Part 1

To improve SDR1.0, we had to modify and add an additional component on it. LTspice software was used to test the results. There are 5 main parts of our SDR circuit, a bandpass filter (the additional component of the circuit), a RF preamplifier, a mixer, an audio amplifier and an Analog-to-Digital Converter (ADC). Each part was simulated individually first, the final complete SDR2.0 circuit was simi at the end. (Results in Appendix[6])

Bandpass filter

An additional bandpass filter was built to improve the performance after doing the search. The design of the Bandpass filter had to meet a number of specifications including a maximum ripple of less than 0.1dB within the WSPR region (7.0400-7.0402 MHz), a centre frequency at 7.0401 MHz, a minimum of -3dB attenuation 0.25 MHz from the centre frequency (6.7901MHz – 7.2901 MHz), and a minimum -50dB attenuation at 10 MHz. The input and output impedance were required to be 50 ohms with 1% tolerance capacitor and inductor values. With a filter designing tool (https://rf-tools.com/lc-filter/), we could choose the design and run the simulation on LTspice. (Testing results in Appendix [1])

Preamplifier

The preamplifier in SDR 1.0 had design weakness in power supply rejection as “any change in supply voltage will directly change the biassing voltage set by the resistor divider” Moreover, due to its non-inverting characteristic, fluctuation in signal voltage on the supply line could be fed back to the operational amplifier resulting oscillations such as “motor boating”. Therefore, to reduce the potential noise levels from power supply and increase the stability of op-amp and signal output, the approach of decoupling the biassing network from supply was taken. (Testing results in Appendix [2])

Mixer

By feeding the right carrier frequency which was 7.0401 Mhz to the original mixer in SDR1.0, it had successfully downconverted the frequency as expected. (Testing results in Appendix [3]) Therefore, no modification is required.

Audio amplifier

The original audio amplifier only produced 0.7 V output voltage, and the 3dB cut-off frequency was 4.58kHz. The audio amplifier for the SDR1.0 circuit won’t provide the matched voltage to our ADC. According to the active low pass filter equation, the output voltage is related to the ratio between resistors and the 3dB cut-off frequency is also related to multiple of resistors. Hence, modifying the resistor values could result in better output voltage and 3dB cut-off frequency. (Testing results in Appendix [4])

ADC

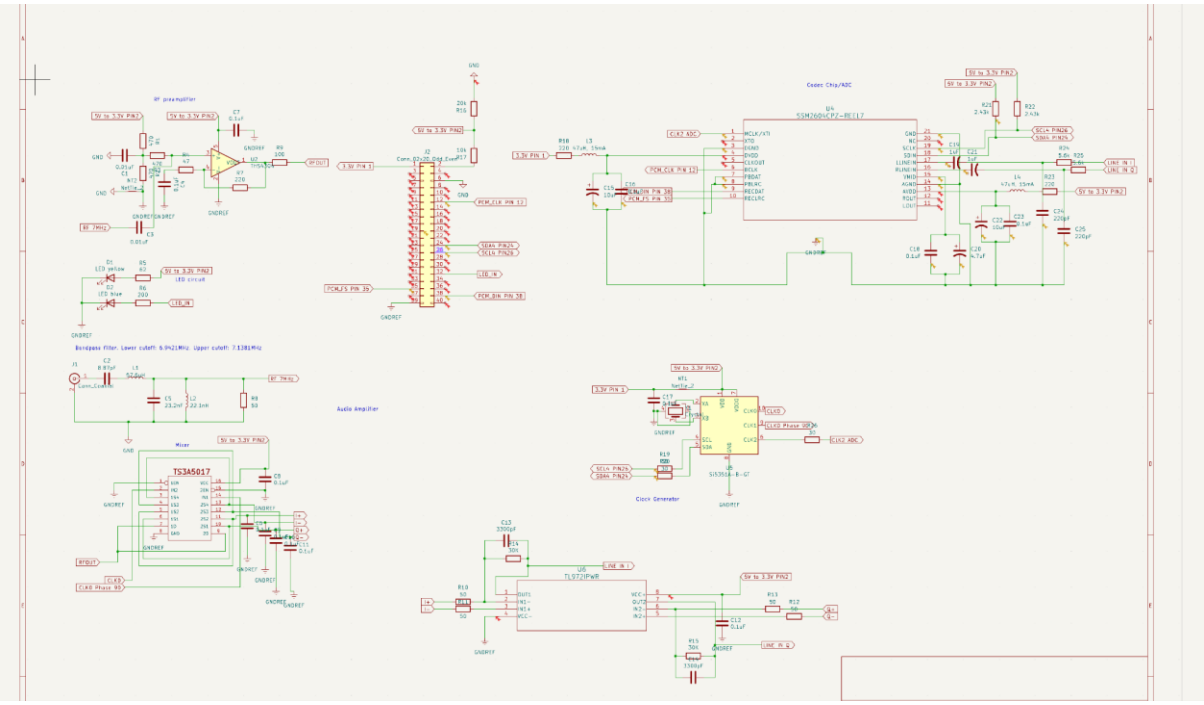
The original SDR 1.0 design uses a usb soundcard to convert analog data to digital. This implementation is sufficient, but the ADC can be much smaller and a separate sound card would need interfacing by a 3.5mm connector and USB cable which will be unnecessarily cluttered and clunky. A codec chip would solve all of this problem, so we implemented a SSM2604 codec chip from Analog instead.(Results in Appendix[5])

Part 2

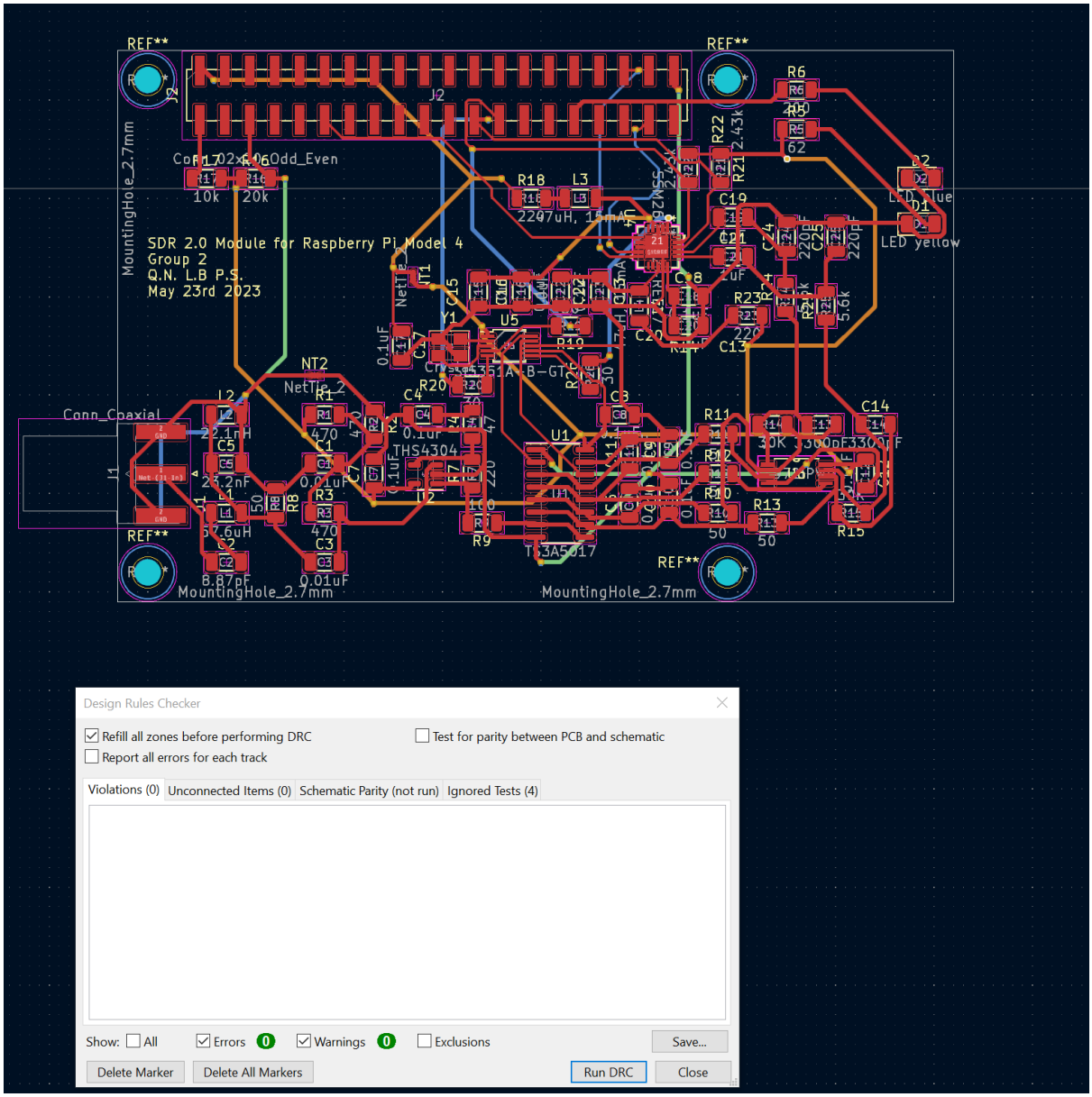
Summary of changes

- Additional bandpass filter implementation (LTspice and PCB)
- Improving RF preamplifier to be more resilient against power supply noise (LTspice and PCB)
- Adjusting the carrier frequency to 7.0401 Mhz.(LTspice and PCB)
- Adjusting resistor values on SDR 1.0 audio amplifier to 50 and 30k Ohms respectively.(LTspice and PCB)
- Add an ADC component (PCB)

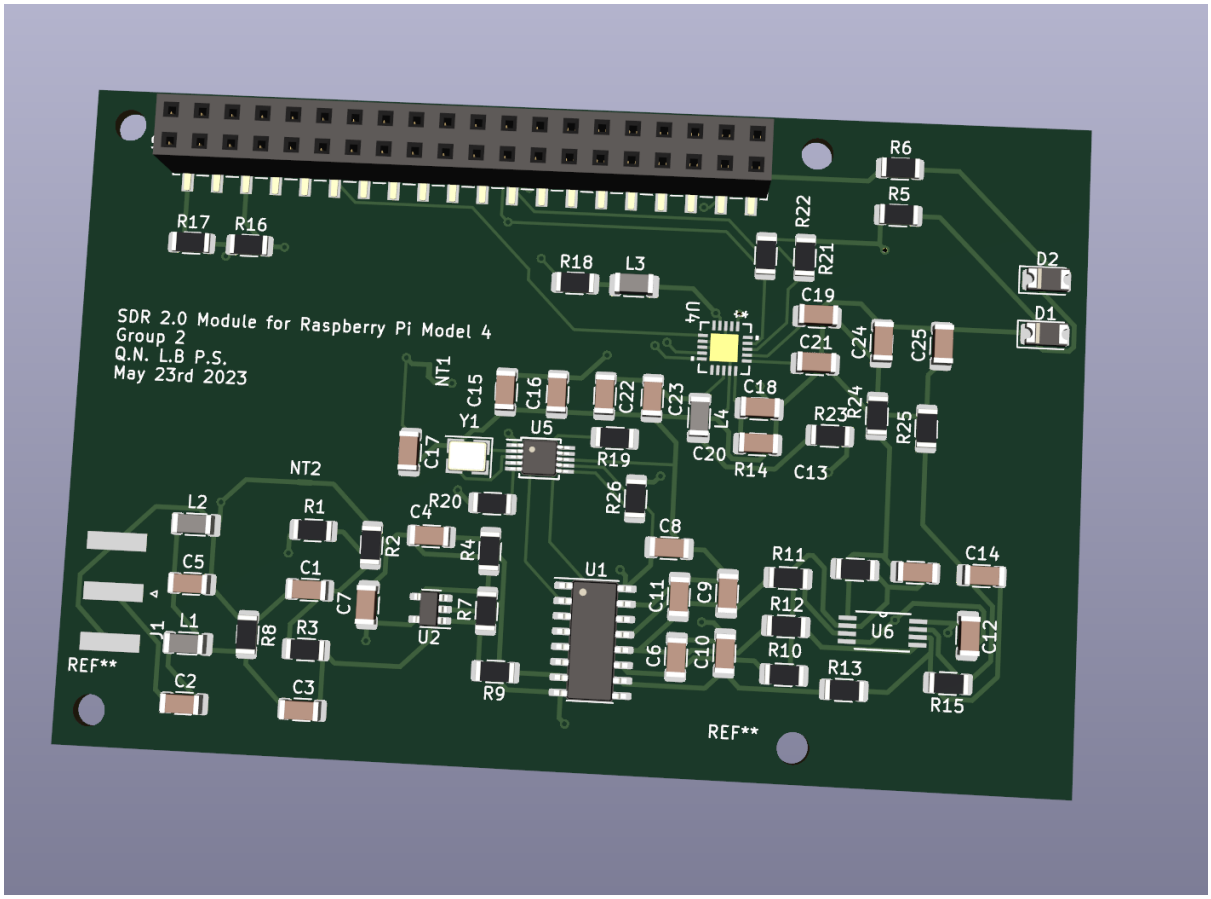
Schematic of PCB



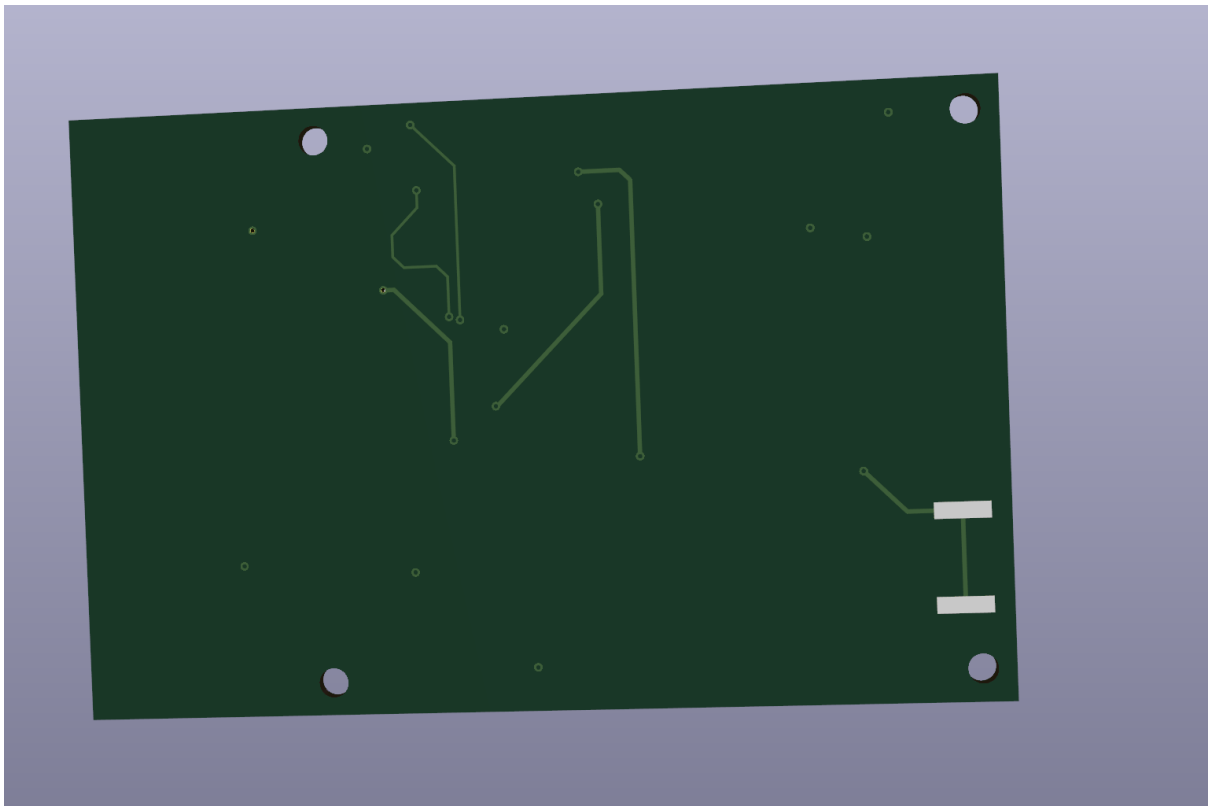
PCB (Appendix [7])



PCB with DRC check



Front 3D view of PCB



Back 3D view of PCB

Part 3

- Linglin Bai, 450537507, lbai7773@uni.sydney.edu.au, Bandpass filter, RF amplifier, PCB schematic design
- Prosper Su, 490447846, kusu6330@uni.sydney.edu.au Mixer, audio amplifier, PCB schematic design
- Quang Nguyen Nguyen, 510122717, vngu6249@uni.sydney.edu.au ADC, source code, PCB schematic design and tracing

Appendix

[1] Bandpass filter schematic and results

With a filter designing tool (https://rf-tools.com/lc-filter/), a 2nd order Butterworth Bandpass filter were generated with a lower cut-off frequency at 6.9421 MHz and a upper cut-off frequency at 7.1381 MHz which resulted in a centre frequency of 7.0401 MHz meeting the requirement of design. The input and output impedance were set to 50 Ohms, and a 1% tolerance was allowed for the capacitor and inductor values.

Monte Carlo simulation (n=100) was conducted on the Bandpass filter in LTSpice with 1% tolerance for capacitors and inductors. The simulation swept from 5.253MHz to 9.434MHz with 1001 steps per decade. The diagram below was the resulting Bode plot.

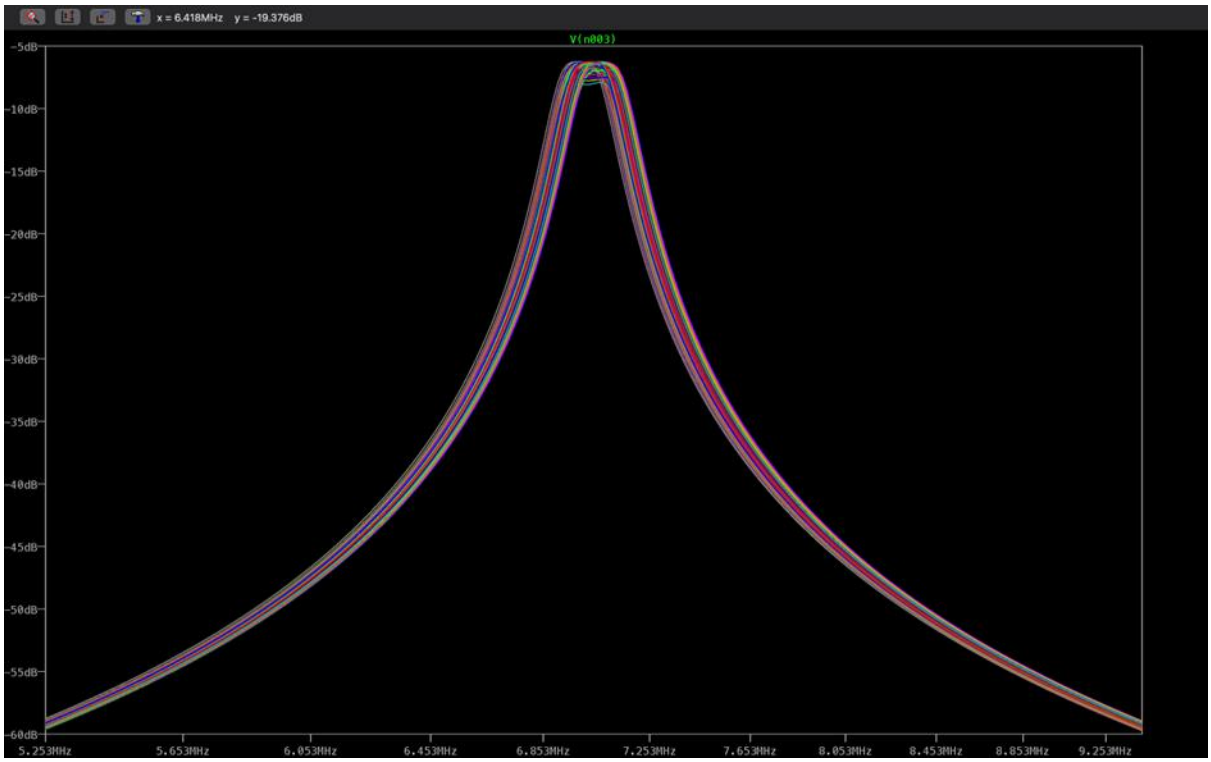


Figure 1.0 Bode plot

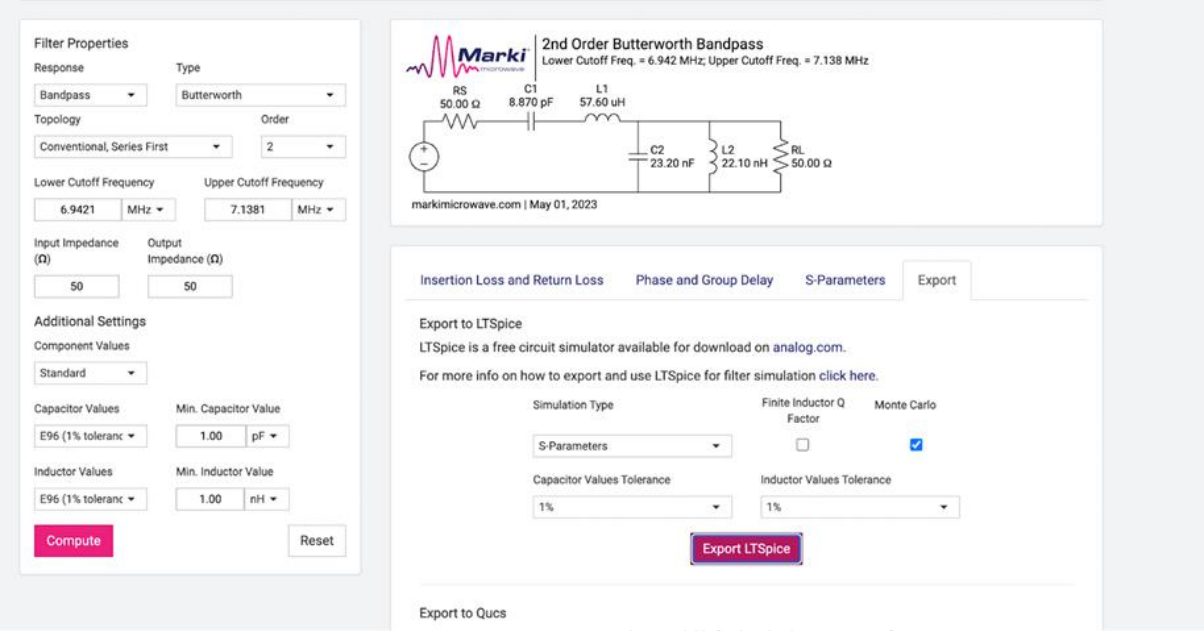


Figure 1.1 2nd order Butterworth Bandpass filter

The LTSpice circuit exported is shown below. The specification of input and output impedance of 50 ohm was met, evidenced by the circuit diagram with RL and resistance of V1 set to 50 ohm. A Monte Carlo simulation was conducted on the circuit.

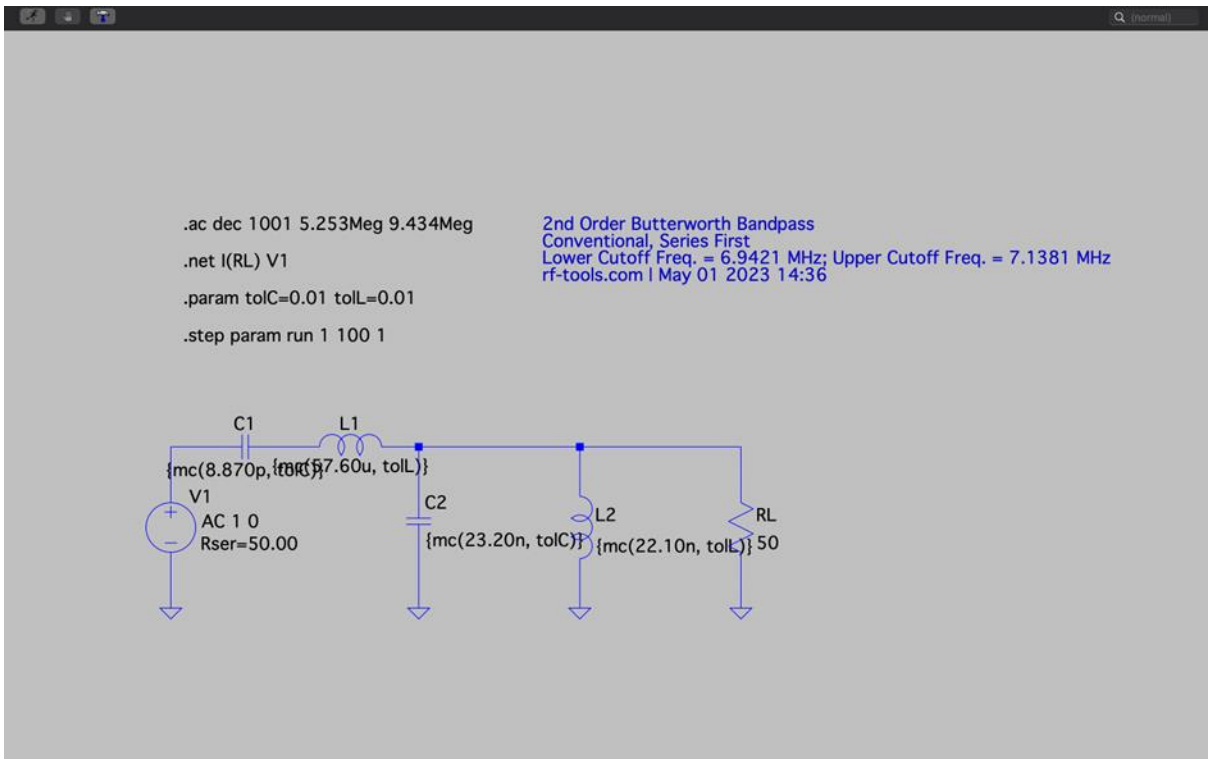


Figure 1.2 2nd order Butterworth Bandpass filter circuit simulation in LTspice

The first requirement for the bandpass filter was that the maximum ripple within the WSPR region (7.0400-7.0402 MHz) should be less than 0.1dB. The following screenshot was a zoomed in version of figure 1.0 on the region of interest, indicated in x-axis. It was clear from the plot that all instances

of the simulation were almost straight lines with almost no ripple which satisfied the requirement of maximum ripple.

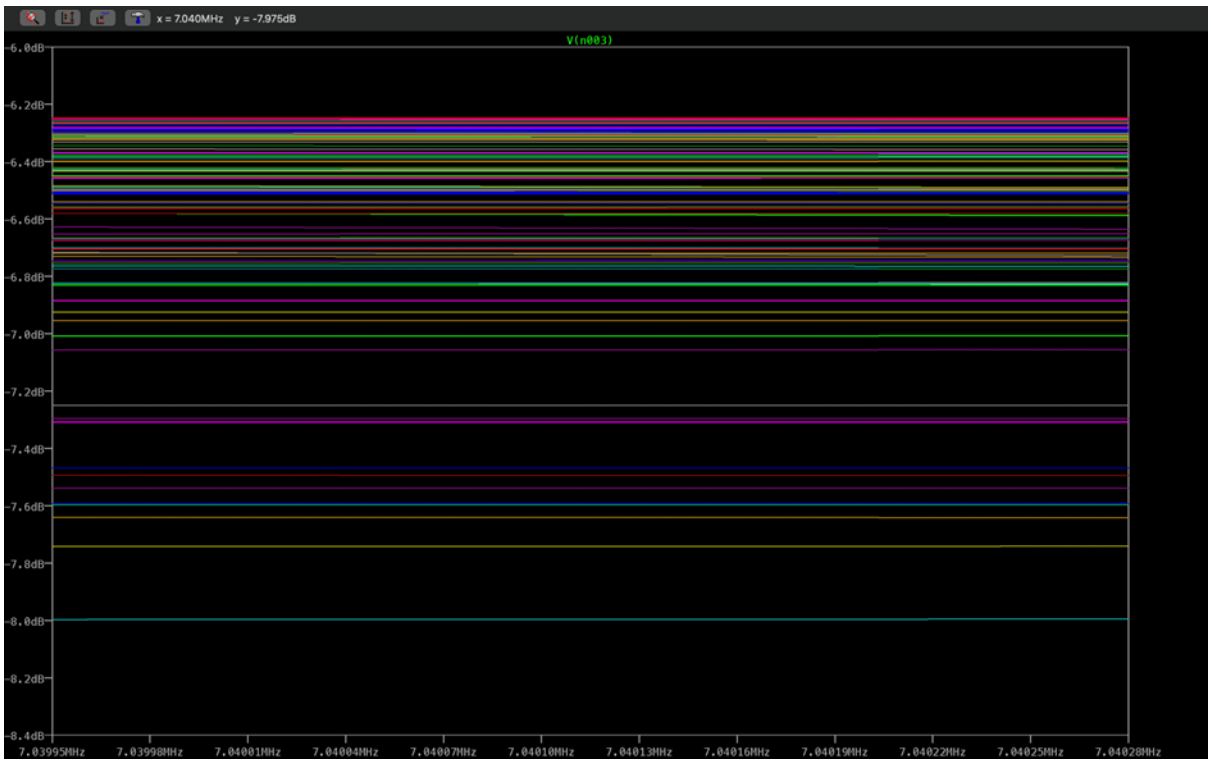


Figure 1.3 Zoomed in version on the region of interest

The bandpass filter is also required to have a minimum -3dB attenuation at 0.25 MHz (6.7901MHz – 7.2901 MHz) from centre frequency 7.0401 MHz. It can be seen that the lowest peak was roughly -8dB, and the attenuation observed on the frequencies 6.7901MHz and 7.2901 MHz was well below -11dB, 3dB below the lowest peak. Therefore, the requirement was met.

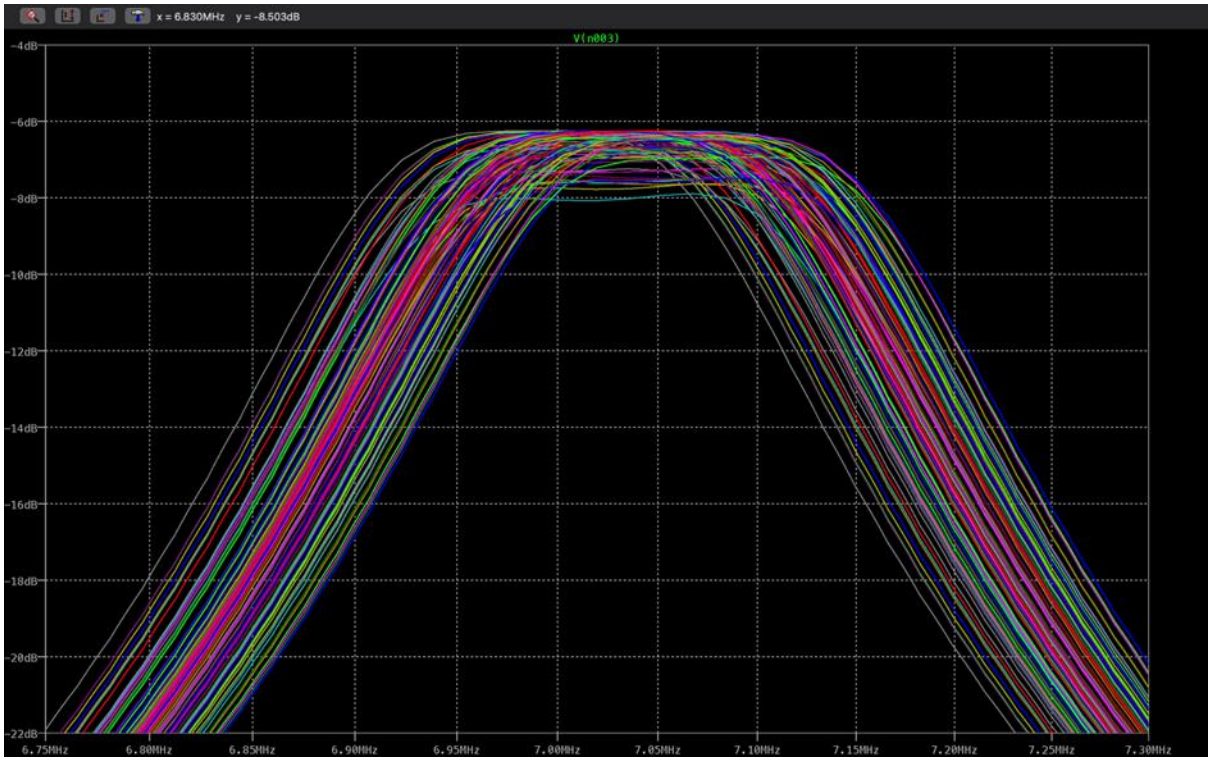


Figure 1.4 Zoomed in version

Moreover, a minimum of -50dB from centre frequency 7.0401MHz at 10 MHz was required. It was observed from figure 1.0 that at 9.4Mhz, the -60dB was almost achieved which was more than 50dB below the peak at centre frequency. With a decreasing trend, the requirement for a minimum of -50dB at 10 MHz was also met.

[2] RF preamplifier schematic and results

The schematic of circuits for preamplifier 1.0 and 2.0 were shown below (figure 2.0 and figure 2.1).

Power supply rejection ability of the preamplifiers was tested in LTSpice by replacing the voltage input of 3.3V with command “V=3.3+0.1*white(1e6*time)” which mimicked the noise presented in the power supply. Both input signals were set to a sine wave with 0.01 Vpp in amplitude and a 0 DC offset. As a result, a decrease trend in the peaks of the output wave produced by preamplifier 1.0 was observed (figure 2.2) while a stable sine wave with constant DC offset was produced by preamplifier 2.0 (figure 2.3). Therefore, the improved version of the preamplifier was proven to have an adequate power rejection from the supply as desired.

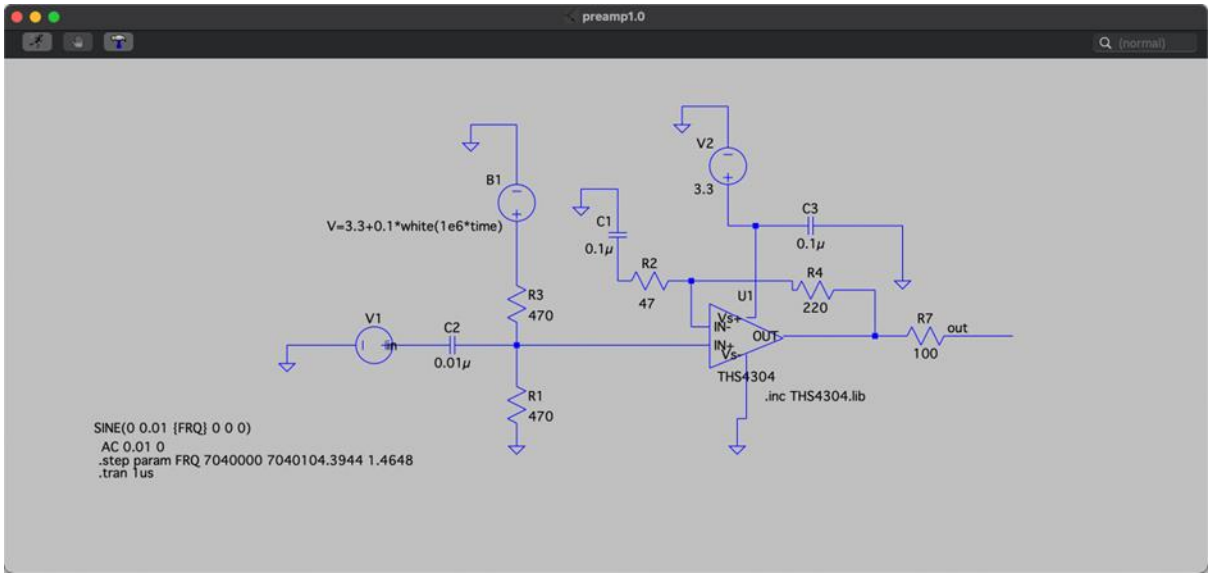


Figure 2.0 RF preamplifier 1.0 circuit simulation in LTspice

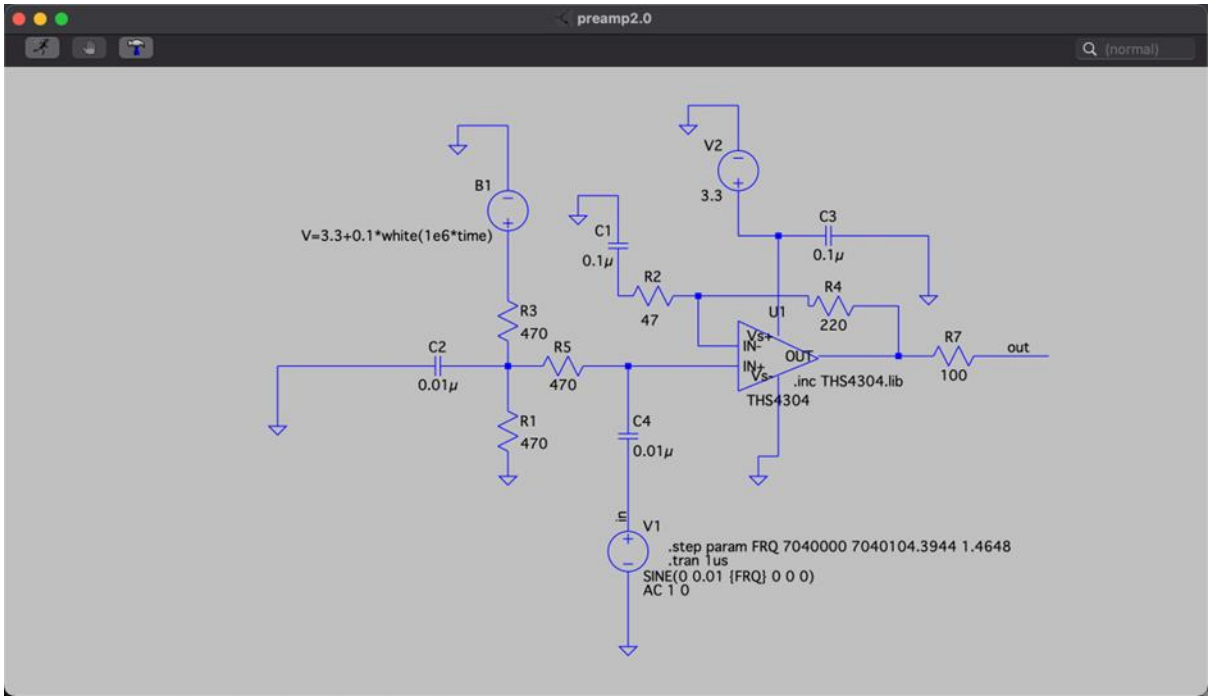


Figure 2.1 RF preamplifier 2.0 circuit simulation in LTspice

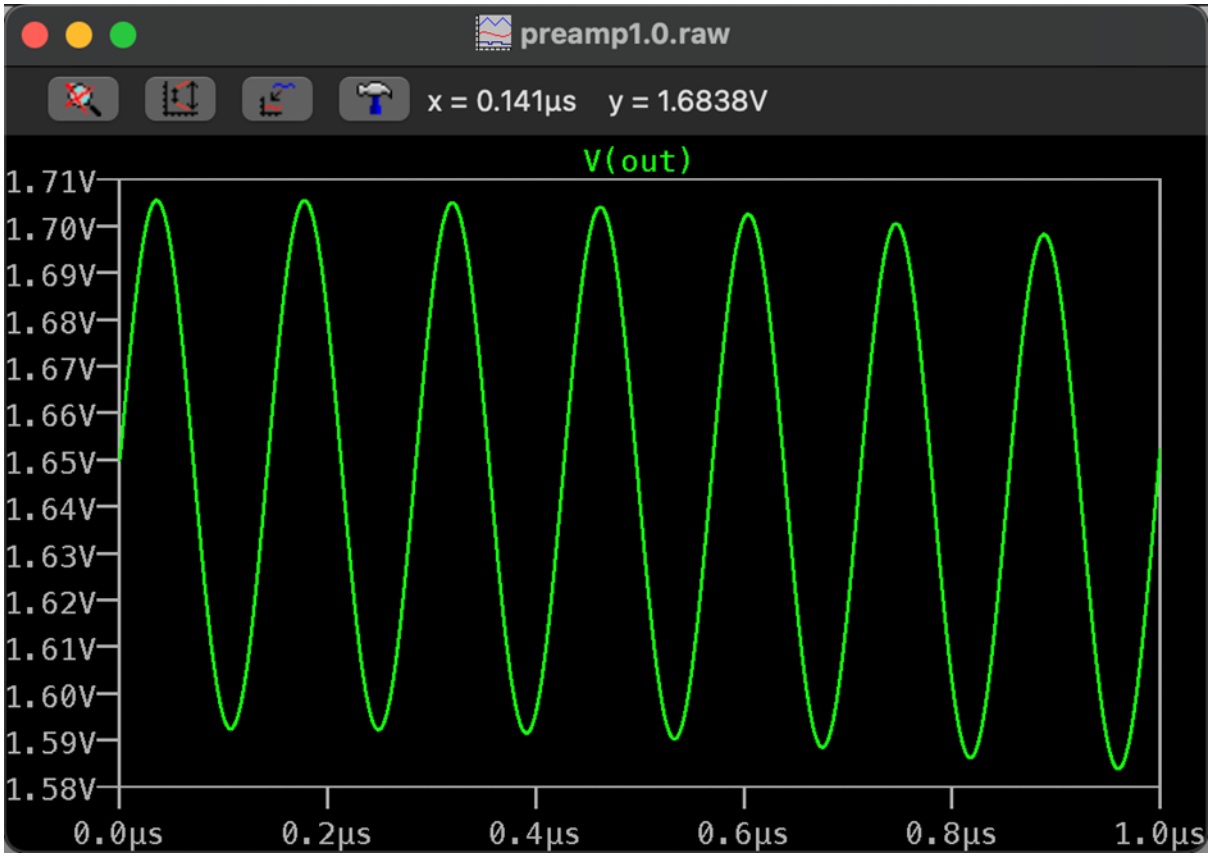


Figure 2.2 Magnitude of RF preamplifier 1.0

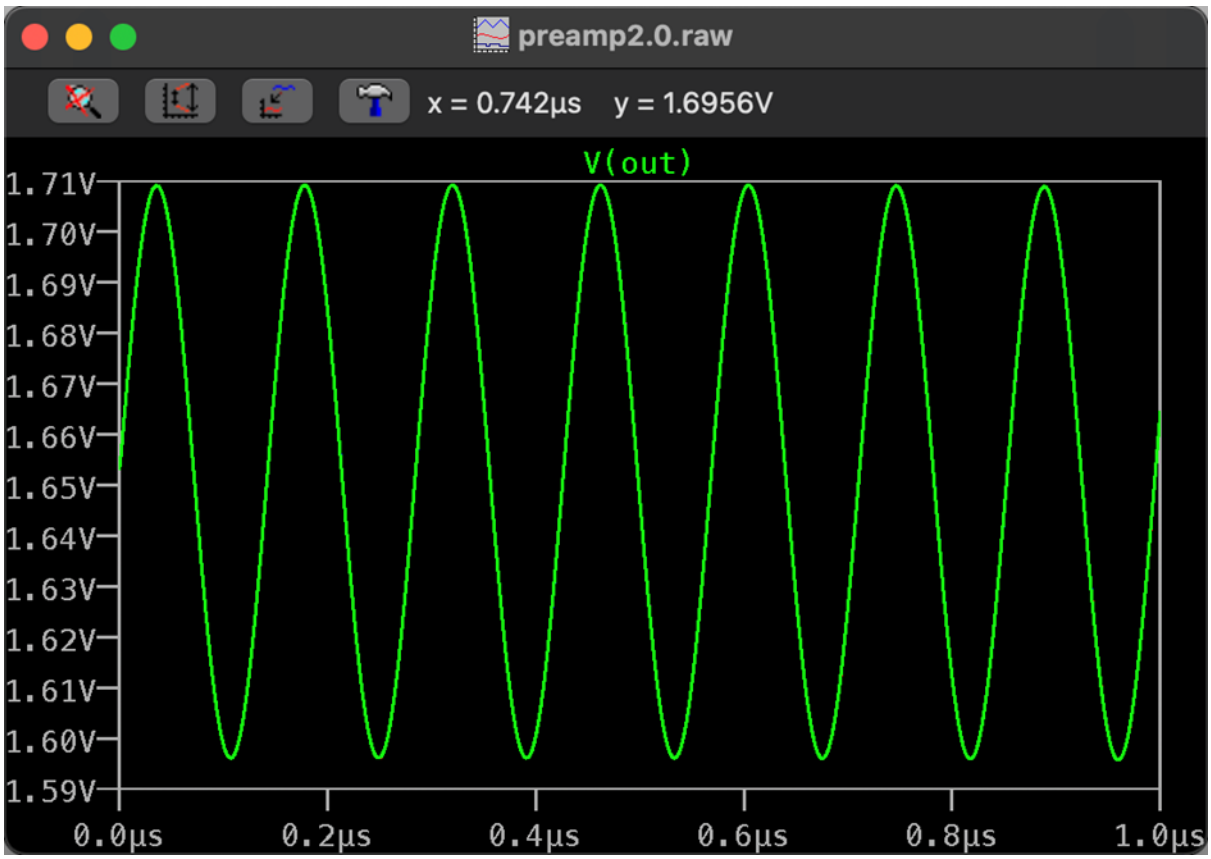


Figure 2.3 Magnitude of RF preamplifier 2.0

[3] Mixer schematic and results

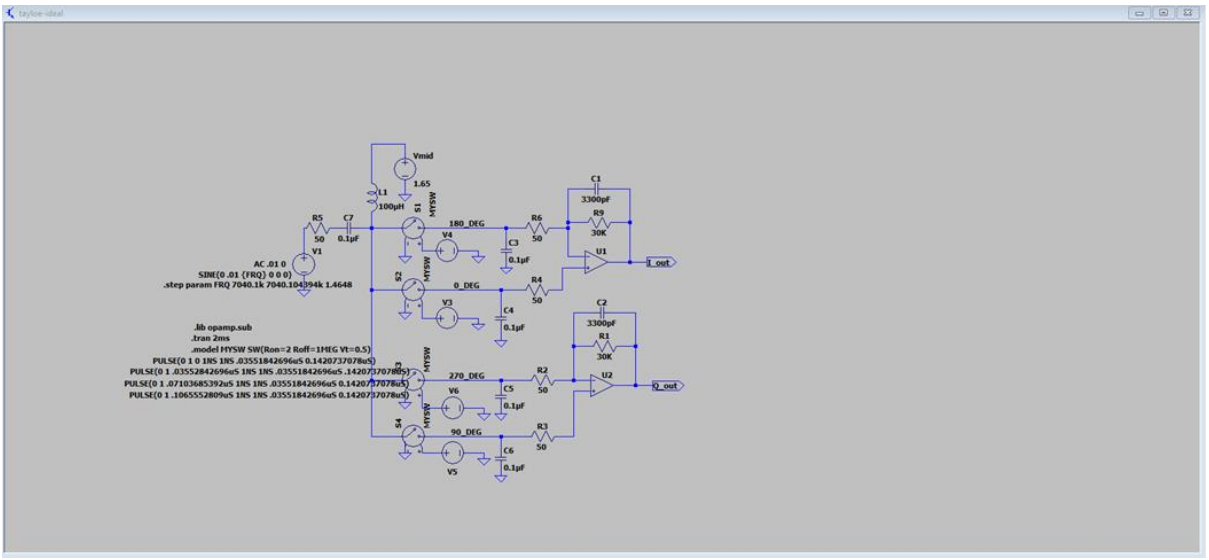


Figure 3.0 Mixer circuit simulation in LTspice

The overall circuit is shown as above (mixer + audio amplifier). To begin with, the pulse modifications required – pulse period is 1/fc (fc stands for carrier frequency). This is because we want to remove the carrier frequency modulation which will be 7040.1kHz to 7040.104394kHz to extract our 1.4kHz to 1.6kHz WISPR signal by multiplying by a sinusoidal wave with the same frequency as the carrier. Then, to match up with the input voltage of the analogue-to-digital converter. The resistor modifications are required – 50 ohms and 30k ohms respectively. (This will be further explained in the audio amplifier section) The results also indicate that it has good performance in either frequency domain or voltage domain. (figure 3.1)

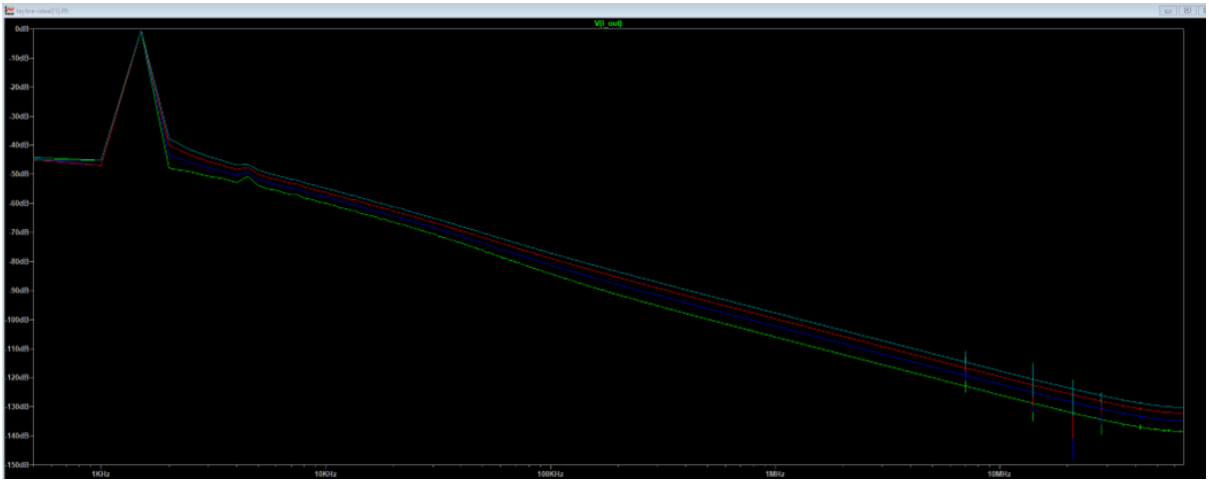


Figure 3.1 FFT graph (FFT stands for Fast Fourier Transform or frequency domain graph)

Figure 3.2 Zoomed in version on the region of interest

The apparent triangle shape values in FFT shows that only that range of frequency can pass through the entire circuit. If we further magnify this range, we can see that it's the range we want (1.4kHz to 1.6kHz) This proves that the circuit has good performance in the frequency domain. (figure 3.2)

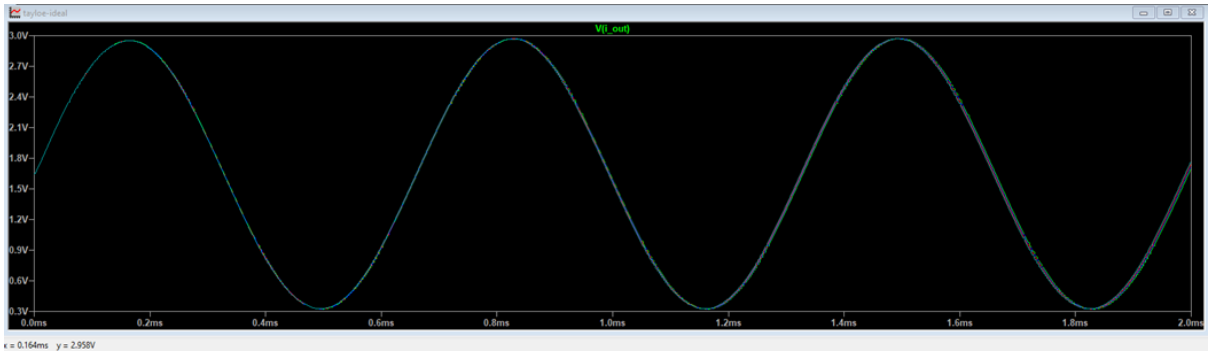


Figure 3.3 Magnitude of circuit with maximum voltage annotation

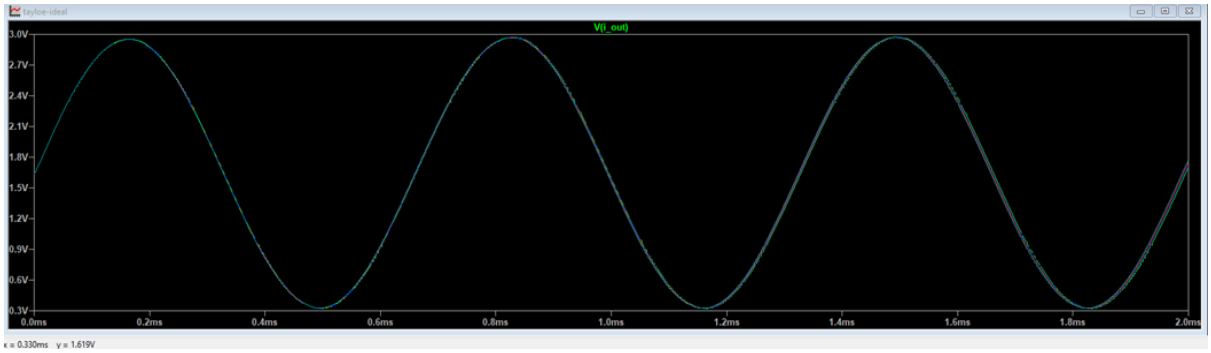


Figure 3.4 Magnitude of circuit with centre voltage annotation

After the mixer + audio amplifier, the signal then will be input through the ADC (analog-to-digital converter). Hence, the matched voltage is required (around 1V). To obtain the voltage, we need to calculate the root mean square of the value by using the following equation:

$$V_{rms} = \frac{V_p}{\sqrt{2}}$$

where Vrms is root means square of voltage in V, Vp is the peak voltage from the origin in V.

In this case, Vp = 2.958 (figure 3.3) – 1.619 (figure 3.4) = 1.339 V, so our Vrms will be 0.9468V which is close to the target 1V. This also provides proof of feasibility of our circuit.

[4] audio amplifier schematic and results

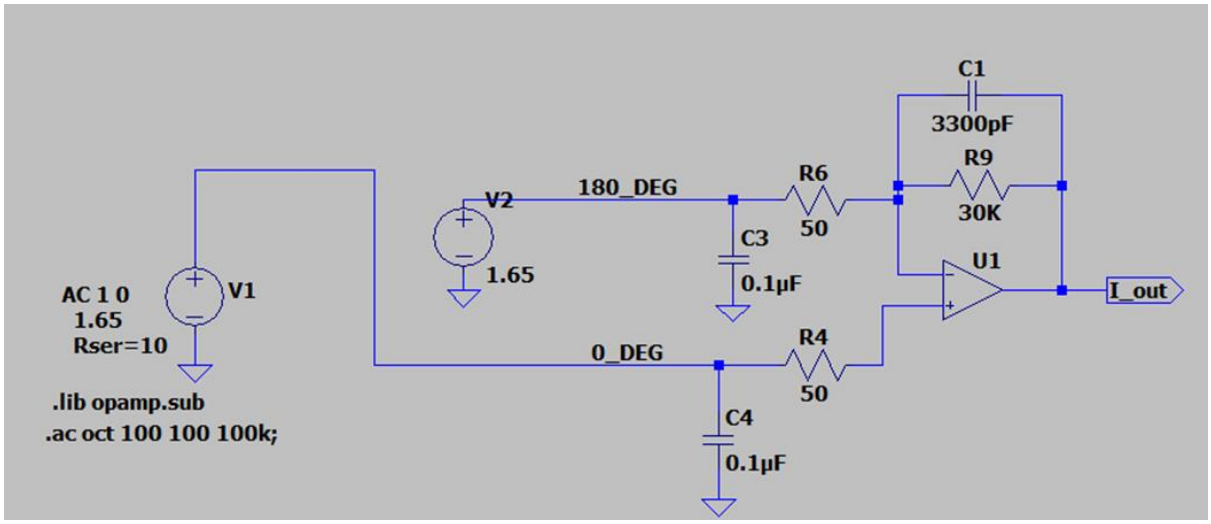


Figure 4.0 audio amplifier circuit simulation in LTspice

The audio amplifier is composed of two sets of low pass amplifiers. (figure 4.0 is only one set) The reason why two low pass amplifiers are used is because we try to filter out the higher frequency from the inputs. The left frequency will be in the range of 1.4kHz to 1.6kHz which is expected.

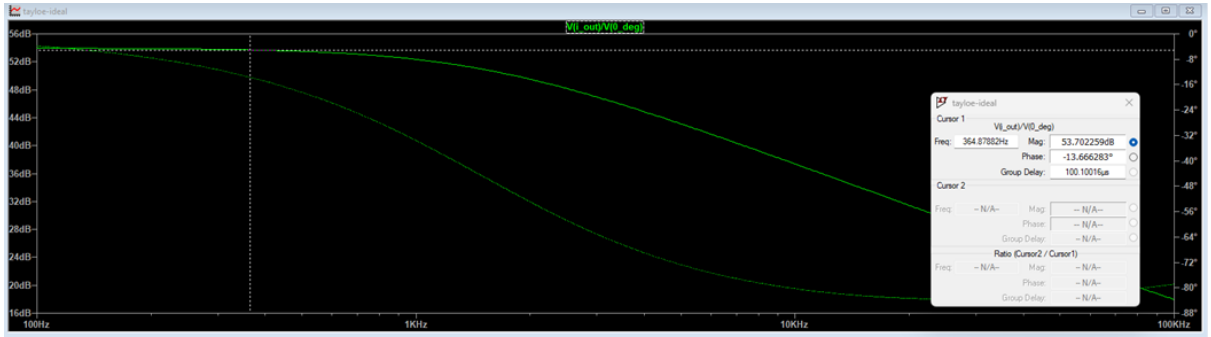


Figure 4.1 Magnitude and frequency relationship of circuit with highest gain

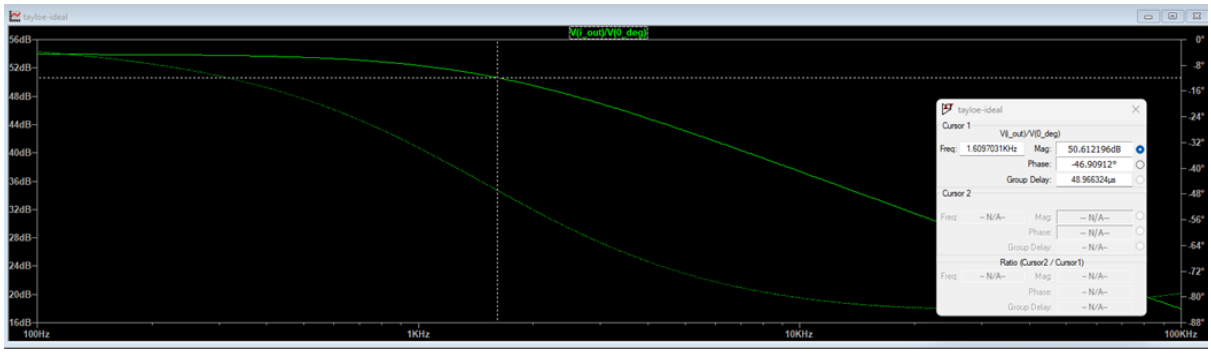


Figure 4.1 Magnitude and frequency relationship of circuit at 3 dB cut-off frequency

As the results showed (figure 4.0 and 4.1), the cut-off frequency is 1.609kHz which satisfies the requirements – we want to keep 1.4kHz to 1.6kHz. Therefore, it has been proved to be a good amplifier and low pass filter.

[5] ADC schematic and results

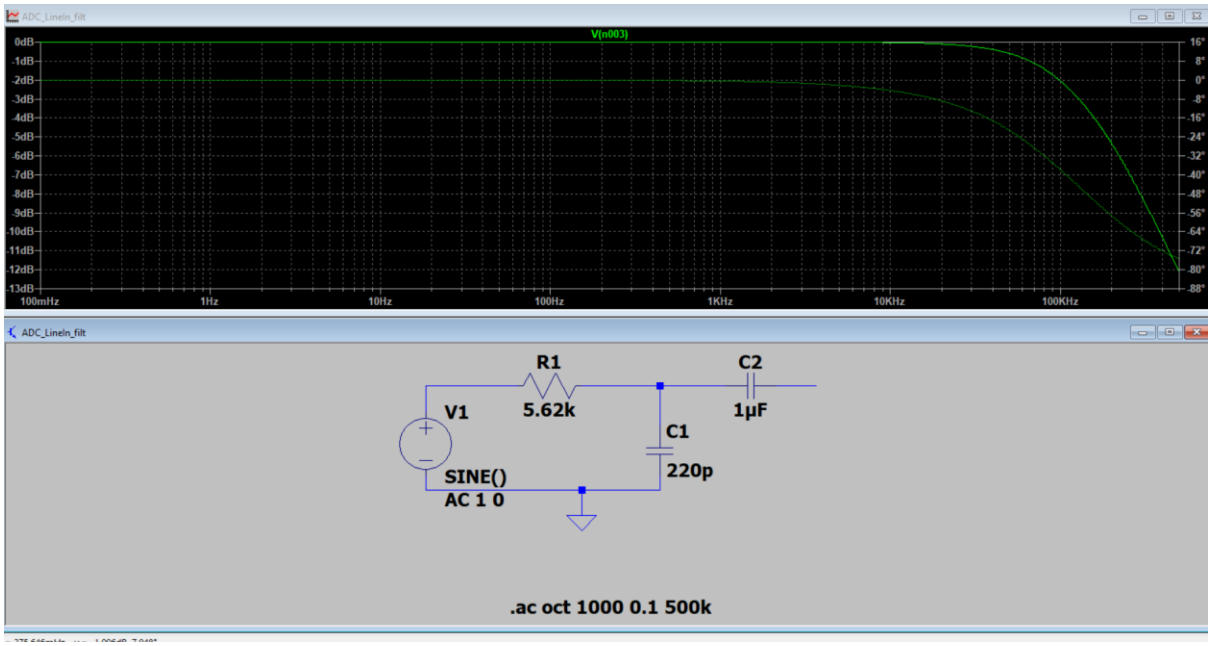


Figure 5.0 Magnitude and frequency relationship of an ADC line in filter

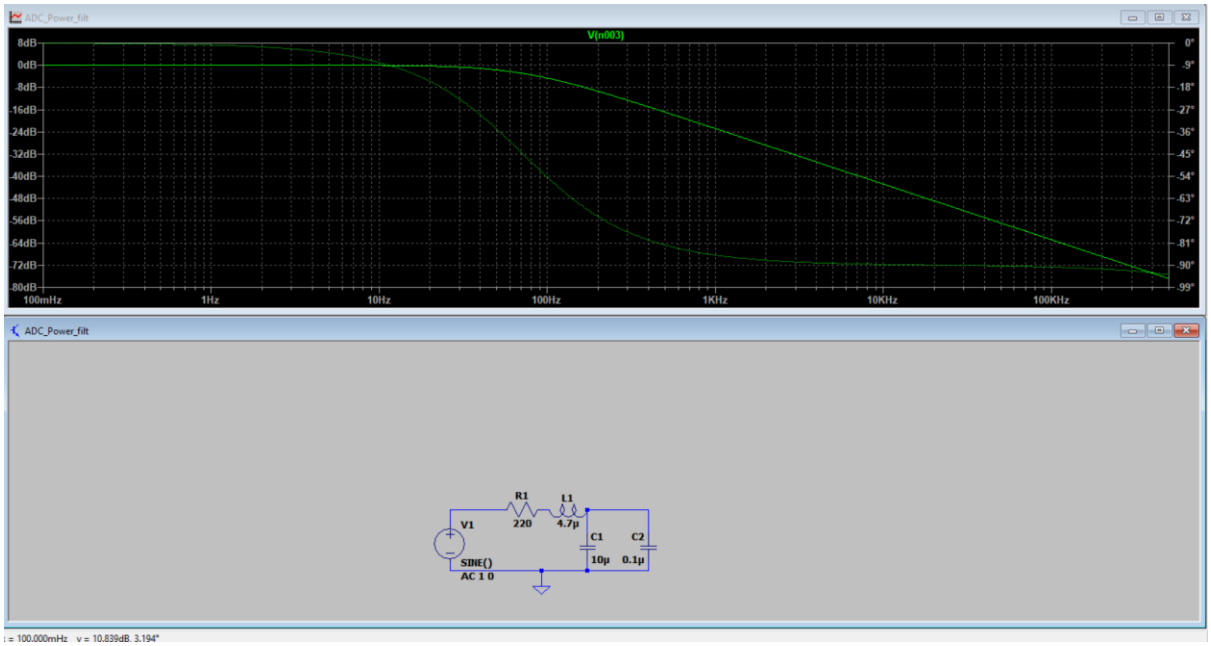


Figure 5.1 Magnitude and frequency relationship of an ADC power filter

The ADC chip was selected due to its simple interface of I2C and I2S which the raspberry pi supports through pin 24 and 26 (SDA4 and SCL4) and pin 12, 35, 38 (PCM pins) respectively. Its 2 left right inputs for ADC are sufficient for our use case and 90dB SNR is also good. Its sampling rate can go up to 96kHz but we would adjust it down to 12kHz because that is what our pulse audio file/decoder records at (refer to parec.c file in src). For interfacing, it requires a clock input from either a crystal or a master clock. For this, we used the spare clock output (CLK2) from the Si5351 and with less than 50ps of jitter, it meets the specification of the SSM2604. This saves power as we can power down the oscillator inside the ADC and uses the Si5351 as the master lock instead. Its internal preamp is also handy if we happen to over amplify our signal or under amplify it (1V rms is recommended by the datasheet).

Unfortunately, we could not find the SSM2604 model in LTspice to test it. Instead, we based our implementation and interfacing as close as possible to the Typical Application Circuit and Evaluation board of the manufacturer (Analog) to make sure it will work. Some tests on the power filter and line in are presented in the simulation result to understand why the manufacturer implemented it this way.

One of the problems we encountered when implementing this solution is that since the ADC chip is so small and has so many filters, it is completely surrounded by resistors and capacitors. Thus we have to use vias to route the clock to the ADC and a 30 ohm resistor to terminate it in series. It is a similar situation with the SCL and SDA line to the clock generator.

Another problem is the ADC requires a digital voltage supply which needs to be much more stable than an analog one. To solve this, we voltage divide the 5V pin into 3.3V to power all other components (op-amp, tayloe mixer, clock generator) and try to limit the 3.3V pin exclusively for the ADC. The clock generator also uses both the voltage divided 5V pin and 3.3V pin as it requires two 3.3V supply (refer to schematic), but with only 2 components using 3.3V pin and filtering at the

digital voltage supply pin (DVDD), we think it would be sufficient for the ADC. Furthermore, the 5V pin can provide much more power to other components compared to the 3.3V pin so using only the 3.3V line would not be a good idea anyway.

[6] SDR2.0 schematic and results

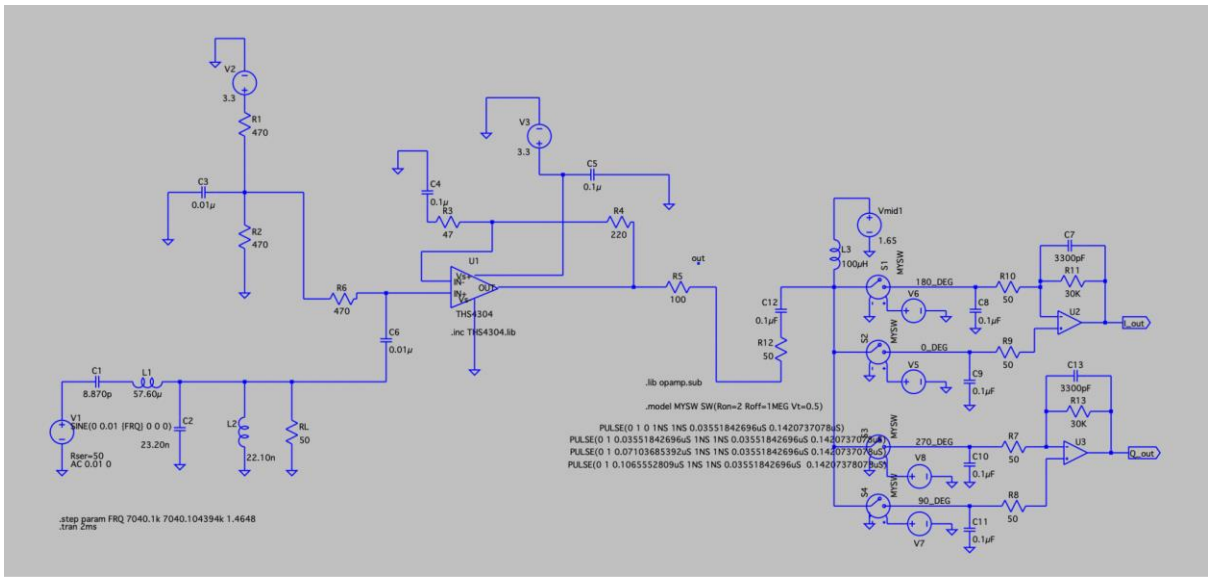


Figure 6.0 SDR2.0 circuit simulation in LTspice

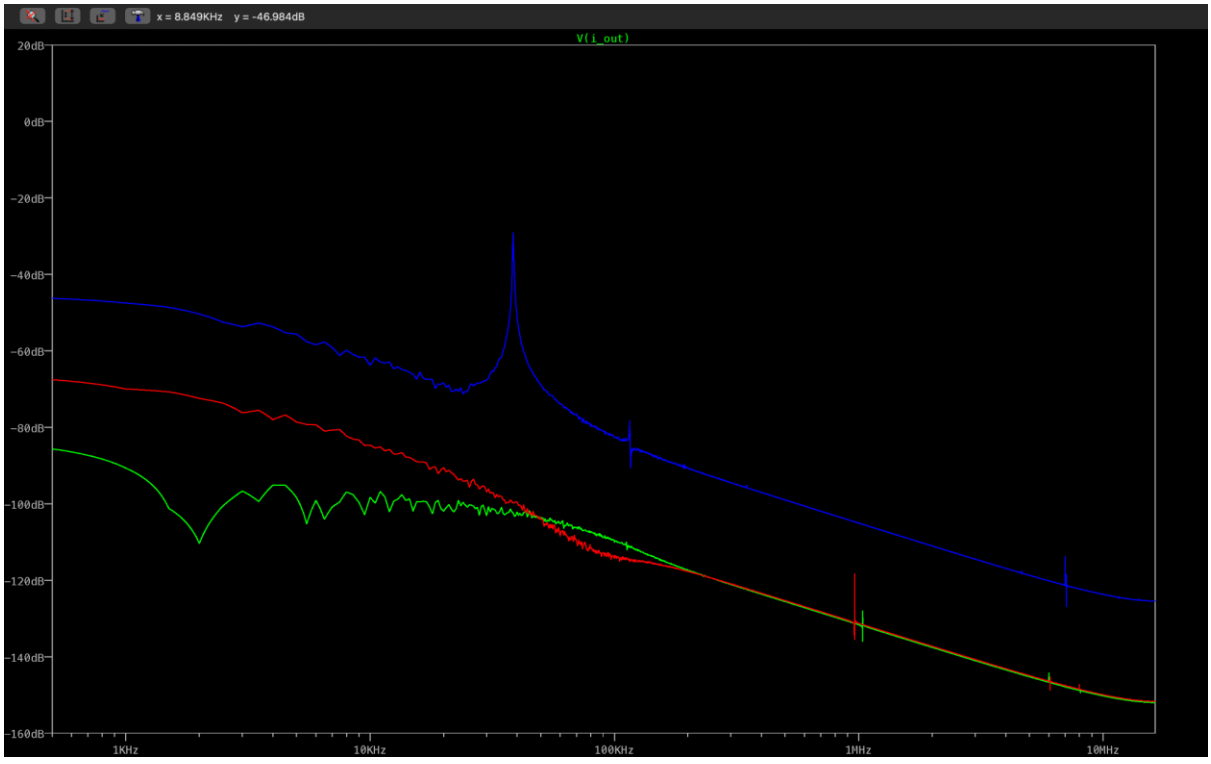


Figure 6.1 Transient analysis of frequency on 6Mhz (green), 7Mhz (blue) and 8Mhz (red)

As figure 6.1 showed, only the 7Mhz (blue) can pass through the entire circuit. The spike shape indicated that only that range which was 6.7901MHz – 7.2901 MHz could go through the circuit.

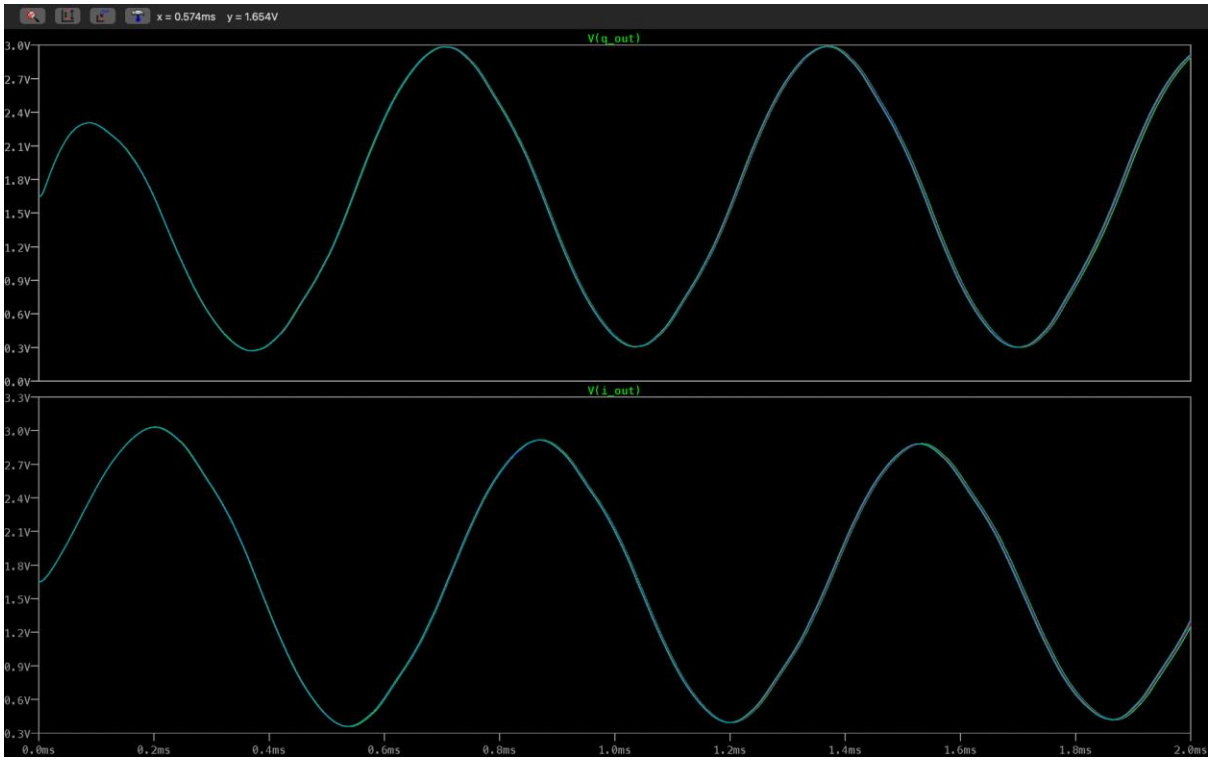


Figure 6.2 Transient analysis of frequency range from 7.0401 MHz to 7.040104394 MHz with 1.4648Hz step increment in 90 degrees phase shift.

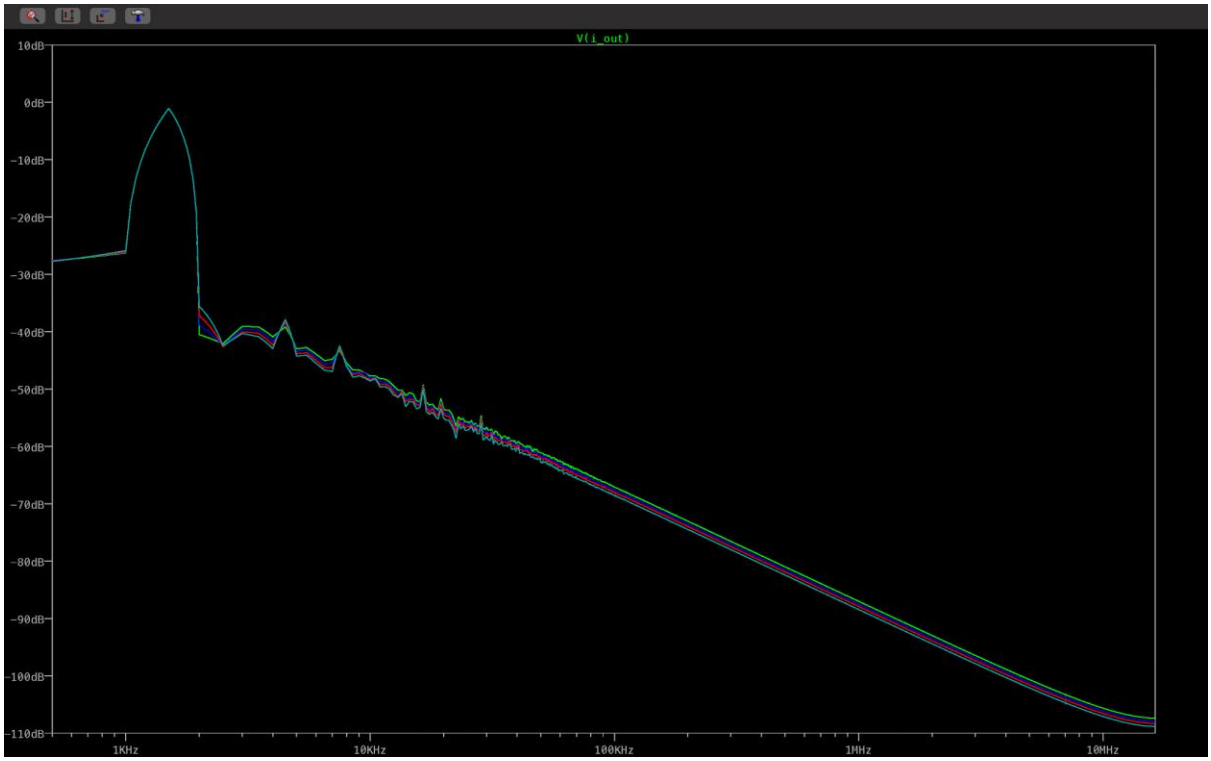


Figure 6.3 The final output of circuit.

Based on figure 6.2 and 6.3, it showed that the circuit produced 0.94 V and 1.4 - 1.6 kHz which were expected results for our project.

[7] SDR2.0 pcb results

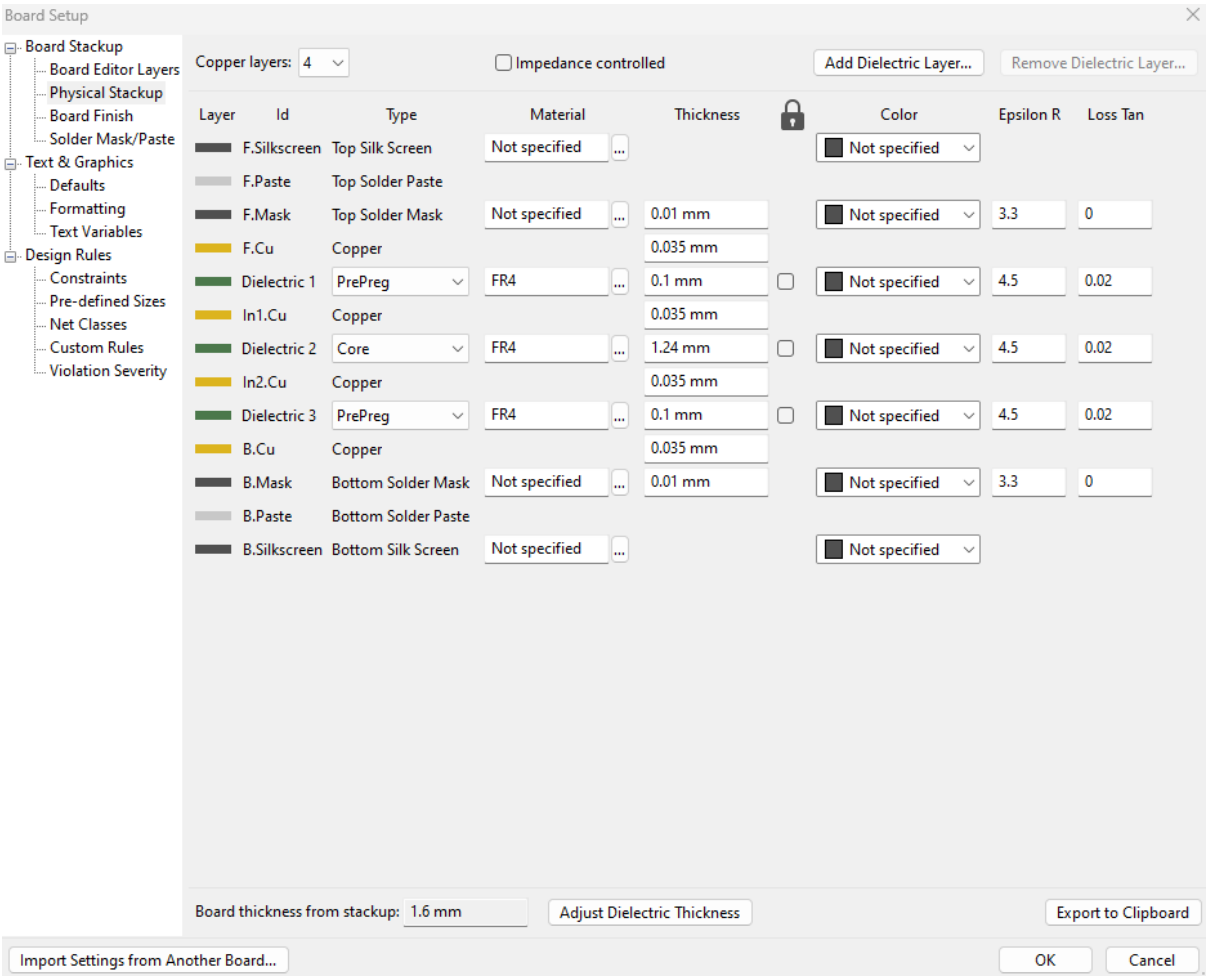


Figure 7.0 SDR 2.0 pcb board settings

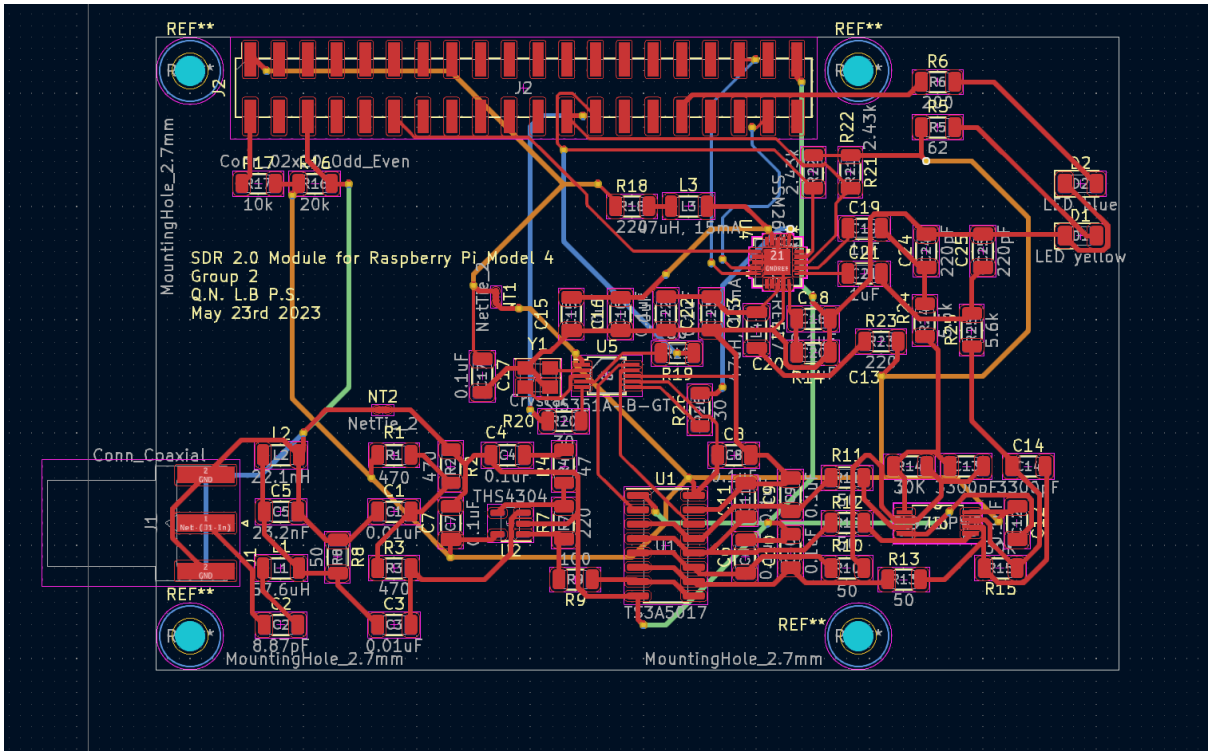


Figure 7.1 pcb tracings

After a thorough research on the internet, 4 layers pcb was chosen to use to implement our SDR2.0. The complexity of the circuit was considered to be high due to large amounts of components being on the same board. This led to potential radiation and interface generated by the currents in each component. As a result, 4 layers pcb would have better isolation and reduction to those effects. The four layers (figure 7.0) were F.Cu (red tracing), In1.Cu (light green), In2.Cu (orange) and B.Cu (blue) (figure 7.1). These layers had their own functionalities which were signal, ground, power supply and signal.

The dimension is also designed to match the raspberry pi model 4's mounting holes and pin header.