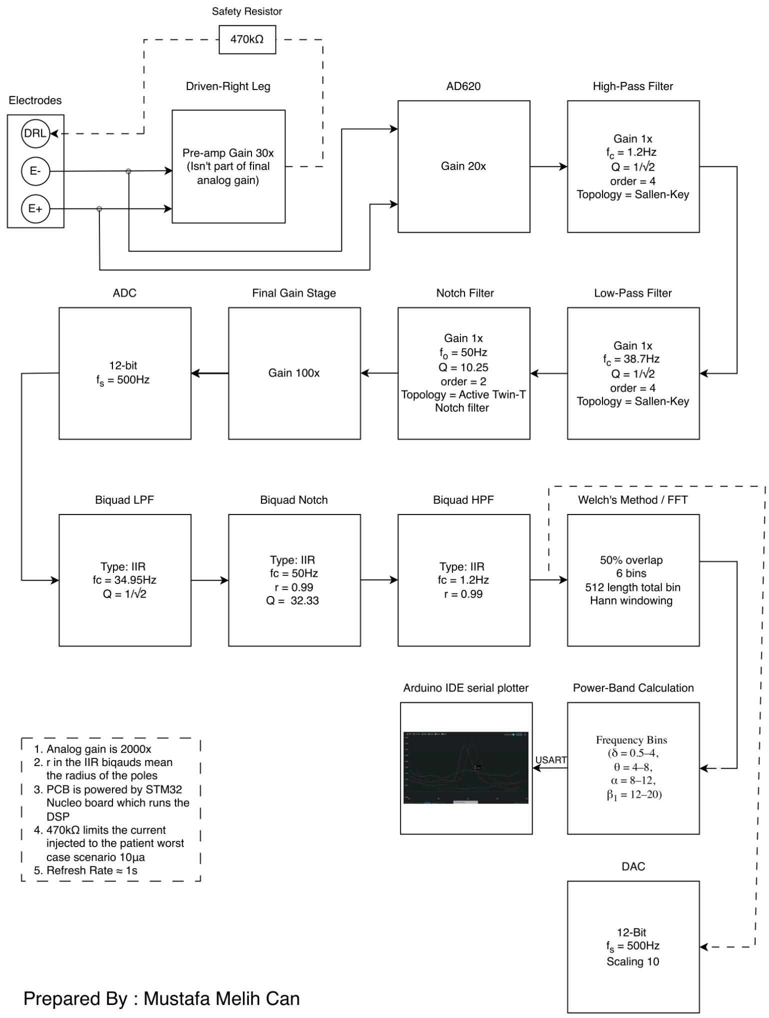


Figure 1. 1 Channel EEG System Pipeline

The analog stages "prepares the signal" by adding a total gain of 2000x and does filtering for a clearer and aliasing-free signal. According to Nyquist theorem $f_s > 2f_{max}$; I chose $f_s = 500$Hz (~10× f_{max}) to make the signals smoother on the DAC.

Hardware Design

Ok the hardware has two important roles. The first is Common Mode Noise Rejection (CMNR) that is accomplished through Driven Right Leg (DRL) and Instrumentation amplifier (INST AMP) stages. The second role is filtration and adding gain.

Common Mode Noise Rejection

The DRL is the first line of defence in us improving Signal to Noise Ratio (SNR). It basically averages the E+ and E-, adds gain, inverts it and sends it back to the body with a current limiting resistor. We basically use an adder topology since $30\frac{(x_1+x_2)}{2} = 15(x_1 + x_2)$.

Here is the DRL block and the INST AMP schematic on figure 2:

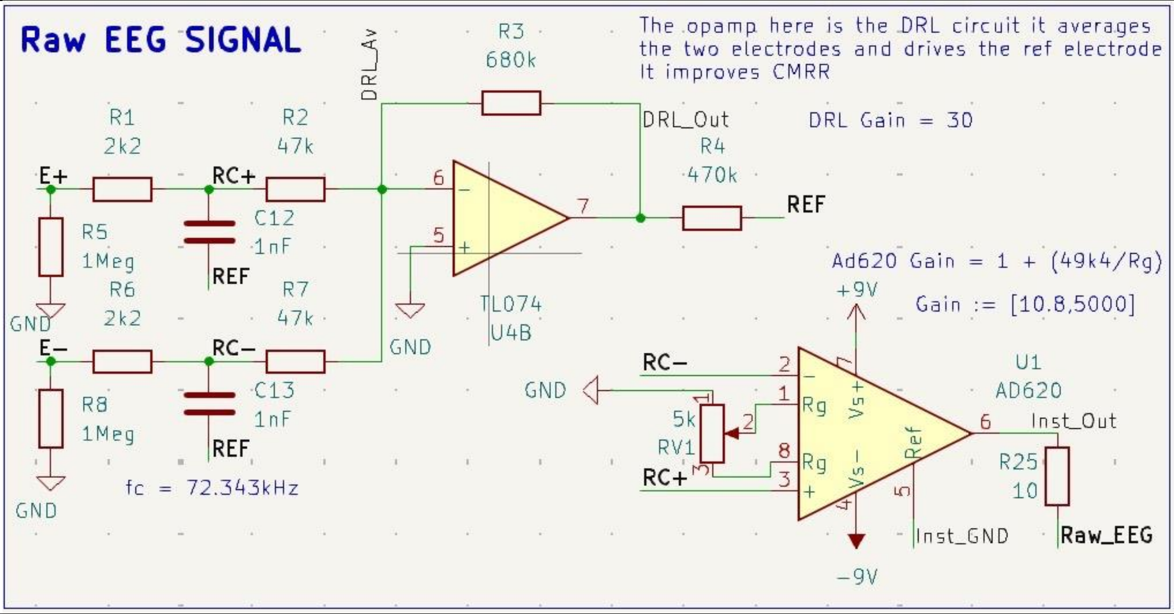


Figure 2. CMNR BLOCK

As you can see the basic DRL has some additions in design. I added a 1MΩ to the GND to make sure that the electrodes wouldn't float if they are not connected to anything else. Then the electrode stages depicted by E- and E+ connects to an RC lowpass stage with a fc at 72.34 kHz this is done to clean the signal before the signal reaches the INST AMP. And since the fc is far away we don't need to make it active and can get away with a passive one. Then we have the DRL stage with the H(s) as follows:

$$30 \cdot \frac{V_{in1}(s) + V_{in2}(s)}{2} = V_{out}(s)$$

Plus the DRL has a current limiting which limits the max current to:

$$\frac{Vmax}{R4} = \frac{5V}{470k\Omega} = 10\mu A$$

After the DRL stages we feed the RC+ and RC- stages to our INST AMP with a gain setting equation as follows:

$$Gain\ = 1 + \frac{49k4}{RV_1} : \{RV1\ \coloneqq\ [1{,}5k]\}$$

RV1 is a potentiometer (POT) so we can write the gain range of the INST AMP as follows:

$$Gain : \{[10.8{,}5000]\}$$

However, the gain i used is 20x for the inst amplifier and for your project it should get max of 100 to 200 to not swamp the EEG signal with further 50Hz hum and white noise we can add the further gain needed later. Why did we choose the gain 20x for now because alpha jumps and general eeg signals are around 10 to 20uV we if we want to make them as large as possible however we generally have a 50Hz hum that can be as large as 100mV to 250mV.

Considering these the max gain is calculated to maximise eeg signal while not saturating our opamp range $\pm 5V$; when we pick the gain around 20x that is the ideal spot hum and eeg signal volts are as follows:

$$V_{EEG} \cdot G = 20\mu V \cdot 20 = 0.1mV$$
$$V_{50Hz} \cdot G = 250mV \cdot 20 = 5V$$

As you can see the main constraint for our initial INST AMP gain stage is the environmental hum. The gain setting is set using a POT as you can see from the Figure 2. Pot works as a INT AMP gain setting.

Filtering and Gain Stage

a. High-Pass Filter (HPF)

The purpose is to remove motion, breathing, DC offset and heartbeat artifacts below 1.2Hz. 4th degree filter is decided to be made because the phase distortion by the 4th order is considered acceptable, therefore the order is not a constraint. And breathing artefacts and heartbeat artefacts wanted to be removed as efficiently as possible. So by cascading two second order HPF's i decided to get a 4th order HPF the second order HPF transfer function is as follows:

$$H_{HP}(s) = \frac{s^2}{s^2 + \frac{\omega_c}{Q}s + \omega_c^2}$$

The Sallen-Key 2nd order HPF topology is used. Please see the appendix to see the derivation of the transfer function which results in our topology seen in figure 3. and the transfer function seen is as follows:

$$H_{HP}(s) = \frac{s^2}{s^2 + \frac{c_1 + c_2}{c_1 \cdot c_2 \cdot R_2}s + \frac{1}{c_1 \cdot c_2 \cdot R_1 \cdot R_2}}$$

Please check appendix for the derivation.

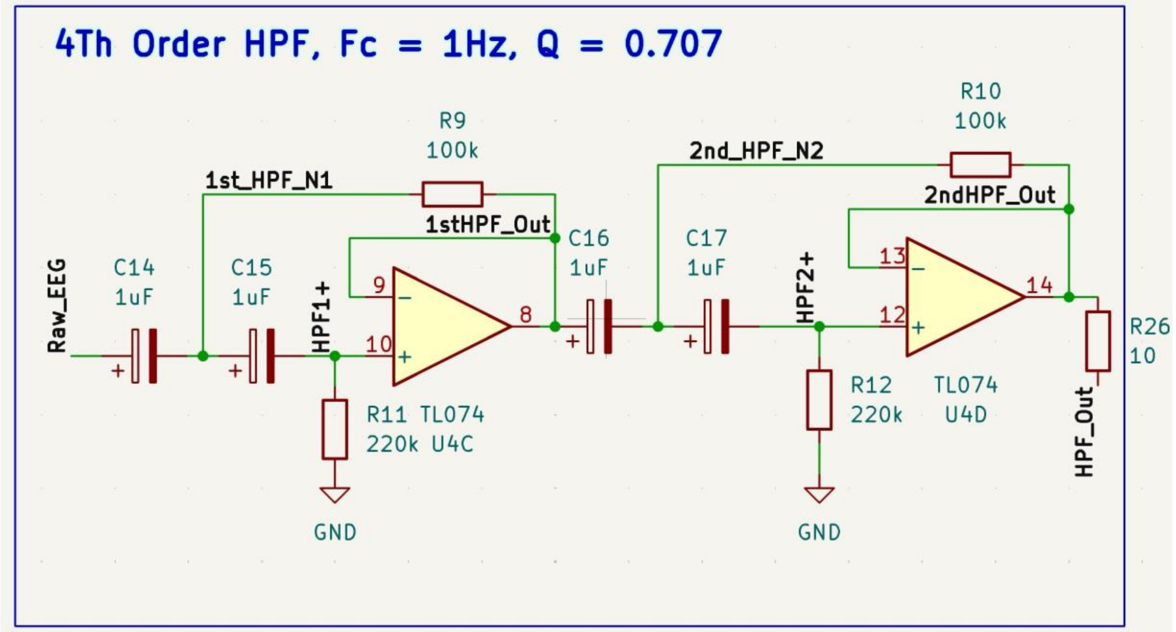


Figure 3. Sallen-Key 4th order HPF

Using the transfer previous transfer functions we get the equalities:

$$\frac{\omega_s}{Q} = \frac{c_1 + c_2}{c_1 \cdot c_2 \cdot R_2}$$
$$\omega_s^2 = \frac{1}{c_1 \cdot c_2 \cdot R_1 \cdot R_2}$$

Our desired f_c = 1Hz and $Q = \frac{1}{\sqrt{2}}$ which creates a Butterworth filter. fc is chosen as higher than usual eeg devices to remove the breathing and heartrate artefacts more efficiently since we cannot use ICA algorithm like multiple channel electrodes we need to remember that we have a single channel eeg so cleaner is always better and our purpose is to catch alpha and low beta waves so we can compromise some data lose in the delta wave range. We can choose the capacitor values arbitrarily and values that are physically easy to find and implement we can choose both capacitors as 1μF which when we seperate R_1 and R_2 we get the following equations (R_1=R_{11}=R_{12}, R_2=R_9=R_{10}, C_1=C_2=C_{14}=C_{15}=C_{16}=C_{17}):

$$R_2 = \frac{Q \cdot (c_1 + c_2)}{c_1 \cdot c_2 \cdot 2 \cdot \pi \cdot f_c} = 225k\Omega \approx 220k\Omega$$
$$R1 = \frac{1}{c1 \cdot c2 \cdot (2 \cdot \pi \cdot f_c)^2 \cdot R_2} = 112k\Omega \approx 100k\Omega$$

To verify we plug our R_1, R_2, C_1, and C_2 back to our equations to find f_s and Q. Which when we plug them in we get them as:

$$Q = 0.741,\ f_c = 1.08Hz$$

b. Low-Pass Filter (LPF)

The purpose is to remove EMG and power line noise and also to act as our antialising filter HPF for further DSP application. A similar decision made to HPF made in our LPF as well a 4th order filter with –24dB/octave is chosen. Because both EMG which is between 30Hz to 100Hz range and 50Hz hum is are strong noise sources which we want to eliminate as strongly as possible. The topology is at Figure 4.

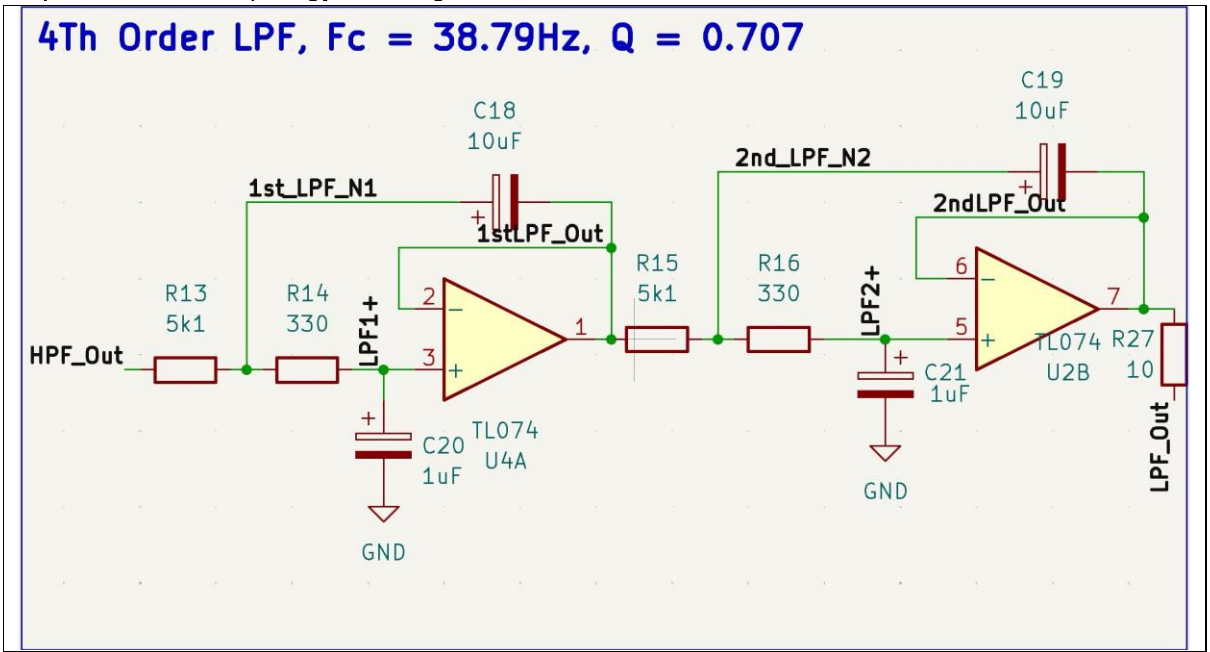


Figure 4. Sallen-Key 4th order LPF

A similar design process to HPF is followed therefore the details are omitted. However the final Q and f_s are as follows:

$$Q = 0.755,\ f_s = 38.79Hz$$

c. Notch Filter

The purpose is to remove power line noise in Turkey with standard 50Hz. An active Twin-T notch filter is used for our design. Because we can set the Q using a simple voltage divider and the Twin-T topology doesn't determine the Q and we can use them to control the center frequency also with this topology we only use 2 opamps. The topology can be seen in Figure 5.

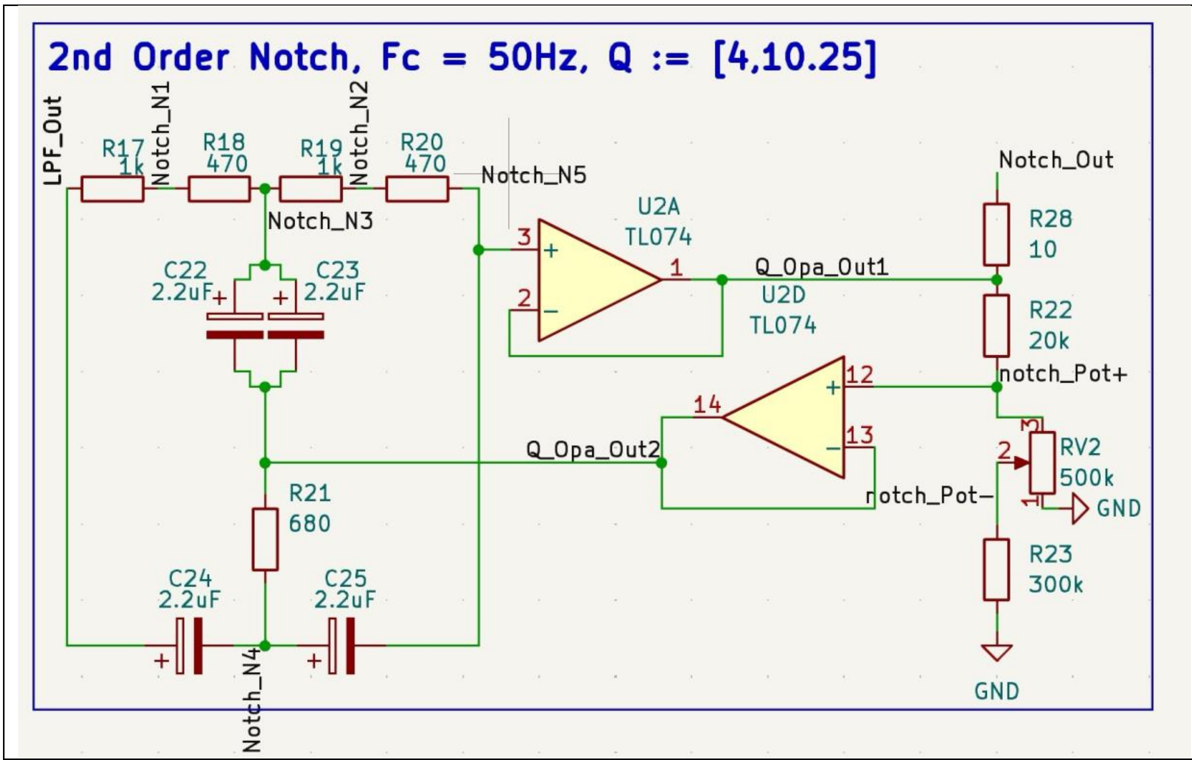


Figure 5. Active Twin-T Notch Filter

For context of my circuit lets the set values are as follows: $R_{17} + R_{18} = R_{19} + R_{20} = R$, $R_{21} = \frac{R}{2}$, $C_{22} = C_{23} = C_{24} = C_{25} = C$

The following contains the transfer function of the Notch Filter:

$$H_N(s) = \frac{s^2 + \omega_0^2}{s^2 + s\frac{\omega_0}{Q} + \omega_0^2} = \frac{s^2 + \left(\frac{1}{RC}\right)^2}{s^2 + s\frac{4 \cdot (1-K)}{RC} + \left(\frac{1}{RC}\right)}$$

Where:

$$K = \frac{RV2 + R23}{R22 + RV2 + R23}$$

$$Q = \frac{1}{4(1-K)} so,\ Q_{min} = 4,\ Q_{Max} = 10.25$$

Since RV2 is a potentiometer, it was used to tune the effective gain range [4,10.25], so that it can attenuate 50Hz while not cause ringing in the circuit that is why it is capped at 10.25 value.

$$f_0 = \frac{1}{2\pi RC} = 50Hz$$

$$f_0 = \frac{1}{2\pi RC} = \frac{1}{2\pi \cdot 1.470 \cdot 10^3 \cdot 2.2 \cdot 10^{-6}} = 49.21Hz \approx 50Hz$$

Be careful when increasing the Q on your project with the potentiometer because as you can see the center frequency in our case is around 49.21Hz and if we increase the Q very high such as 30 40 we might make the notch very steep and deep causing our filter to miss the center frequency which is 50Hz in our case.

When we simulate all of the filters in LTspice we get the following filter response:

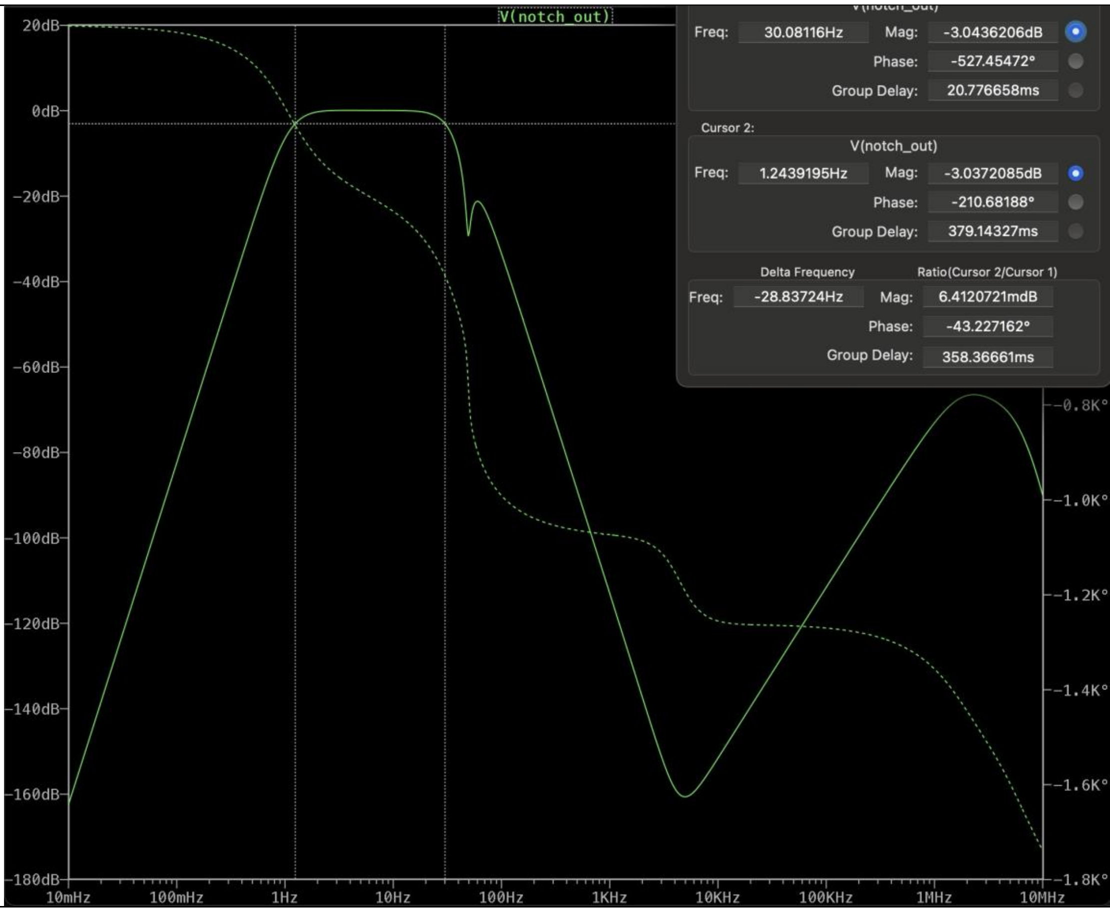


Figure 6. Cut off of our cascaded LPF and HPF (Bandpass range := [1.2Hz, 30Hz])

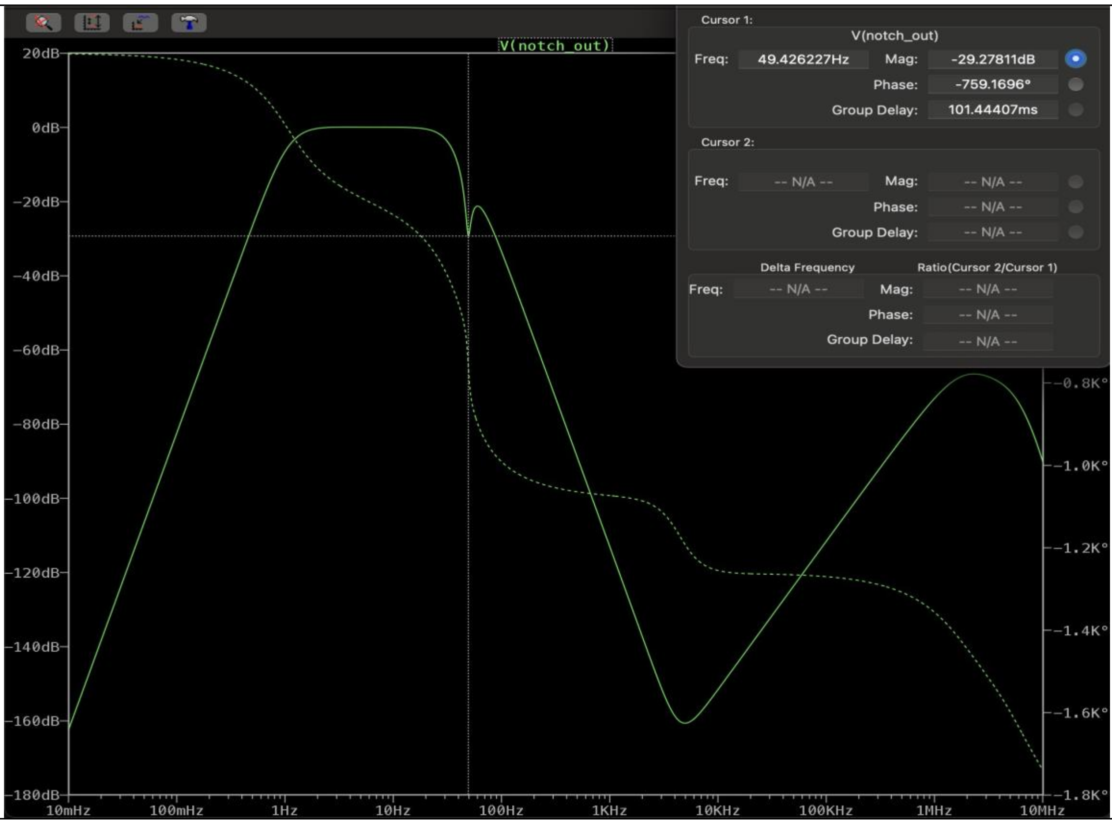


Figure 7. Center Frequency and attenuation of –29dB at 49.42Hz mains hum

ç. Final Gain Stage

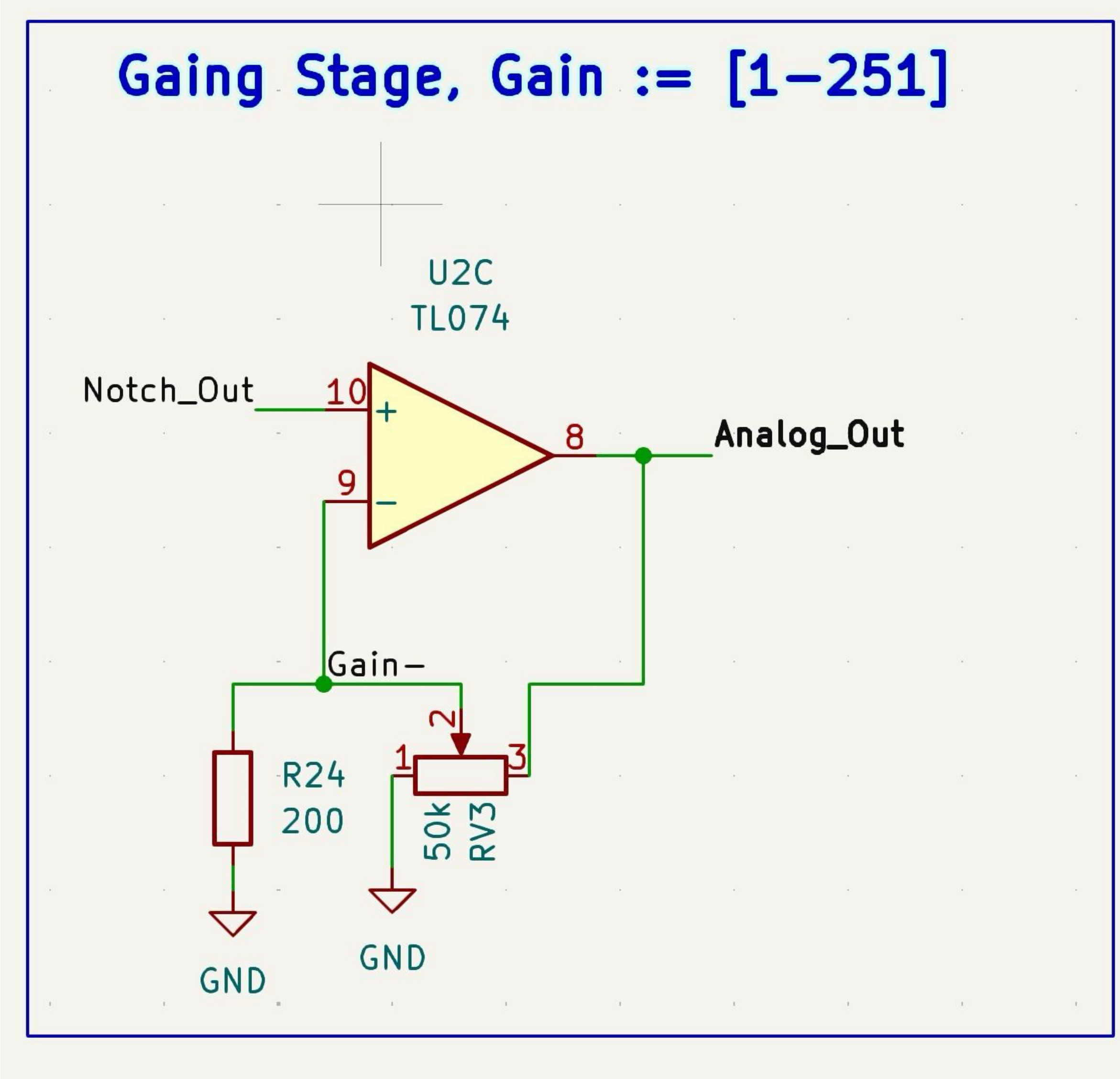


Figure 8. Final Gain Stage

The Gain function is as follows:

$$Gain = 1 + \frac{Rv3}{R2},\ Rv3 \in [0{,}50k\Omega],\ R2 = 200\Omega$$

This final stage gives a gain setting with a potentiometer between 1x to 250x. Allowing me to set the output of the PCB between the ADC optimum range which is between 0V and 3.3V. This gain stage allows for a DC offset adder as well because the PCB output is generally is around 200mV so we can easily increase the gain to add a 1.65V DC offset that will make the output of the PCB optimum for the ADC stage. If yours doesn’t come out like that you can add a DCshifter which you can find the design on the appendix section.

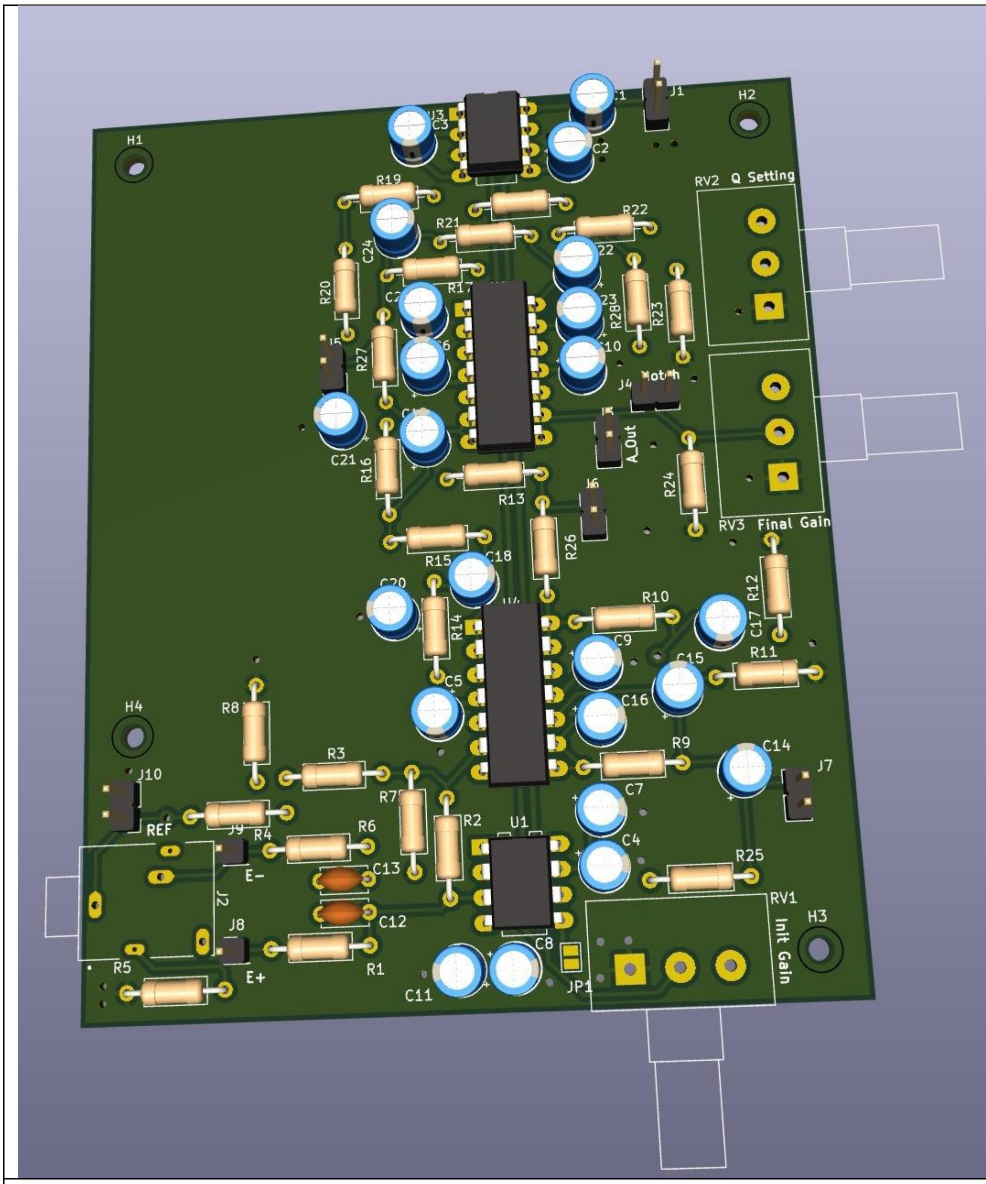


Figure 9. Final Assembled PCB of the EEG Analog front-end

Software Design

The software in this project completes the rest of the functionalities of the EEG. There are two functions that the digital side accomplish:

1) Digital Signal Processing
    a. Welch's method (Reducing white noise)
    b. ARMA Biquad Filters (Remove hum, artefacts)
2) Displaying EEG Frequency Bands
    a. Do Fast Fourier Transform (FFT) and calculate Power Spectral Densiy (PSD) to show different frequency ranges (Alpha, Beta, etc.)
    b. Data Display: Using USART to show data on a serial display

An appropriate Sampling frequency 500Hz is chosen. It is 15x the max frequency (35Hz) that our system will be interested. And is well over the nyquist theorems minimum sampling frequency ($f_s > 2f_{max}$).

Digital Signal Processing

A. Biquad Filters

A biquad filter basically means a two zero, two pole cascading second order (SOS) filters. The SOSs can be cascaded without making the whole system unstable they act as the building blocks of our larger complicated filtering systems. In this instance we want to create the same filters we created in the analog side to further clean our signal digitally. A biquad equation has the form:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{a_0 + a_1 z^{-1} + a_2 z^{-2}}$$

The form then can be turned into the difference equation using inverse Z transform:

$$y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] - a_1 y[n-1] - a_2 y[n-2].$$

The difference equaion results in the following figure configuration:

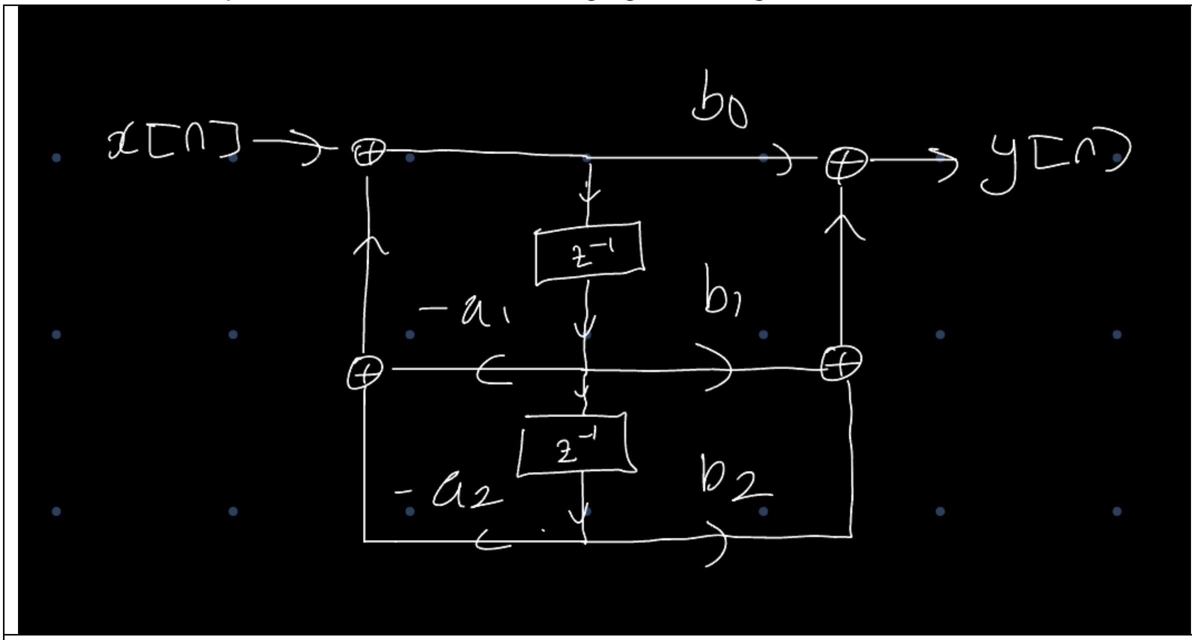


Figure 10. Cascade form of ARMA Biquad Filter Block Diagram

By using CMSIS-DSP library of the ARM cortex MCU’s when we get the coefficents of our filters we basically can plug them in the CMSIS functions. Which will do the multiplying and shifting of our buffer for us which depicted on the Figure 10.

A1. Biquad Notch Filter

To get a notch filter we are going to put zeros at 50Hz and poles around 50Hz with the radius r1. Using our sampling frequency, to calculate that region we use the formula:

$$\omega = \Omega T;$$

Where Ω means the analog angular frequency, ω is the digital angular frequency and, T is our sampling period.

Assuming we want to place our zeros and poles at 50Hz then to find where are we going to place them between [-π, π] which is digital angular frequency range:

$$\omega_0 = \Omega_0 T = 2\pi \cdot \frac{f_0}{f_s} = 2\pi \cdot \frac{50}{500} = \frac{\pi}{5}$$

Here we see that our poles/zeros should be on the π/5. However, since our filter needs real coefficients we are going to put them at $\omega_0 = \frac{\pi}{5}, -\frac{\pi}{5}$ on the complex plane. Plus we going to place the zeros right on the unit circle making a very deep notch and placing the pole near the zero but not at unit circle. This is done to dampen the notch effect and control the notch by controlling the radius of our poles. Lets see the mathatical formulation:

$$H_N(z) = \frac{\left(1 - z^{-1}e^{j\frac{\pi}{5}}\right)\left(1 - z^{-1}e^{-j\frac{\pi}{5}}\right)}{\left(1 - z^{-1}r_1e^{j\frac{\pi}{5}}\right)\left(1 - z^{-1}r_1e^{-j\frac{\pi}{5}}\right)}$$

$$H_N(z) = \frac{1 - 2\cos\left(\frac{\pi}{5}\right)z^{-1} + z^{-2}}{1 - 2r_1\cos\left(\frac{\pi}{5}\right)z^{-1} + r_1^2 z^{-2}}$$

Therefore our notch equation is completed.

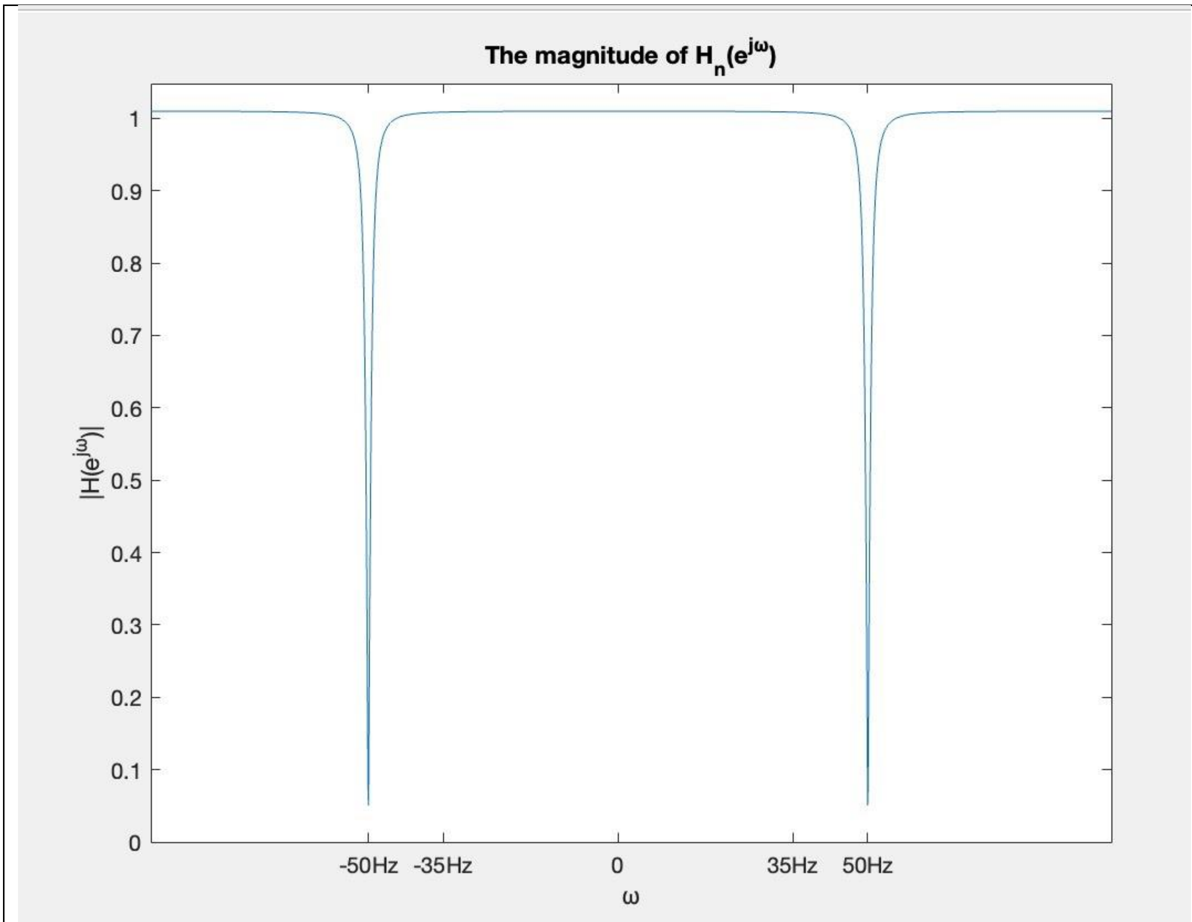


Figure 11. Notch Filter Response |H(e^jw)|

When we try some r1 values and simulate the results on Matlab we get an r1 value of 0.99. Which we can see the results on Figure 11. and the pole-zero plot on figure 12.

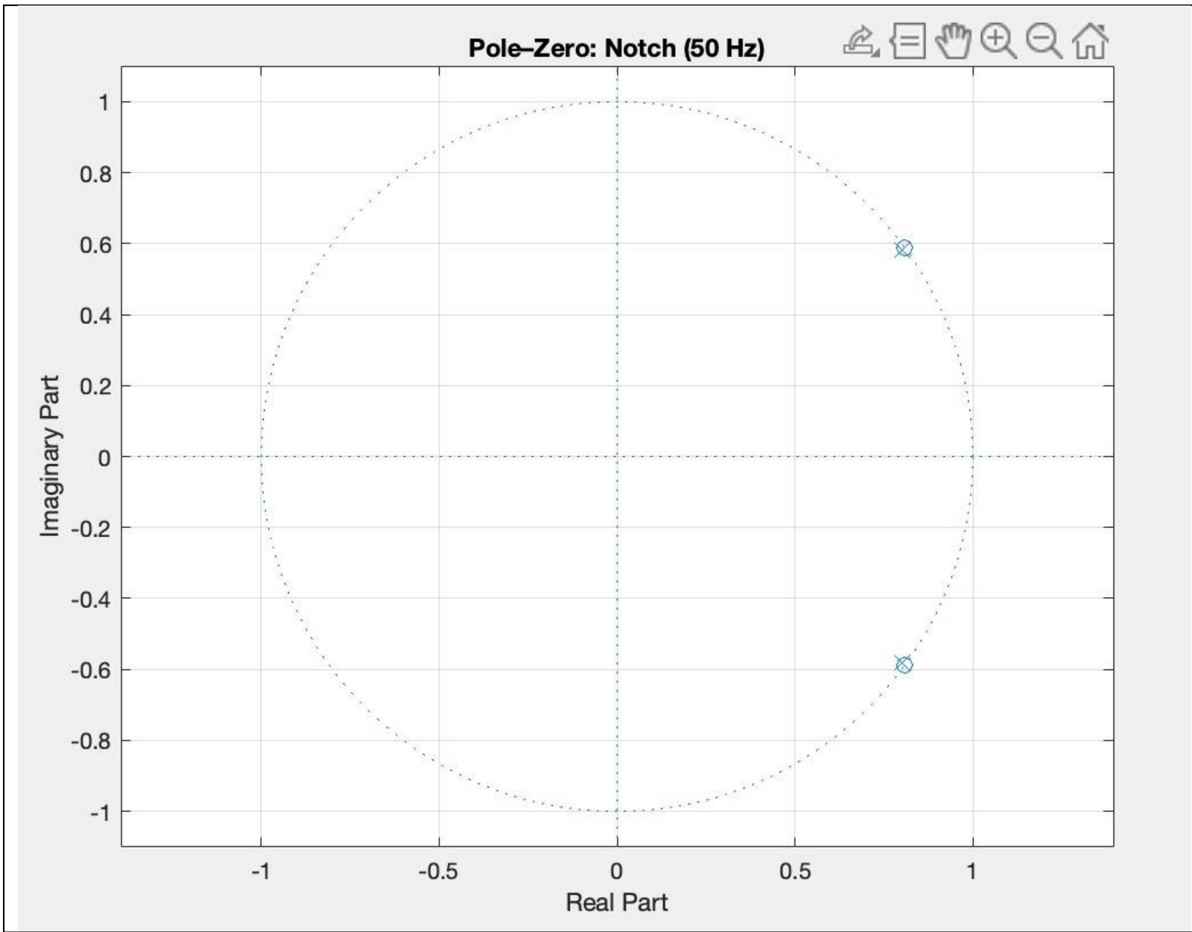


Figure 12. Zero-Pole Plot of Notch Filter

The final coefficients are:

$$b_{notch} = \left[1,\ -2\cdot r1\cos\left(\frac{\pi}{5}\right), 1\right]$$

$$a_{notch} = \left[1, -2r_1\cos\left(\frac{\pi}{5}\right), r_1^2\right]$$

A2. Biquad HPF

Standard HPF and LPF biquad designs are used. The HPF is designed as:

$$H_{HP}(z) = \frac{(1-z^{-1})^2}{(1-r_2 z^{-1})^2}$$

$$H_{HP}(z) = \frac{1-2z^{-1}+z^{-2}}{1-2r_2 z^{-1}+r_2^2 z^{-2}}$$

And:

$$r_2 = \exp\left(-\frac{2\pi f_{HP}}{f_s}\right) = \exp\left(-\frac{2\pi 1.2}{500}\right) = 0.985$$

Here is the Pole-Zero plot of the filter:

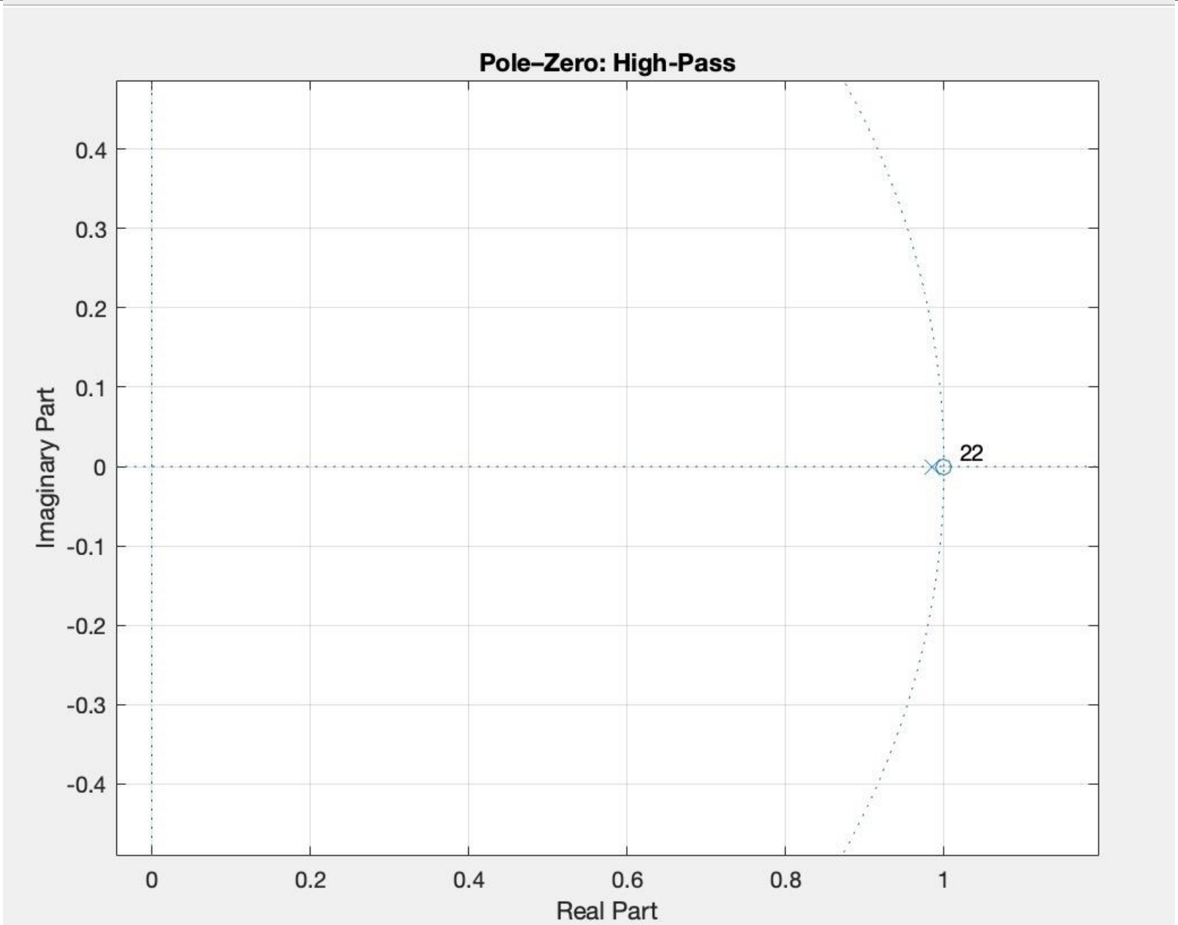


Figure 13. The pole-zero plot of HPF

As you can see from the figure 13. our biquad pole and zeros are on the real axis which is consistent with our desgin choices and our application. Now we must find the HPF coefficients therefore the coefficients for the HPF is found:

$$b_{HPF} = [1,\ -2,\ 1];$$
$$a_{HPF} = [1,\ -1.980,\ 0.9801];$$

With these coefficients we get the following filter response:

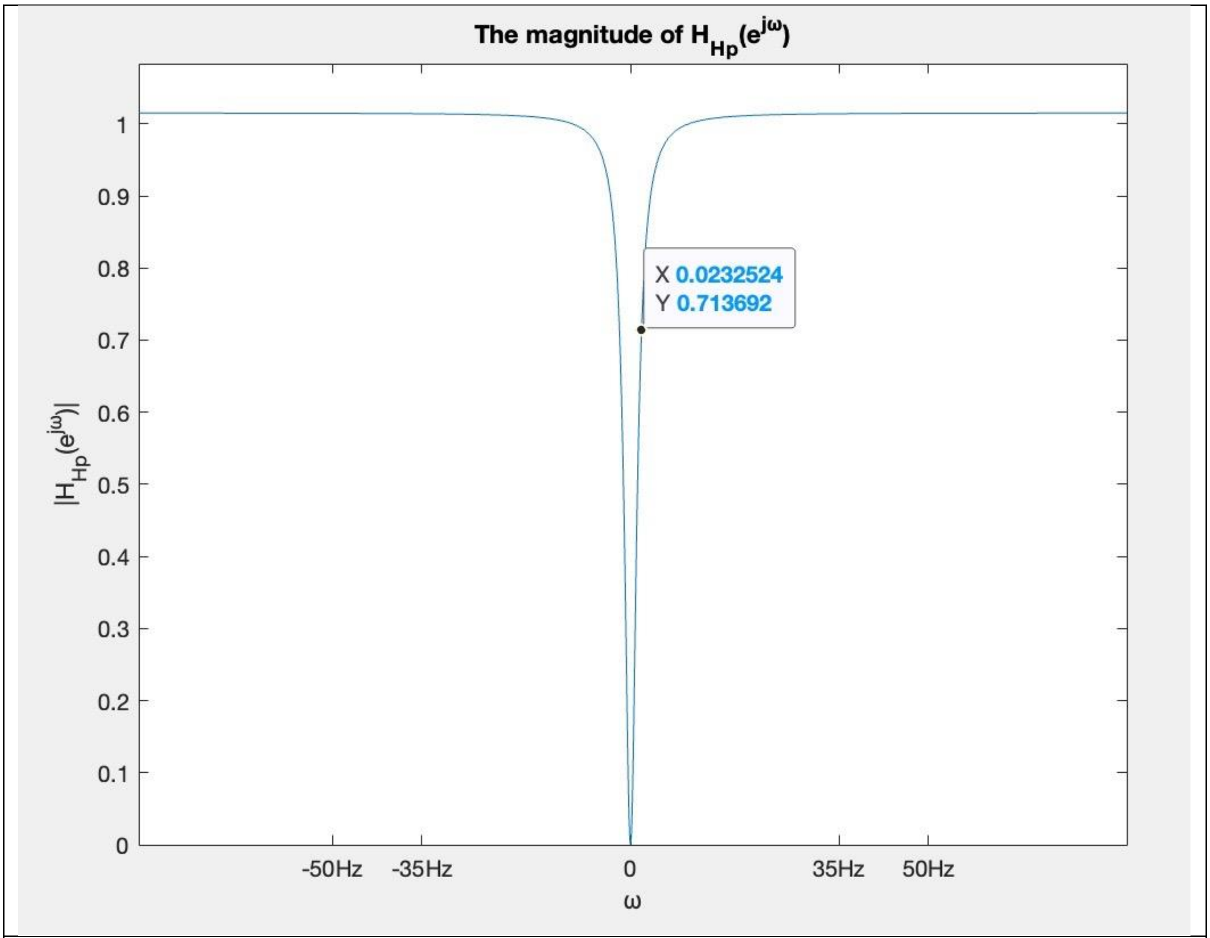


Figure 14. The frequency response of ARMA HPF

A3. Biquad LPF

A bilinear approach is used to LPF design. Here is the analog 2nd-order butterworth LPF:

$$H_{HP}(s) = \frac{\Omega_c^2}{s^2+\frac{\Omega_c}{Q}s+\Omega_c^2},\ Q = \frac{1}{\sqrt{2}}, \Omega_c = 2\pi f_c$$

When we do the bilinear mapping we get:

$$\Omega_c = \frac{2}{T}\tan\left(\frac{\omega_c}{2}\right),\ s = \frac{2}{T}\cdot\frac{1-z^{-1}}{1+z^{-1}}, T = \frac{1}{f_s}, \omega_c = \frac{2\pi f_c}{f_s}$$

Then we expand to get the final z equation as:

$$HLPF(z) = \frac{0.0366 +\ 0.0731z^{-1} + 0.0366z^{-2}}{1 + \ -1.3909z^{-1} + 0.5372z^{-2}}$$

Using these coefficients when we plot the pole-zero diagram we get:

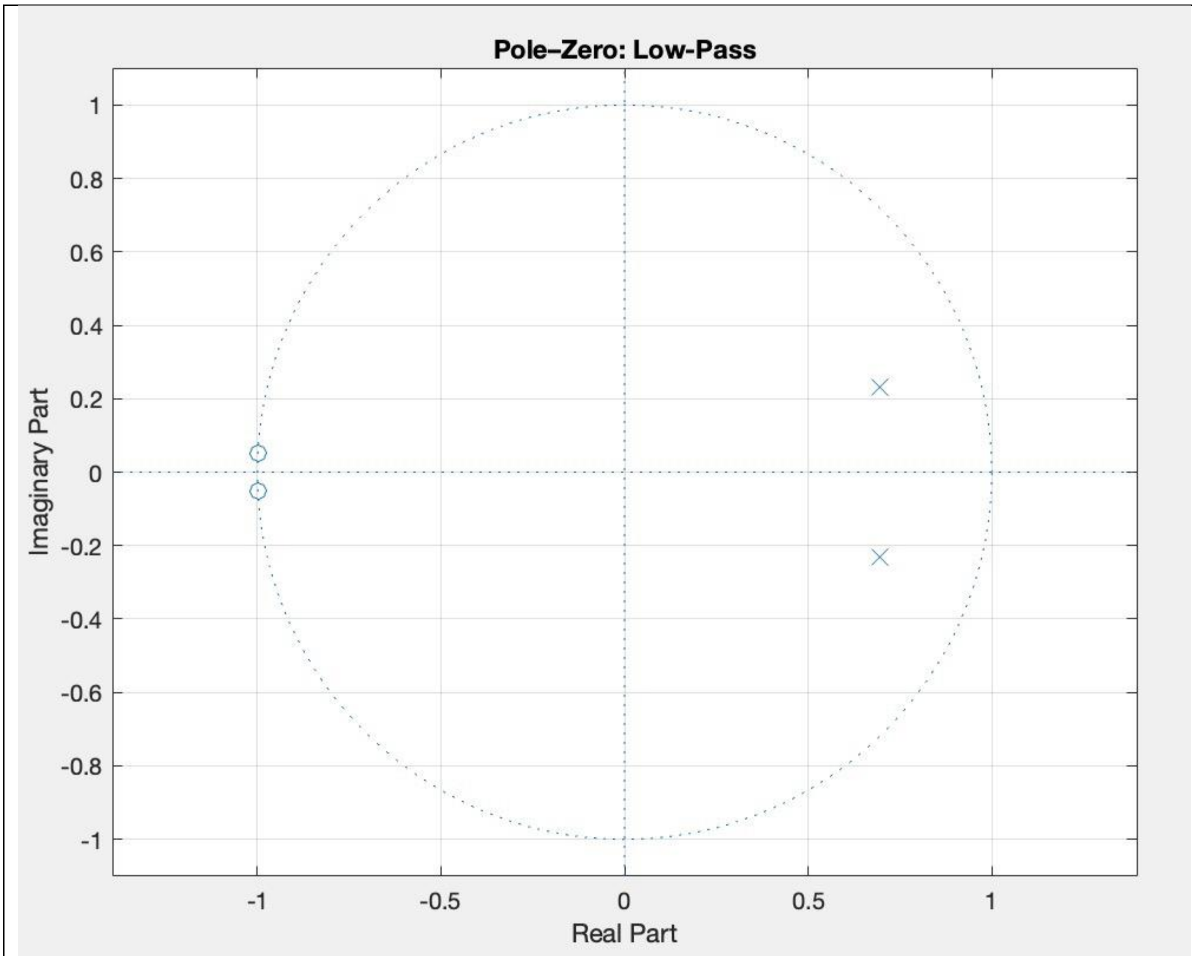


Figure 15. The pole-zero plot of LPF

As you can see from Figure 15. that our poles are on the unit circle at the Nyquist frequency and our zeros have a phase of 0.32 radians and radius of 0.733 which means it is on the 25.46 Hz frequency in the Ω (analog angular frequency) domain. This is consistent with our Bilinear transform design.

Therefore our coefficients are:

$$b_{LPF} = [0.0366,\ 0.0731,\ 0.0366];$$
$$a_{LPF} = [1, -1.3909, 0.5372];$$

Here is the frequency response of our LPF becomes:

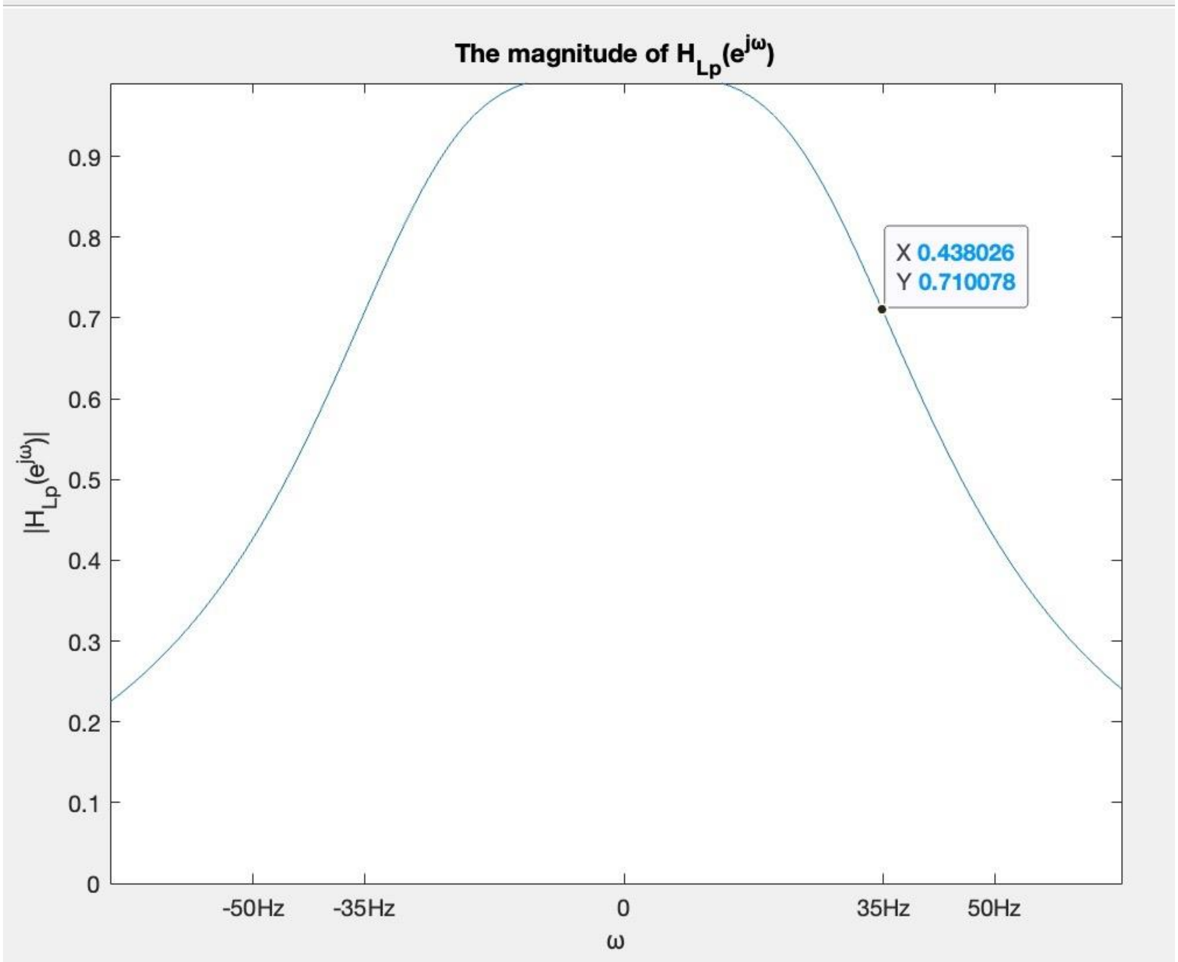


Figure 16. Frequency Response of LPF

You can check the github for the full code implementation of these filters. Also here are the final plots for the all three cascaded ARMA filters:

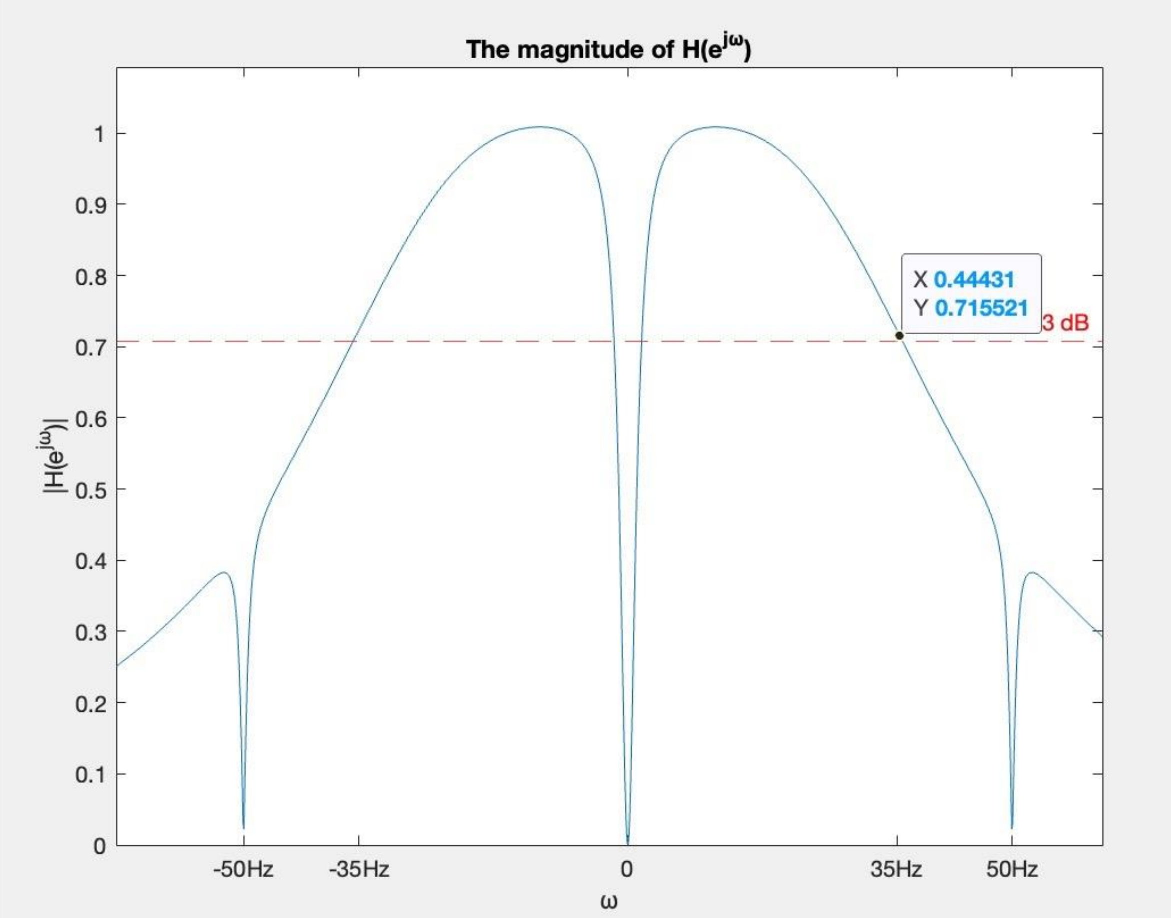


Figure 17. Frequency Response of the full ARMA Chain

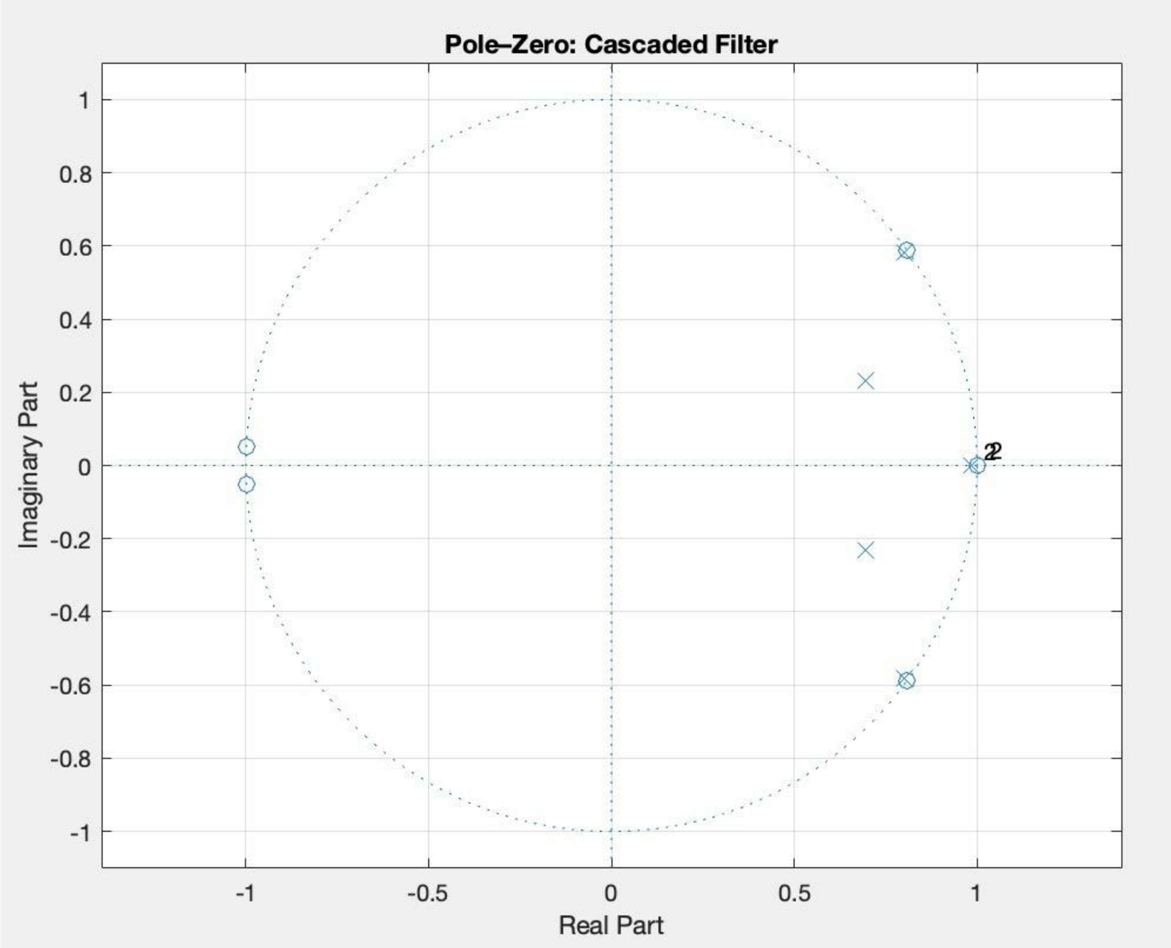


Figure 18. Pole-zero plot of the full ARMA Chain

B. Welch's Method

The Welch's method basically prepares the data for a FFT application the purpose of it is to remove white noise probabilistically. It basically takes the average of FFT windows allowing for us to probabilisticly construct the time correlated data such as alpha jumps when a blink occurs and destruct the time unccorralted such as white noise. So in order to make this happen we are going to choose a bin size which in our application is best set at 512 which will approoximately give 1Hz bins which is enough resolution to calculate Power Spectral Density (PSD) across different bands.

$$\frac{f_s}{Bin\ size} = \frac{500}{512} = 0.976Hz\ resolution$$

The algorithm is basically is 6 steps:

1. ontinuously shift incoming samples into a 512-sample buffer
2. From this buffer, take the first 256-sample segment (with 50% overlap) as the analysis window
3. Apply hemming window to the segment
4. Compute a 256-point FFT to obtain the frequency spectrum
5. If this the third 256-point FFT (i.e., covering the full 512-sample buffer), shift to the next buffer and repeat.
6. Average with all 6 FFT blocks therefore reducing variance (basically white noise)

You can check github for the full implementation of FFT/Welch's algorithm.

Displaying EEG Frequency Bands

A. FFT and PSD

As we have our cleaned data we are going to take the FFT of our signal. After filtering, the EEG signal is transformed into the frequency domain using the Fast Fourier Transform (FFT). The FFT efficiently computes the Discrete Fourier Transform (DFT):

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\cdot\frac{2\pi}{N}kn},\ k = 0,1, \ldots N-1$$

In Practice the algorithm goes like this:

- The EEG stream is divided into 256-sample segments.
- Each segment is windowed (Hamming) to reduce spectral leakage.
- A 256-point FFT is computed using the CMSIS-DSP library on the STM32 microcontroller.
- The squared magnitude of the FFT gives the power spectrum of that segment.
- By averaging multiple overlapping spectra (Welch's method), the Power Spectral Density (PSD) is obtained.
- The PSD is then integrated over standard EEG frequency bands (delta, theta, alpha, beta) to calculate EEG band powers. Which basically means adding the subsiquent indeces of our FFT output buffer and squaring it.

B. Data Display

Once the EEG signal is filtered and transformed into the frequency domain, the final task is to make the results observable in real time. In this project, the STM32 sends the calculated band powers (Delta, Theta, Alpha, Beta) to a PC via USART serial communication. The arduino IDE serial plotter receives this powers and plots them continuously every second, allowing visual tracking of band activity. This setup makes it easy to observe alpha jumps when the subject closes their eyes, while still providing the option to stream raw time-domain signals for debugging. This is done by using DAC at the same sampling frequency and writing our filtered data on the display out of the board therefore we can show it on our oscilloscope. You can see that on Figure 19. The blue line is the DAC output and yellow is the input to our DSP system. The approach is modular and can be extended to more channels or alternative visualizations (e.g., bar graphs, scrolling waveforms, or spectrograms).

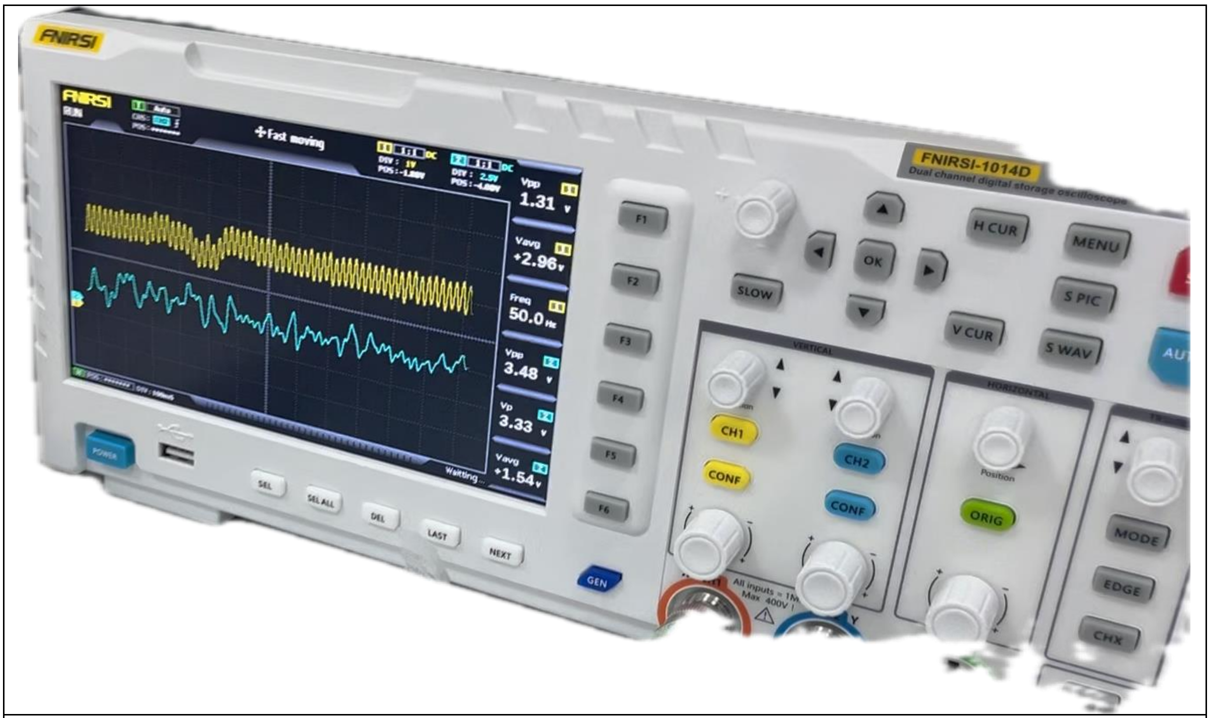


Figure 19. The oscilloscope output

Results

Our main purpose is to see an Alpha Jump so that we can verify we are getting signals from the brain. Therefore lets start the results section with an alpha jump result. You can see on figure 20. an alpha jump that is 4x the normal noise levels:

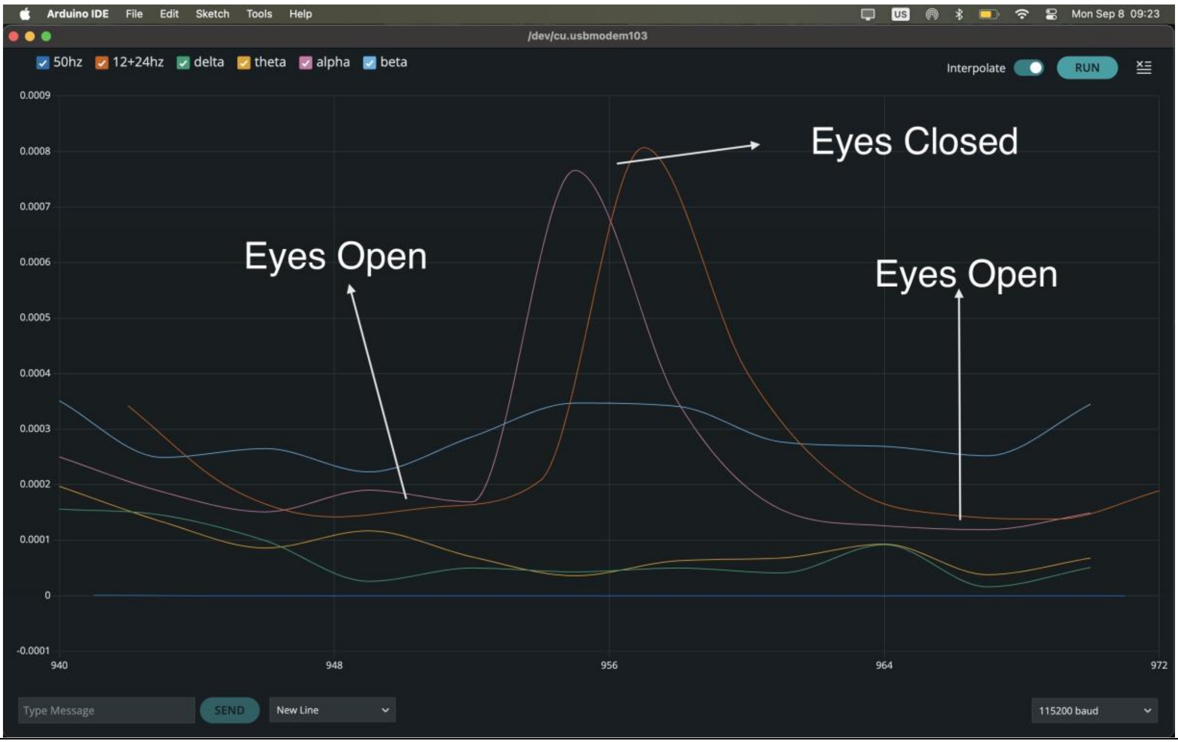


Figure 20. Eyes closed state resulting in alpha jump 4x the normal noise level

The figure 20. shows an Occular test done on O2 site on the brain of the patient. It is expected to see an alpha jump when the patient closes their eyes resulting in a jump in the alpha PSD. As you can see the resulting jump is 4x which is very significant. (see the end of the results section to see SNR and ALPHA ratios)

Further real time results for the system can be seen on the video:
https://youtu.be/_DtyJymNVPs

Once soldered and tested the final PCB was assembled Figure 21. shows the completed analog front-end board.

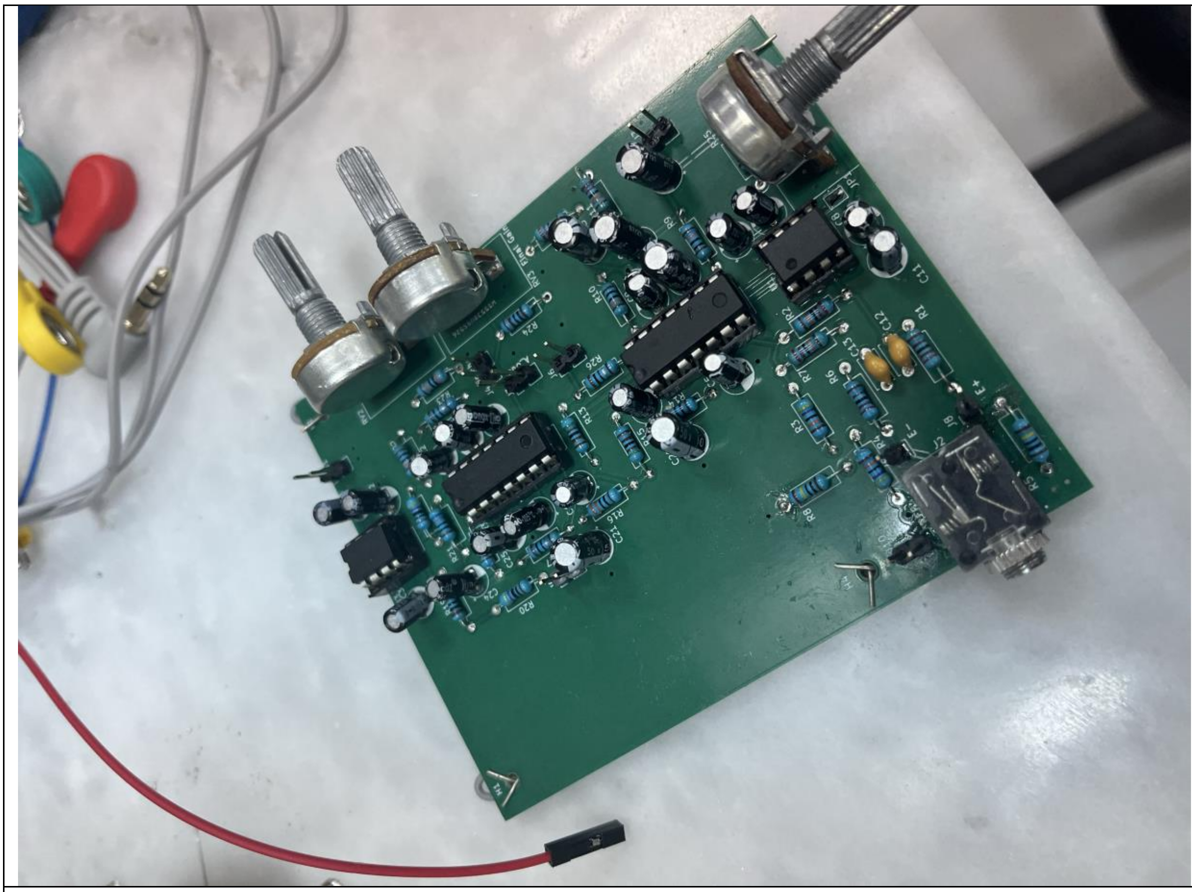


Figure 21. Fully assembled PCB

To improve SNR, the electrodes were shielded and twisted to minimize external interference. The electrodes shielding has a ground connection that also is connected to the PCB as well. The Brown wire shown in Figure 23. works as our shielding ground.

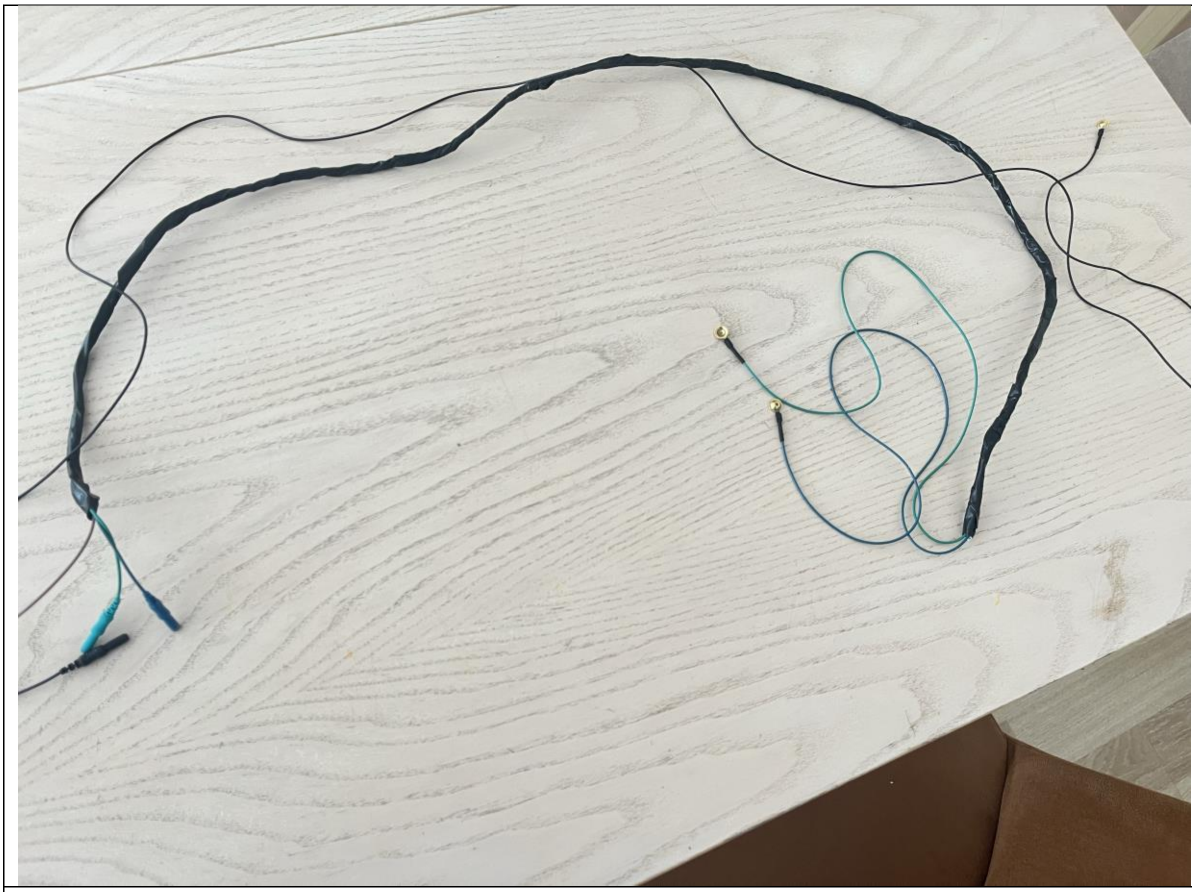


Figure 22. Faraday Shielded + Twisted EEG electrodes

Figure 23. Electrode Board Connections (Brown = Shield Ground Connection To PCB)

The full system assembled and used is shown in Figure 24.

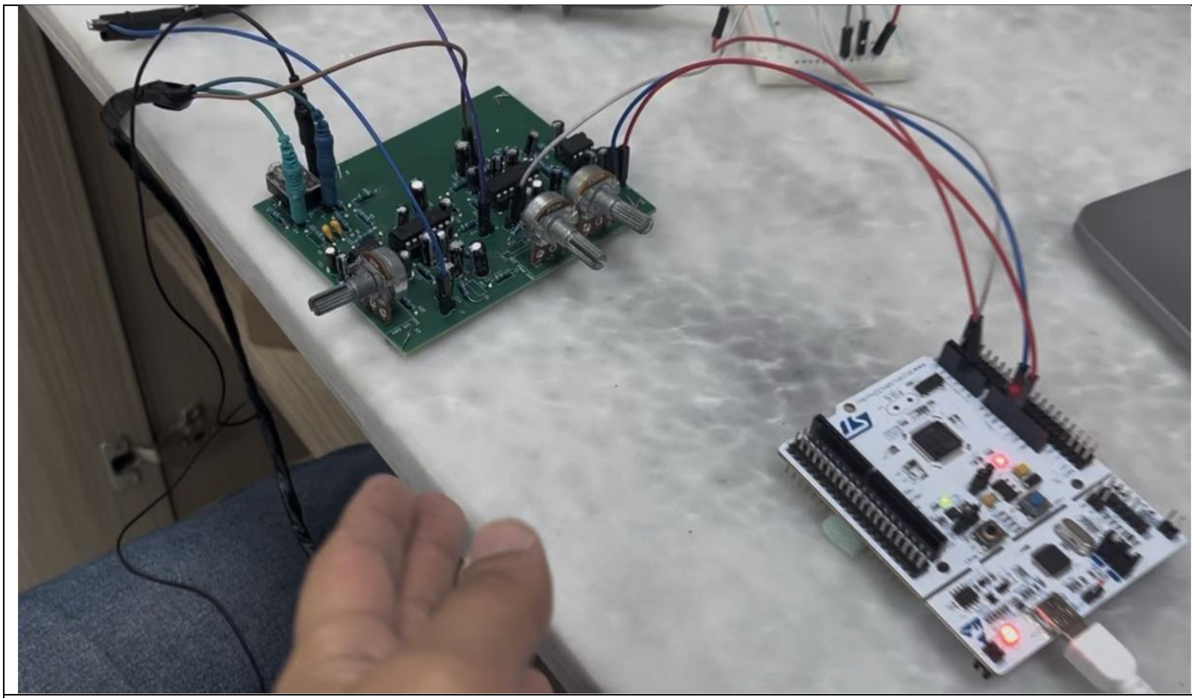


Figure 24. Assembled Full EEG pipeline (PCB+STM32)

Electrode placement followed the standard O2 reference configuration, with the DRL electrode on the forehead (Figure 25).

Figure 25. The electrodes placed (Green = Active O2, Blue = Ref Mastoid Bone, Black = DRL/GND Fz/Forehead)

To also see the real time EEG performance the oscilloscope is connected displaying both outputs of Analog and Digital side on Figure 26. The blue line on the oscilloscpe is the output of the full EEG pipeline. It is the conventional EEG signal that everybody is used to seeing amplified 10 times.

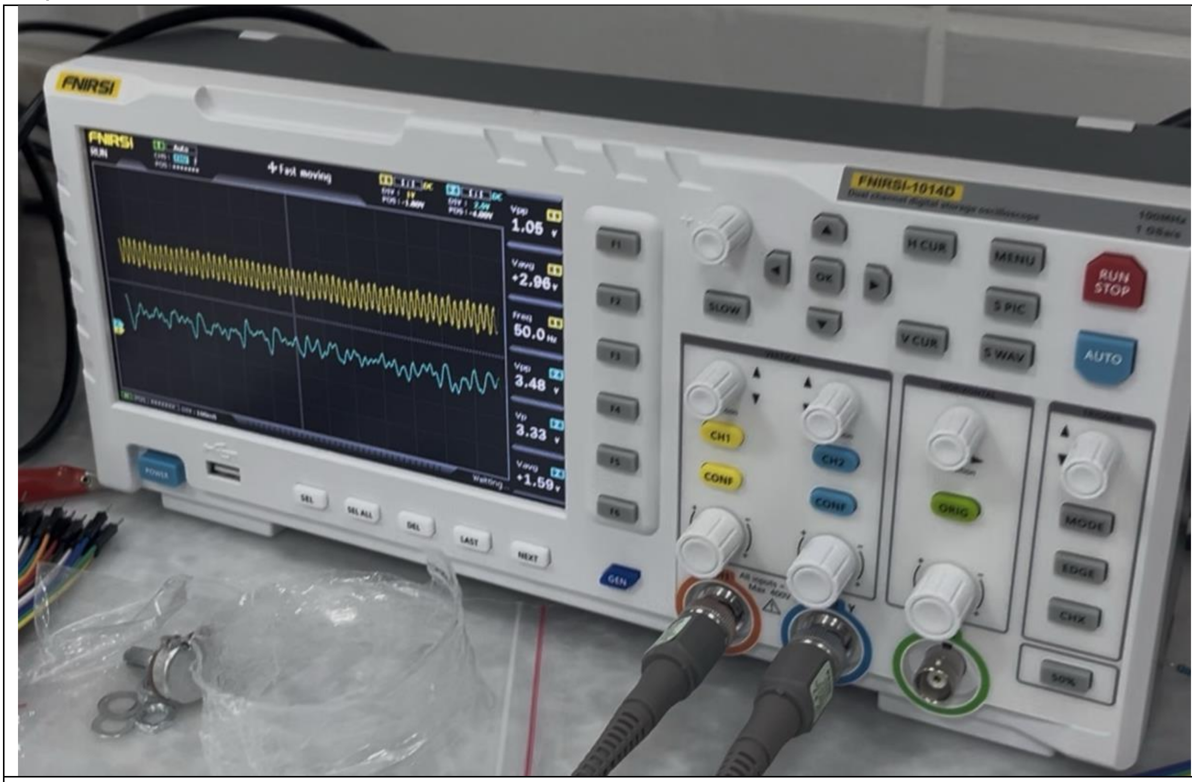


Figure 26 . The oscilloscope showing PCB output = Yellow, EEG signal = Blue

In addition to alpha band detection, the system captured motion-related artefacts, as shown in the following figures. All bands showed transient increases during movement, especially in delta and theta. Which can be seen on Figures 26-27.

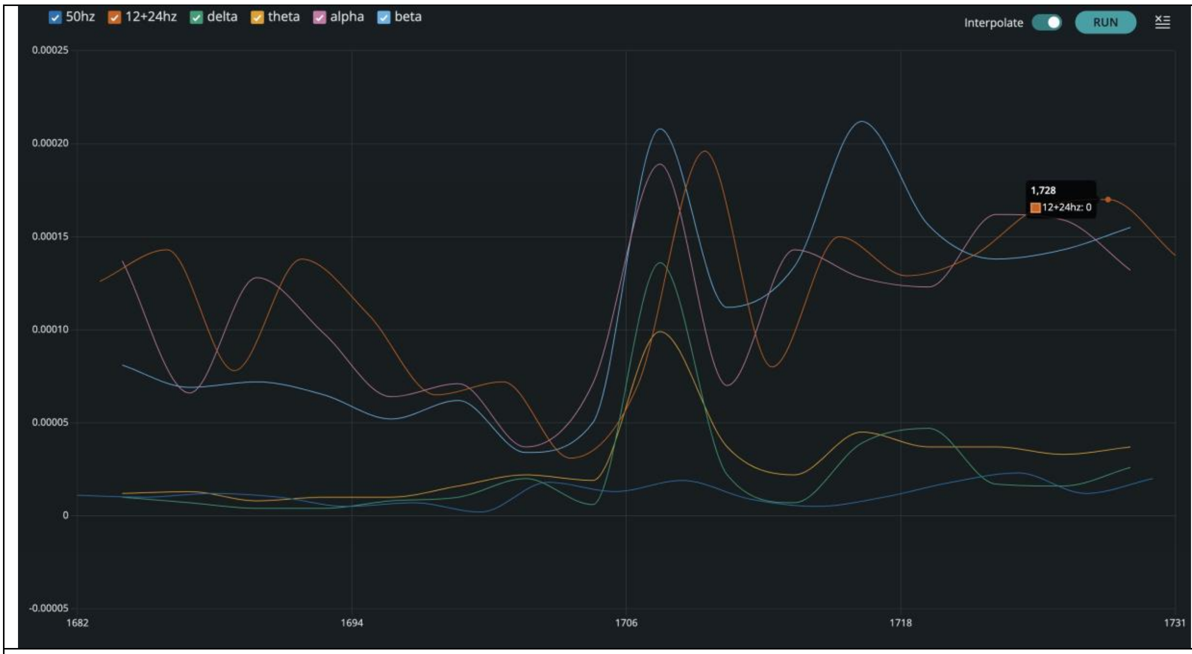


Figure 27. The movement artefact shown by all bands having a 3-4x jump

Figure 28. Jump in delta and theta bands due to movement artefacts

Lets calculate the SNR ratio for the whole system. The values are gotten from the Figure 20.:

$$P_{signal} = 0.8mWatt$$
$$P_{signal} = 0.2mWatt$$

$$SNR = 10\log\left(\frac{P_{signal}}{P_{noise}}\right) = 10\log(4) = 6.02dB \approx 6dB$$

The SNR of 14dB is really good for a DIY system like ours which is in the

Lets calculate the Alpha jump ratio using the observed results in the youtube video which are:

$$P_{eyes\ closed} = 0.45mWatt\ ,\ P_{eyes\ open} = 0.08mV$$

$$Alpha\ Ratio = \frac{P_{eyes\ closed}}{P_{eyes\ open}} = \frac{0.45}{0.08} = 5.6\ times\ jump$$

As we can see from our calculations that the EEG system is successful in measuring real EEG data from the brain. Typical commercial non-invasive EEG devices have SNR ratio of 5 to 14 dB which for a single channel custom EEG project 6dB SNR ratio therefore shows a successfully built EEG.

Problems Faced and Solutions

There were multiple problems faced during the creation of this project the table will highlight those problems that you might face as well and their solutions:

THE PROBLEM	THE SOLUTION

The output showed 50Hz signals which you can see the resulting the graph on figure 29.	Added a Digital notch + Increased the gain of DRL circuit + increased the Q of the analog notch filter using the Potentiometer
The SNR was still high after these connections which resulted in the Figure 30.	Shielded the electrodes and grounded the the output which can be seen on figures 22-23 + started using ECG electrodes and swtiched to EEG electrodes which significantly improved SNR
White noise floor was too high resulting in no alpha jumps being observed	Implemented The welch's method which is a probabilstic algorithm reducing the time uncorralated noise which is white noise in our case.
Electrode connections had high impedance which also causing our electrodes to act as antennas and pick up more 50Hz noise hum	Used EEG electrodes instead of ECG electrodes + used EEG conductive gel
Electrode placement was not 100% accurate causing the signals to be not strong enough	Got a measuring tape and used the 10-20 EEG electrode placement system convention used in hospitals which you can see on Figure 30.

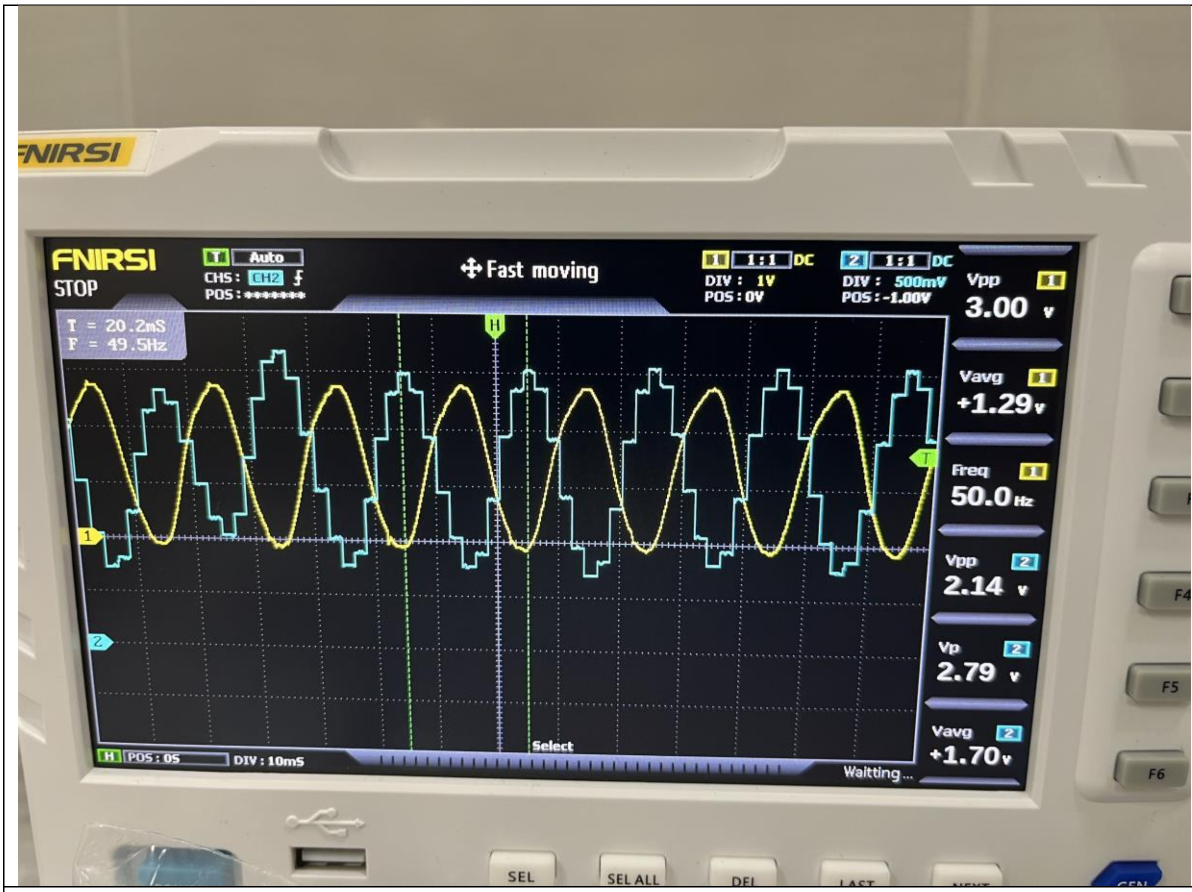


Figure 29. Output only showing 50Hz correlated signal showing SNR is not high enough.

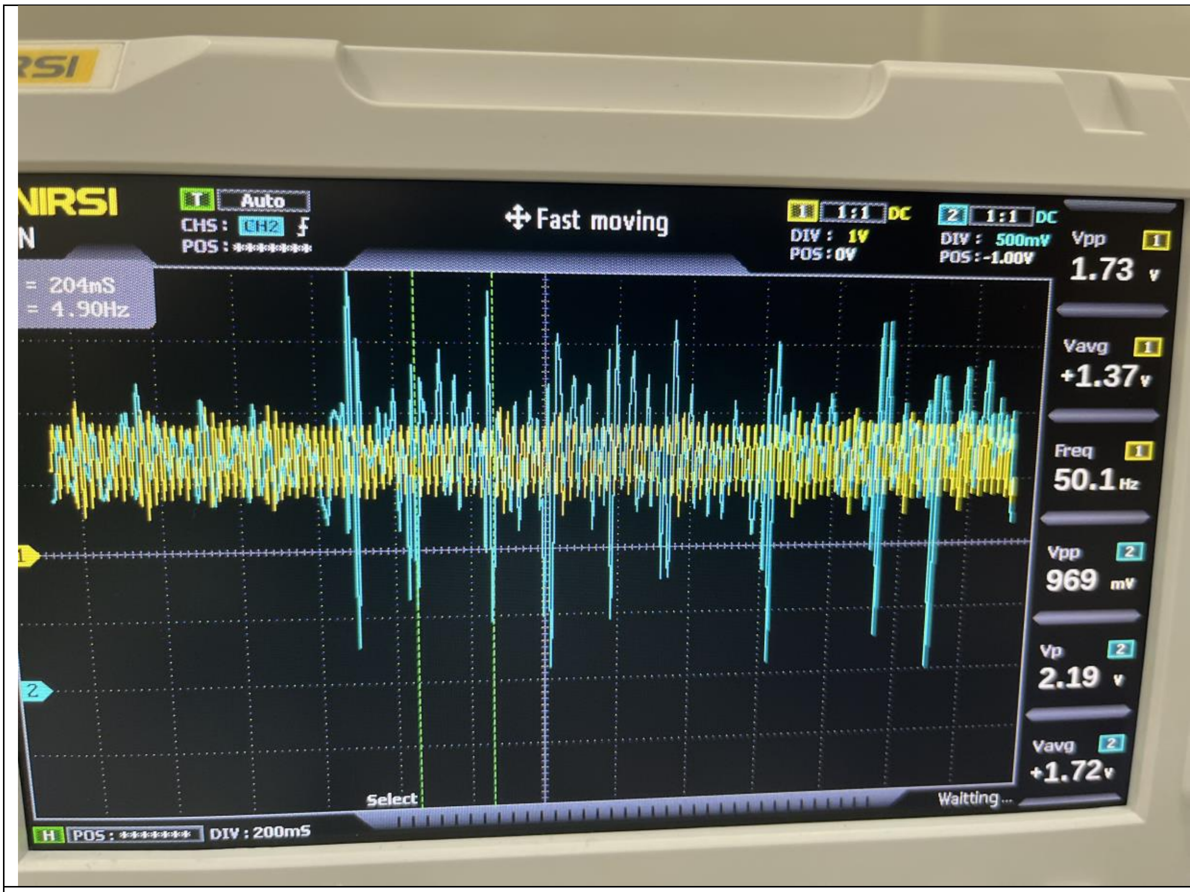


Figure 30. Output Swamped with artefacts and white noise.

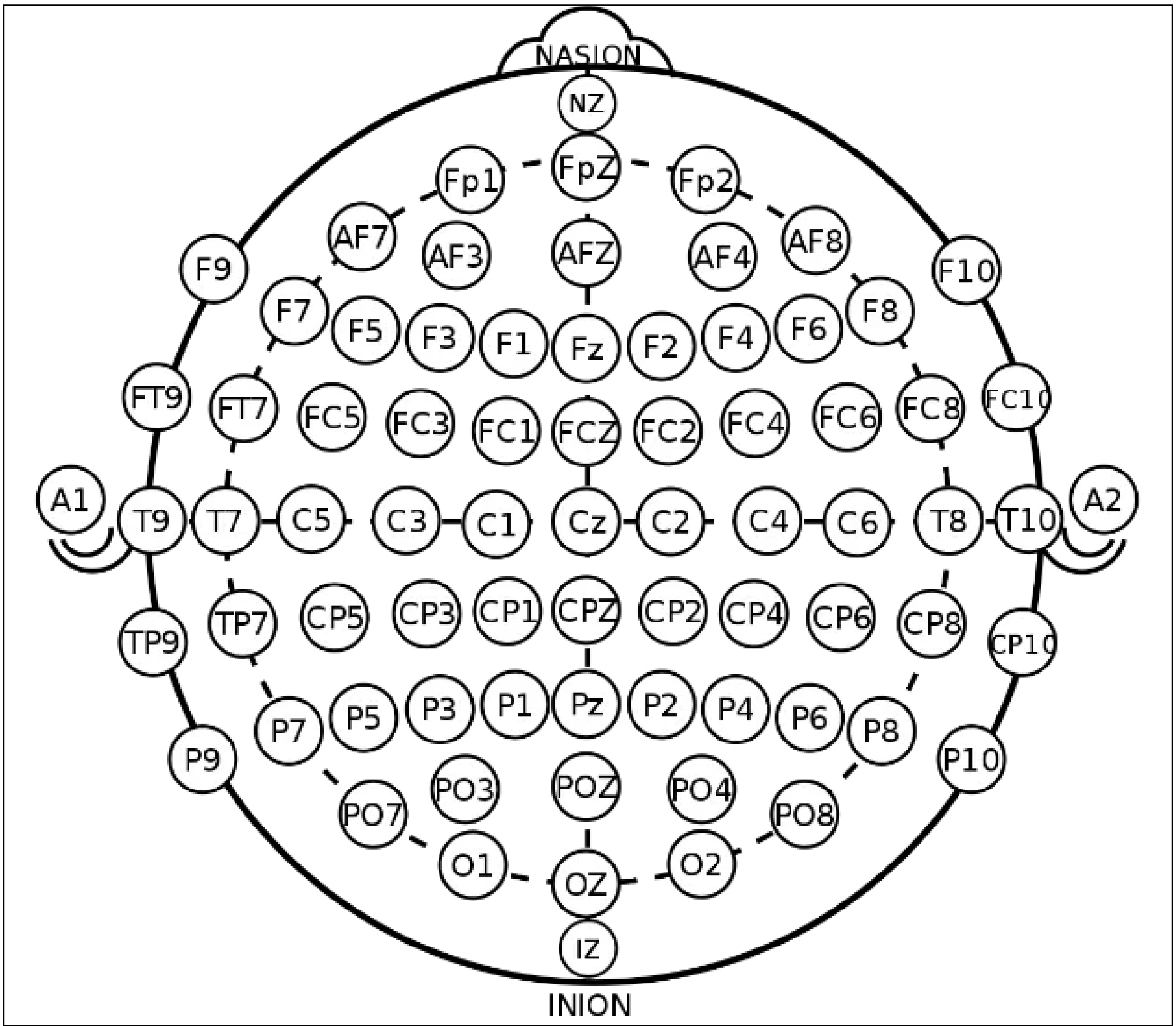


Figure 31. The 10-20 electrode placement map

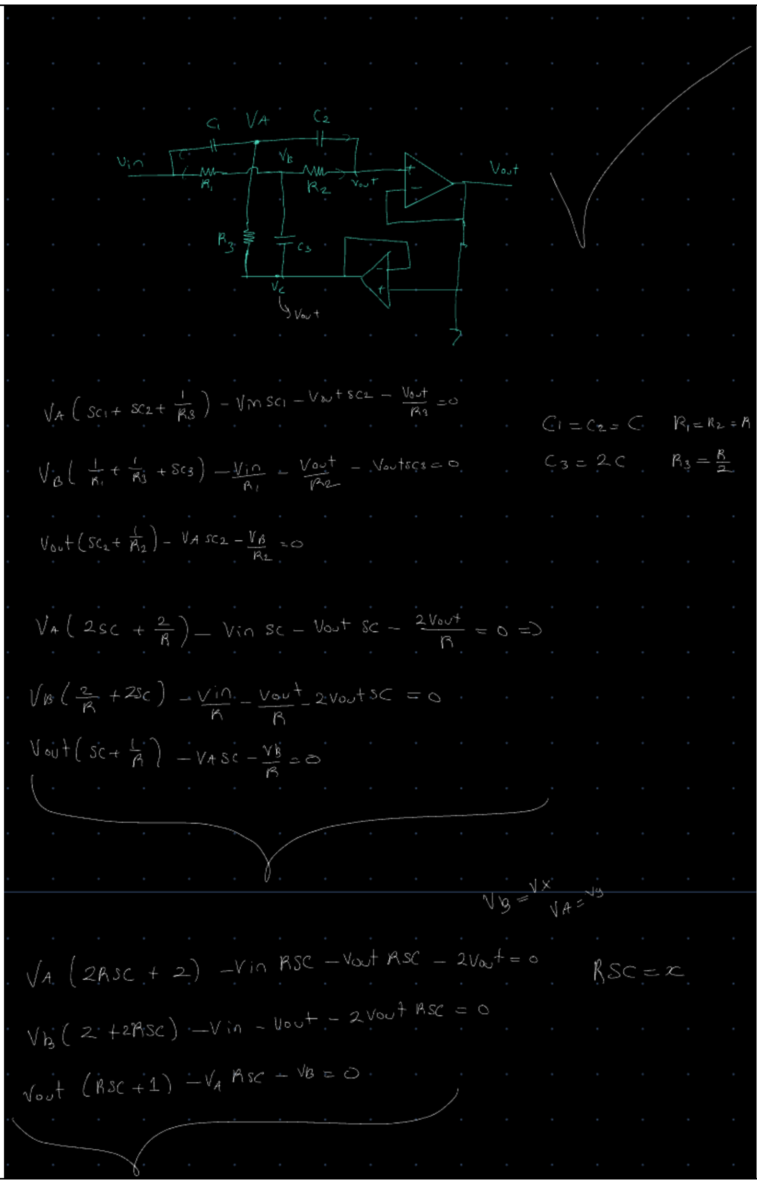
After the changes made the signal quality improved and started look more like EEG which you can see the outputs shown in the Results section.

Conclusion

This project successfully demonstrated the design and implementation of a custom single-channel EEG device, developed from PCB fabrication to real-time brainwave visualization. By combining analog front-end circuitry for amplification and filtering with digital signal processing on an STM32 microcontroller, the system was able to reliably capture and display EEG activity. The device achieved a signal-to-noise ratio in the 6 to 6.8 dB range and achieved a 5.2 Alpha Ratio when the subjects eyes were closed validating its ability to detect genuine neural activity. Key challenges such as 50 Hz interference, electrode impedance, and motion artefacts were addressed through shielding, driven-right-leg circuitry, digital notch filtering, and the application of Welch’s method. While limited by its single-channel architecture and sensitivity to electrode placement, the project establishes a solid foundation for future improvements, including multi-channel expansion, wireless data transfer, and advanced artefact rejection techniques such as ICA/PCA using multiple channels to capture EEG data. Overall this project demonstrates that a low-cost DIY EEG can reach a resonable

SNR levels showing that further EEGs can be made escpecially using DSP methods for research purposes more cheaply and reliably.

Appendix



Appendix A1. Notch filter derivation part 1

$$V_A(2x+2) - V_{in}x - V_{out}(x+2) = 0 \Rightarrow V_A = \frac{V_{in}x + V_{out}(x+2)}{2x+2}$$

$$V_B(2x+2) - V_{in} - V_{out}(x+1) = 0 \qquad = \frac{V_{in} + V_{out}(x+1)}{2x+2}$$

$$V_{out}(x+1) - V_A x - V_B = 0$$

$$V_{out}(x+1) = \frac{V_{in}x^2 + V_{out}(x^2+2x) + V_{in} + V_{out}(x+1)}{2x+2}$$

$$V_{out}\,2(x+1)^2 = V_{in}(x^2+1) + V_{out}(x^2+3x+1)$$

$$V_{out}\left(2(x+1)^2 - (x^2+3x+1)\right) = V_{in}(x^2+1)$$

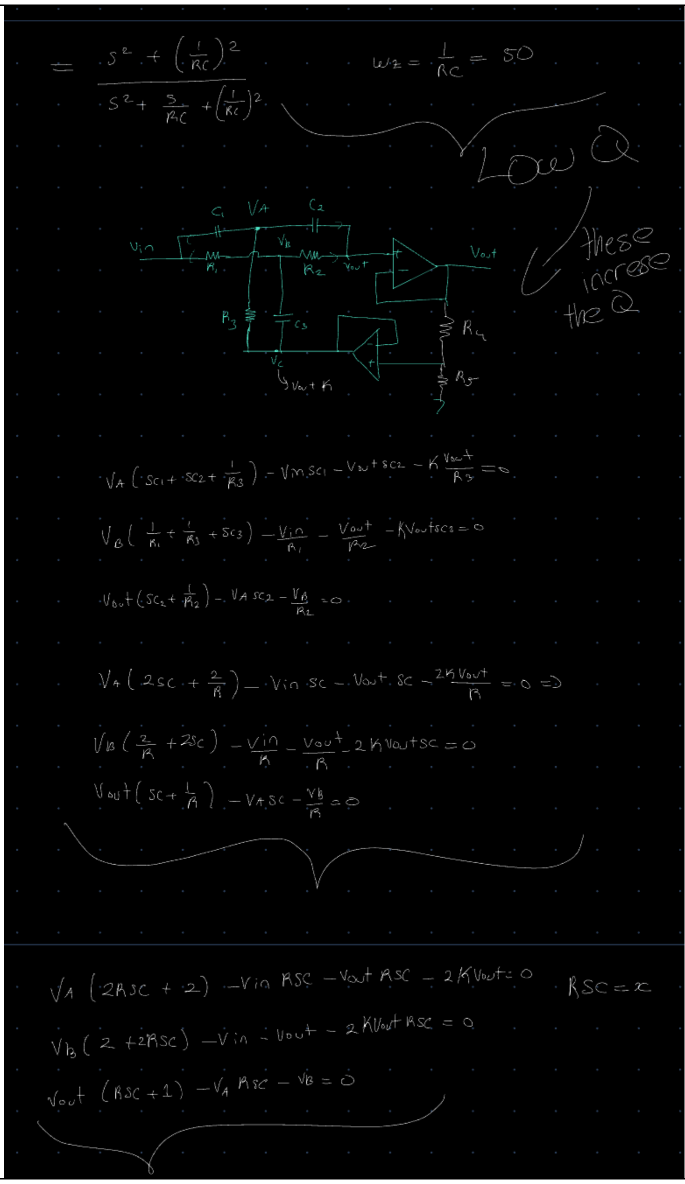
$$V_{out}(2x^2+4x+2-x^2-3x-1) = V_{in}(x^2+1)$$

$$V_{out}(x^2+x+1) = V_{in}(x^2+1)$$

$$H(x) = \frac{x^2+1}{x^2+x+1} \xrightarrow[x \to RSC]{\text{c.o.v.}} \frac{(RSC)^2+1}{(RSC)^2+(RSC)+1}$$

$$= \frac{s^2+\left(\frac{1}{RC}\right)^2}{s^2+\frac{s}{RC}+\left(\frac{1}{RC}\right)^2} \qquad \omega_z = \frac{1}{RC} = 50$$

Appendix A2. Notch filter derivation part 2



Appendix A3. Notch filter derivation part 3

$$V_A(2x+2) - V_{in}x - V_{out}(x+2K) = 0 \Rightarrow V_A = \frac{V_{in}x + V_{out}(x+2K)}{2x+2}$$

$$V_B(2x+2) - V_{in} - V_{out}(2Kx+1) = 0 \qquad = \frac{V_{in} + V_{out}(2Kx+1)}{2x+2}$$

$$V_{out}(x+1) - V_A x - V_B = 0$$

$$V_{out}\,2(x+1)^2 = V_{in}x^2 + V_{out}(x+2K)x + V_{in} + V_{out}(2Kx+1)$$

$$V_{out}\,2(x+1)^2 = V_{in}(x^2+1) + V_{out}(x^2+4Kx+1)$$

$$V_{out}(2x^2+4x+2-x^2-4Kx-1) = V_{in}(x^2+1)$$

$$V_{out}(x^2+4x(1-K)+1) = V_{in}(x^2+1)$$

$$\frac{V_{out}}{V_{in}} = \frac{x^2+1}{x^2+4x(1-k)+1} \underset{\substack{\text{cov.}\\ x \to S}}{=} H(s) = \frac{s^2(RC)^2+1}{s^2(RC)^2+4sRC(1-k)+1}$$

$$= \frac{S^2+\left(\frac{1}{RC}\right)^2}{s^2+s\frac{4(1-k)}{RC}+\left(\frac{1}{RC}\right)^2} \qquad w_0 = \frac{1}{RC}$$

$$Q$$

standard form

$$\frac{S^2+w_0^2}{S^2+S\frac{w_0}{Q}+w_0^2} \qquad \frac{1}{Q} = 4(1-k)$$

$$= \quad \uparrow Q = \frac{1}{\downarrow 4(1-k)} = \frac{1}{\sqrt{2}}$$

$$w_0 = \frac{1}{RC} \qquad \frac{1}{1-k} = \frac{4}{\sqrt{2}}$$

$$F_0 = \frac{w}{2\pi} \qquad 1-k = \frac{\sqrt{2}}{4}$$

$$k = 1-\frac{\sqrt{2}}{4} = 0.6464$$

$$w = 2\pi F_0$$

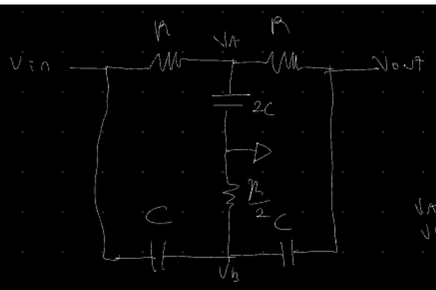
$$\frac{1}{1-k} = 4Q$$

$$f = \frac{1}{2\pi RC} \qquad 1-k = \frac{1}{4Q}$$

$$1-\frac{1}{4Q} = k$$

$$\frac{R_5}{R_4+R_5} = k$$

Appendix A4. Notch filter derivation part 4



$V_A = V_1$
$V_B = V_2$

$V_A\left(\frac{2}{R} + 2SC\right) - \frac{V_{in}}{R} - \frac{V_{out}}{R} = 0 \qquad V_{out}\left(\frac{1}{R} + SC\right) - \frac{V_A}{R} - V_B\,SC -$

$V_B\left(2SC + \frac{2}{R}\right) - V_{in}SC - V_{out}SC = 0$

$V_A(2 + 2RSC) - V_{in} - V_{out} = 0 \qquad V_A = \frac{V_{in} + V_{out}}{2 + 2RSC}$

$V_B(2RSC + 2) - V_{in}RSC - V_{out}RSC = 0 \qquad V_B = \frac{(V_{in} + V_{out})RSC}{2(1 + RSC)}$

$V_{out}(1 + RSC) - V_A - V_B\,RSC = 0$

$$V_{out}(1 + RSC) = \frac{V_{in} + V_{out}}{2(1 + RSC)} + \frac{(V_{in} + V_{out})(RSC)^2}{2(1 + RSC)}$$

$$\underbrace{V_{out}\,2(1 + RSC)^2} = V_{in}(1 + (RSC)^2) + V_{out}(1 + (RSC)^2)$$

$$V_{out}\left(2(1 + RSC)^2 - 1 - (RSC)^2\right) = V_{out}\left(2(RSC)^2 + 4RSC + 2 - 1 - (RSC)^2\right)$$

$$= V_{out}\left((RSC)^2 + 4RSC + 1\right) = V_{in}\left(1 + (RSC)^2\right)$$

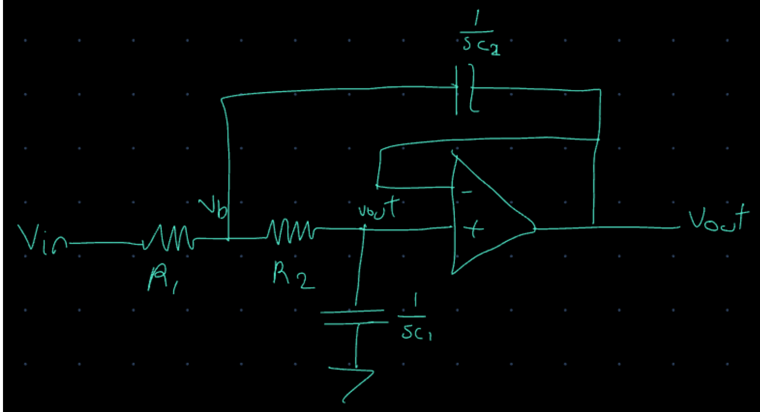
$$= H(s) = \frac{1 + (RSC)^2}{(RSC)^2 + 4RSC + 1} = \frac{\left(\frac{1}{RC}\right)^2 + s^2}{s^2 + s\frac{4}{RC} + \left(\frac{1}{RC}\right)^2} =$$

$$\omega_0 = \frac{1}{RC} \qquad f_0 = \frac{1}{2\pi RC} = \frac{1}{2\pi\, 3{,}10^3 \times 10^{-6}} = \frac{1}{6\pi \times 10^{-3}}$$

$$\frac{10^3}{6\pi} = 53\,\text{Hz} \qquad Q = \frac{1}{4} = 0.25$$

Appendix A5. Notch filter derivation alternative path

$$H(s) = \frac{K \cdot \omega_c^2}{s^2 + s\left(\frac{\omega_c}{Q}\right) + \omega_c^2}$$



$$\frac{V_b - V_{in}}{R_1} + (V_b - V_{out})sC_2 + \frac{V_b - V_{out}}{R_2} = 0$$

$$V_{out} = \frac{\frac{1}{sC_1}}{R_2 + \frac{1}{sC_1}} V_b \qquad V_b = V_{out}\left(\frac{R_2 + \frac{1}{sC_1}}{\frac{1}{sC_1}}\right)$$

$$V_b = V_{out}\left(\frac{R_2}{sC_1} + 1\right)$$

Appendix B1. LPF derivation

$$\frac{V_{out}\left(\frac{R_2}{sC_1}+1\right)-V_{in}}{R_1} + V_{out}sC_1\left(\frac{R_2}{sC_1}+1\right) - V_{out}sC_1$$

$$+ V_{out}\left(\frac{R_2}{sC_1}+1\right)\frac{1}{R_2}$$ i won't bother :)

$$\omega_c^2 = \frac{1}{R_1 R_2 C_1 C_2}$$

$$(xy)^2 = x^2 y^2$$

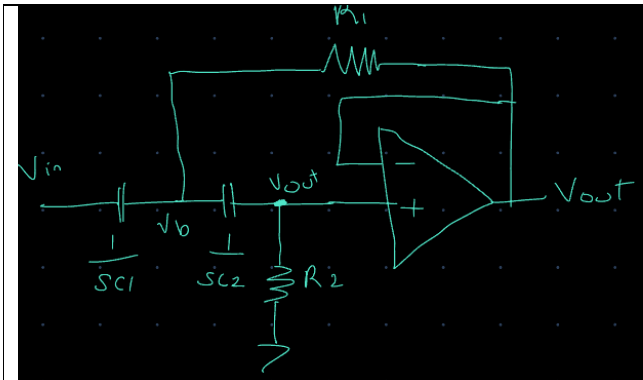
$$Q^2 = \frac{R_1 R_2 C_1 C_2}{C_2^2 (R_1+R_2)^2}$$

$$\frac{Q^2 C_2}{C_1} = \frac{R_1 R_2}{(R_1+R_2)^2} = \frac{R_1 R_2}{R_1^2 + 2R_1R_2 + R_2^2}$$

$$\boxed{\frac{Q^2 C_2}{C_1 R_2} = \frac{R_1}{R_1^2 + 2R_1R_2 + R_2^2}}$$

$$\boxed{R_2 = \frac{1}{R_1 \omega_c^2 C_1 C_2}}$$

Appendix B2. LPF derivation



$$(V_b - V_{in})sC_1 + (V_b - V_{out})sC_2 + \frac{V_b - V_{out}}{R_1} = 0$$

$$V_{out} = V_b \frac{R_2}{\frac{1}{sC_2} + R_2} \Rightarrow V_b = \frac{\frac{1}{sC_2} + R_2}{R_2} V_{out}$$

$$V_b\left(sC_1 + sC_2 + \frac{1}{R_1}\right) - V_{in}sC_1 - V_{out}\left(sC_2 + \frac{1}{R_1}\right) = 0$$

$$V_{out}\underbrace{\frac{\frac{1}{sC_2} + R_2}{R_2}}_{\alpha}\left(sC_1 + \underbrace{sC_2 + \frac{1}{R_1}}_{\beta}\right) - V_{in}sC_1 - V_{out}\underbrace{\left(sC_2 + \frac{1}{R_1}\right)}_{\beta} = 0$$

$$V_{out}\ \alpha(sC_1 + \beta) - V_{in}sC_1 - V_{out}\beta = 0$$

Appendix C1. HPF Derivation

$$V_{out}\left(\alpha(sC_1+\beta)-\beta\right)-V_{in}sC_1=0$$

$$V_{out}\left(\alpha sC_1+\alpha\beta-\beta\right)-V_{in}sC_1=0$$

$$\frac{V_{out}}{V_{in}}=\frac{sC_1}{\alpha(sC_1+\beta)-\beta}$$

$$=\frac{sC_1}{\dfrac{\frac{1}{sC_2}+R_2}{R_2}\left(sC_1+sC_2+\frac{1}{R_1}\right)-sC_2-\frac{1}{R_1}}$$

$$=\frac{sC_1}{\left(\frac{1}{sC_2R_2}+1\right)\left(sC_1+sC_2+\frac{1}{R_1}\right)-sC_2-\frac{1}{R_1}}$$

$$=\frac{sC_1}{\dfrac{sC_1+sC_2+\frac{1}{R_1}}{sC_2R_2}+sC_1+\cancel{sC_2}+\cancel{\frac{1}{R_1}}-\cancel{sC_2}-\cancel{\frac{1}{R_1}}}$$

Appendix C2. HPF Derivation

$$= \frac{sC_1}{\dfrac{sC_1 + sC_2 + \frac{1}{R_1}}{sC_2 R_2} + sC_1}$$

$$= \frac{sC_1}{\dfrac{C_1}{C_2 R_2} + \dfrac{C_2}{C_2 R_2} + \dfrac{1}{sC_2 R_2 R_1} + sC_1}$$

$$= \frac{sC_1}{sC_1 + \dfrac{C_1 + C_2}{C_2 R_2} + \dfrac{1}{sC_2 R_2 R_1}}$$

$$= \frac{s^2 C_1}{s^2 C_1 + \dfrac{s(C_1 + C_2)}{C_2 R_2} + \dfrac{1}{C_2 R_1 R_2}}$$

$$= \frac{s^2}{s^2 + \dfrac{s(C_1 + C_2)}{C_1 C_2 R_2} + \dfrac{1}{C_1 C_2 R_1 R_2}}$$

$$H(s) = \frac{K \cdot s^2}{s^2 + s\left(\frac{\omega_s}{a}\right) + \omega_0^2}$$

Appendix C3. HPF Derivation

$$\frac{w_s}{Q} = \frac{C_1 + C_2}{C_1 C_2 R_2}$$

$$w_s^2 = \frac{1}{C_1 C_2 R_1 R_2}$$

$$\frac{2 \times \cancel{10^{-5}}}{10^{\cancel{-10}\,-5} R_2} = \frac{1}{0.707} \qquad R_2 = \frac{1}{0.707} \times \frac{10^{-5}}{2}$$

$$1 \times 10^{-10} = \frac{1}{R_1 R_2} = R_1 = \frac{1}{R_2 \times 10^{-10}}$$

$$R_1 = \frac{10^{10}}{R_2}$$

$$\boxed{\begin{aligned} \frac{C_1 + C_2}{C_1 C_2} \cdot \frac{w_s}{Q} &= R_2 \\ R_1 &= \frac{1}{C_1 C_2 w_s^2 R_2} \end{aligned}}$$

Appendix C4. HPF derivation