Design Project 2 Report Template

From: Marina Bellido Rodriguez

Teammates: Lillian Vernooy and Nina Jobanputra

What's the secret message?

The bits received after the pilot frame were 01001000 01001101 01000011. The remaining bits represent 8-bit ASCII encoded characters, which translate to the message: HMC

Website used to decode: https://www.ascii-code.com/

72	110	48	01001000	H	H		Uppercase H
77	115	4D	01001101	M	M		Uppercase M
67	103	43	01000011	C	C		Uppercase C

What was the maximum range at which you received the message?

The maximum range at which we received was 77.724 meters. We could have kept on going, since the signal was very strong at that distance, but we ran out of hallway length to test. We didn’t want to move the equipment in case we broke it, so we decided to stop there.

Picture of your setup at 3m range:

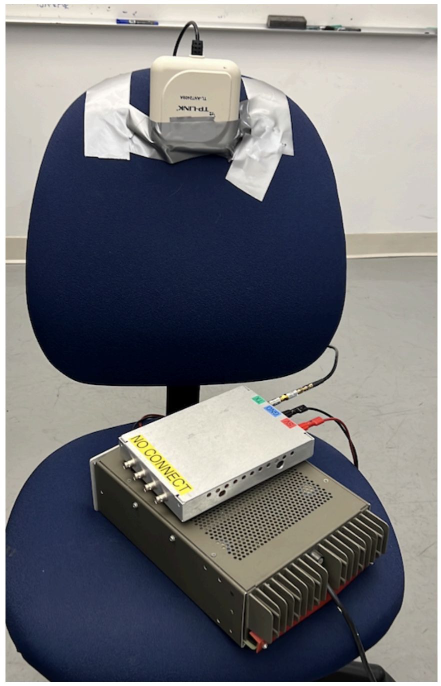

Figure 1: Transmitter for the 3 meter range.

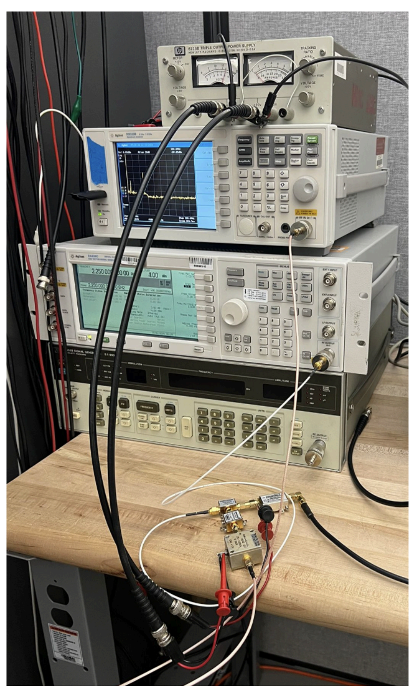

FIgure 2: Set up for the Receiver.

Figure 3: Three meter Range.

Oscilloscope trace of your receiver output at 3m range

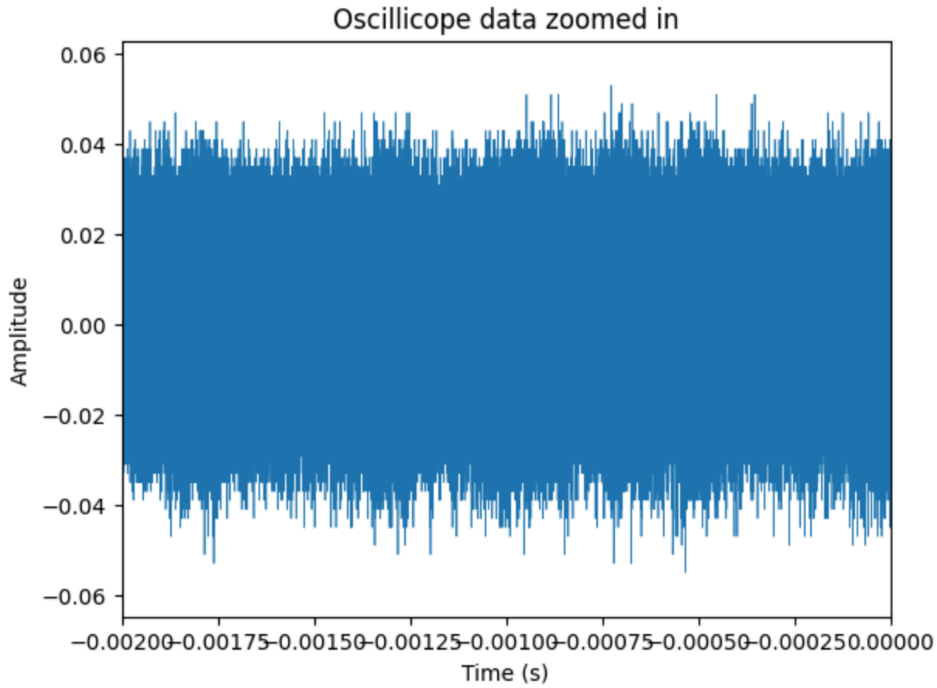

Figure 4: 3 meter scope csv trace plotted in python.

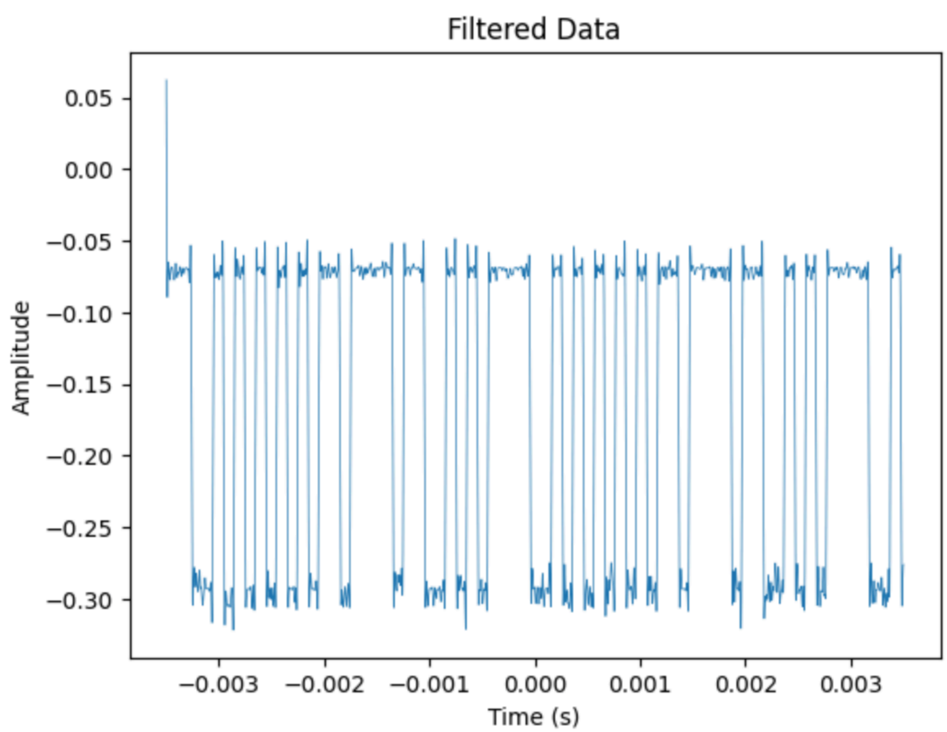

Figure 5: Digital filter applied to the previous graph's data.

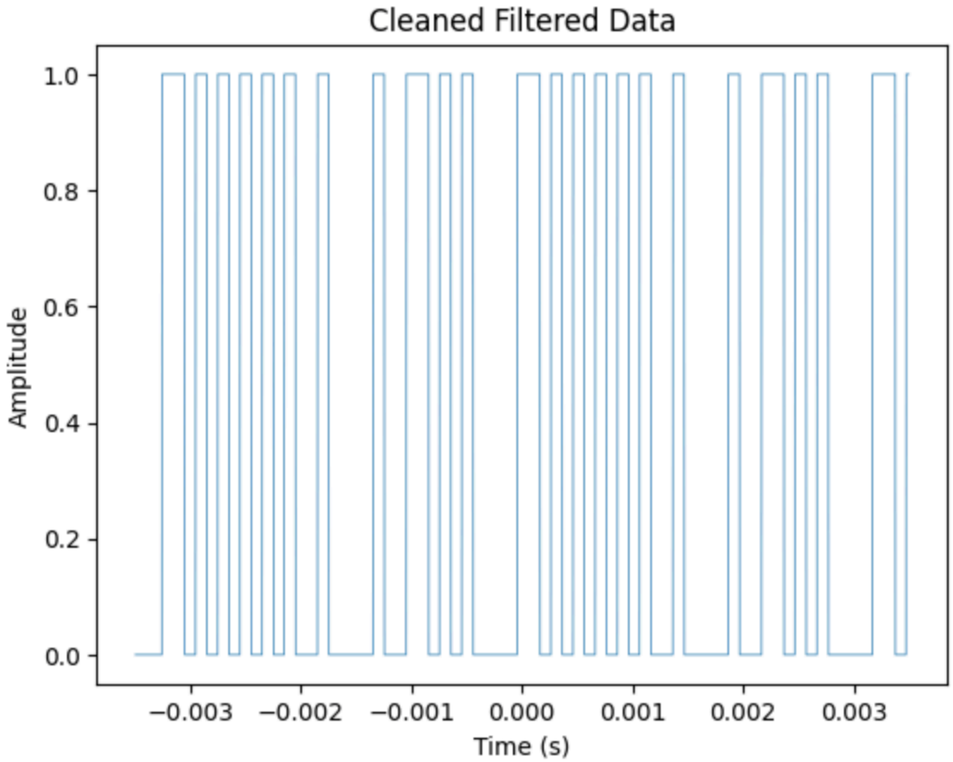

Figure 6: Digital filter data after being cleaned.

Picture of your setup at max range:

Figure 7: Set up of the transmitter at max range recorded

Figure 8: Set up of the transmitter and receiver antenna at max range recorded.

The receiver antenna was connected to the rest of the receiver using a long transmission line cable.

Oscilloscope trace of your receiver output at max range:

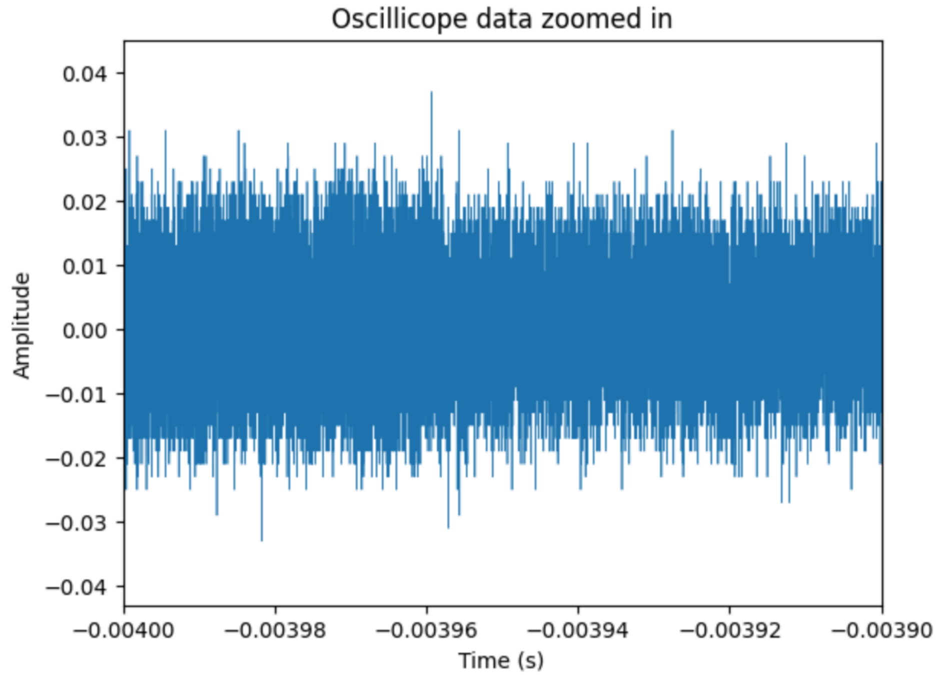

Figure 9: Max range scope csv trace plotted in python.

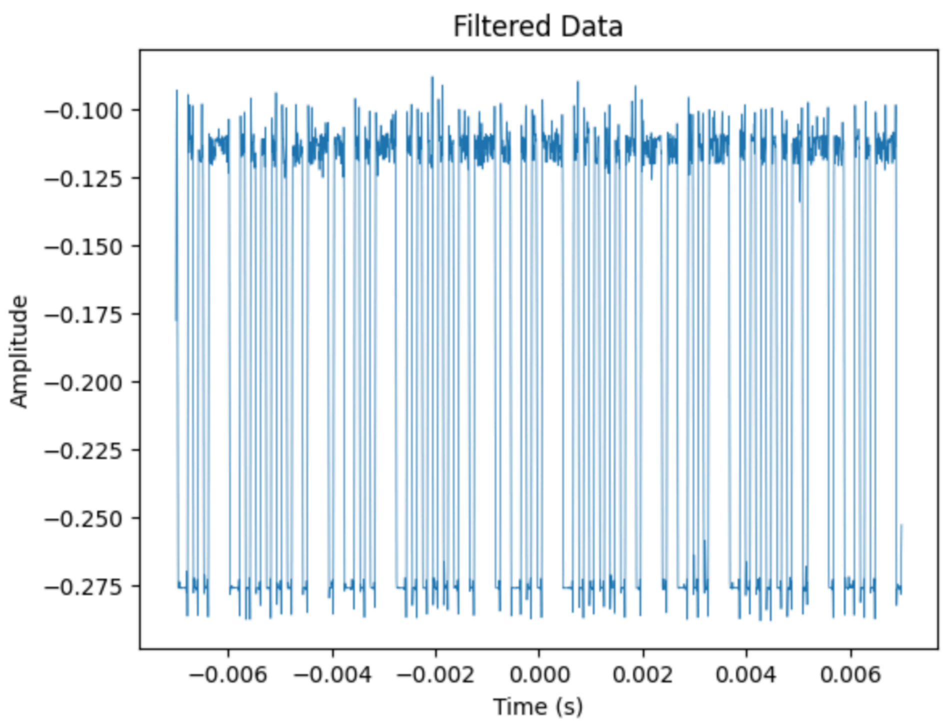

Figure 10: Digital filter applied to the previous graph’s data.

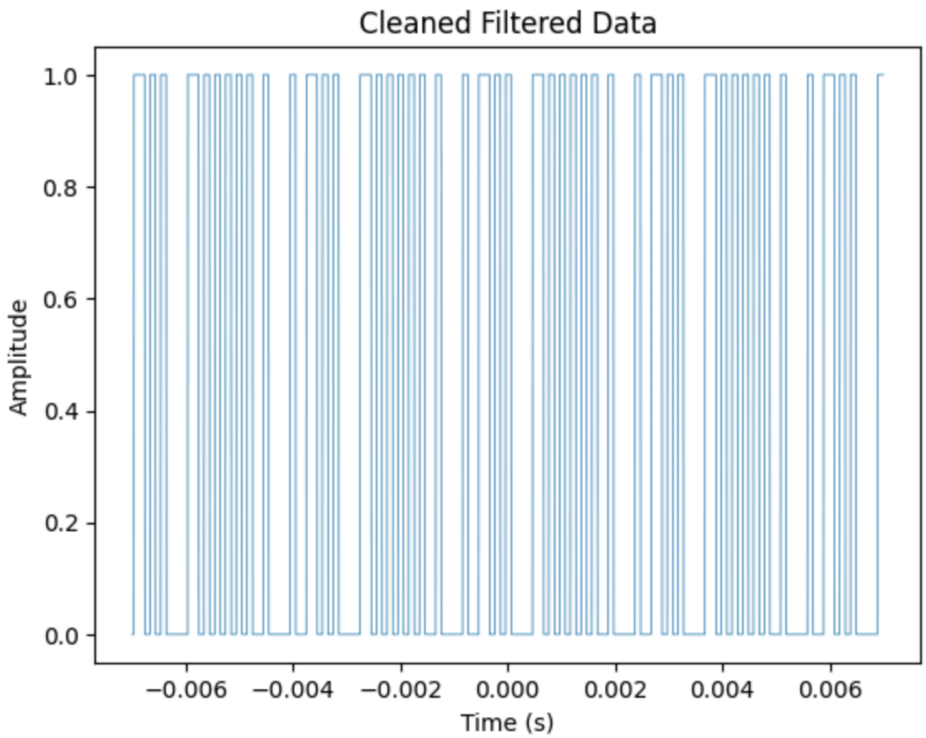

Figure 11: Digital filter data after being cleaned.

One page discussion of how to decode the oscilloscope trace to extract the secret message, possibly using an annotated copy of the oscilloscope trace to help explain your decoding scheme.

The receiver is connected to the oscilloscope and the scope samples at a max of 1 GSa/s, so this is more than sufficient for the 46 MHz upper tone. The sample memory is 14 Mpts, so at 1 GSa/s this is about enough for 18 bytes at 10 kbps. Since the transmitter repeats a message continuously at a bit rate of 10kbps, each bit is 1µs long.

Figure 12, shows an example trace of the oscilloscope which allows us to demodulate based on amplitude. There is a clear distinction between two amplitudes that correspond to the two different frequencies. The lower amplitude corresponds to the higher frequency as it will be more attenuated by the low pass filter's slope, and also all of the cabling and system will attenuate higher frequencies.

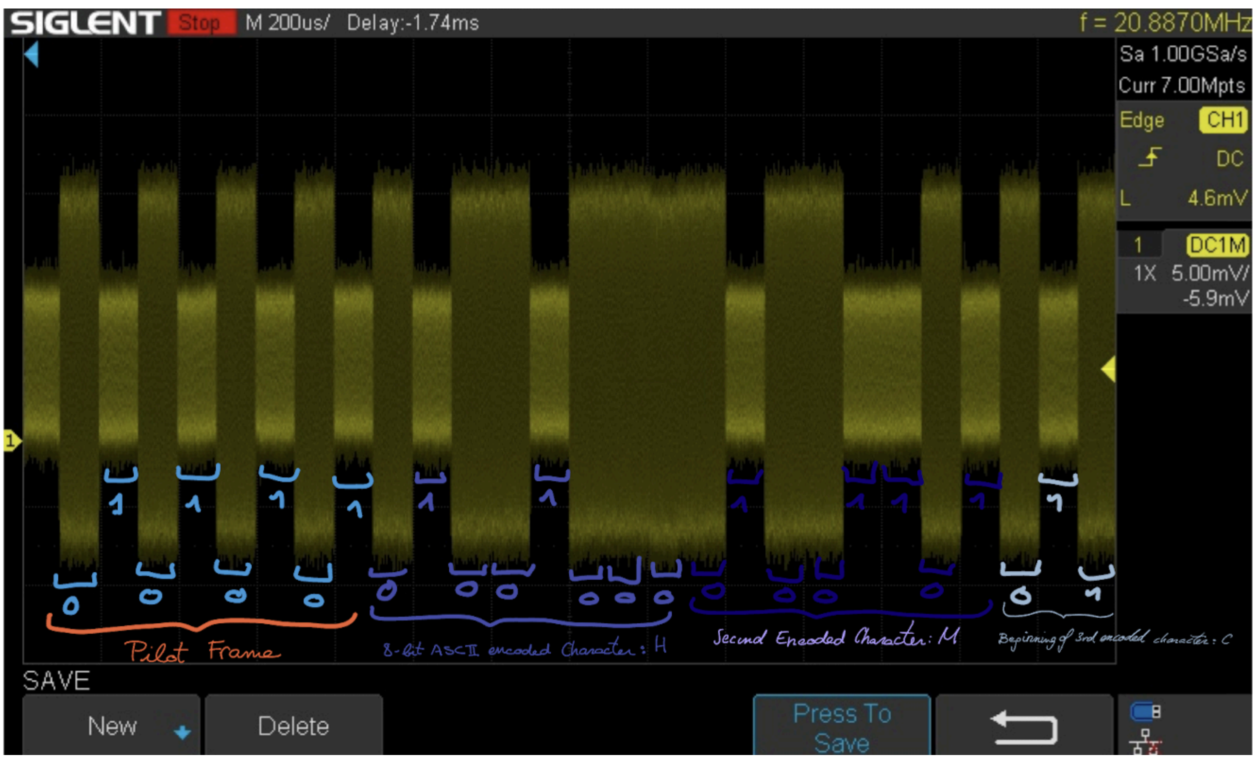

Figure 12: Decoding using the Scope trace

This would be a decoding method, however, my team decided to process the data through a digital filter to make the signal cleaner and easier to read. The digital filter takes the data in Figure 4 and 9 and processes it from real to a complex IQ signal with real and imaginary values. Then we take pairs of adjacent angles and determine the phase angle between adjacent samples. This gives us new points where if the frequency is low then the relative phase is small and vice versa. After that frequency demodulation was applied and another low pass filter (Butterworth) to clean the signal. The final signal can be seen in figure 6 and 11. In the digital filter the polarity was also swapped so the high represents a 1 and the low a 0.

Picture of your receiver

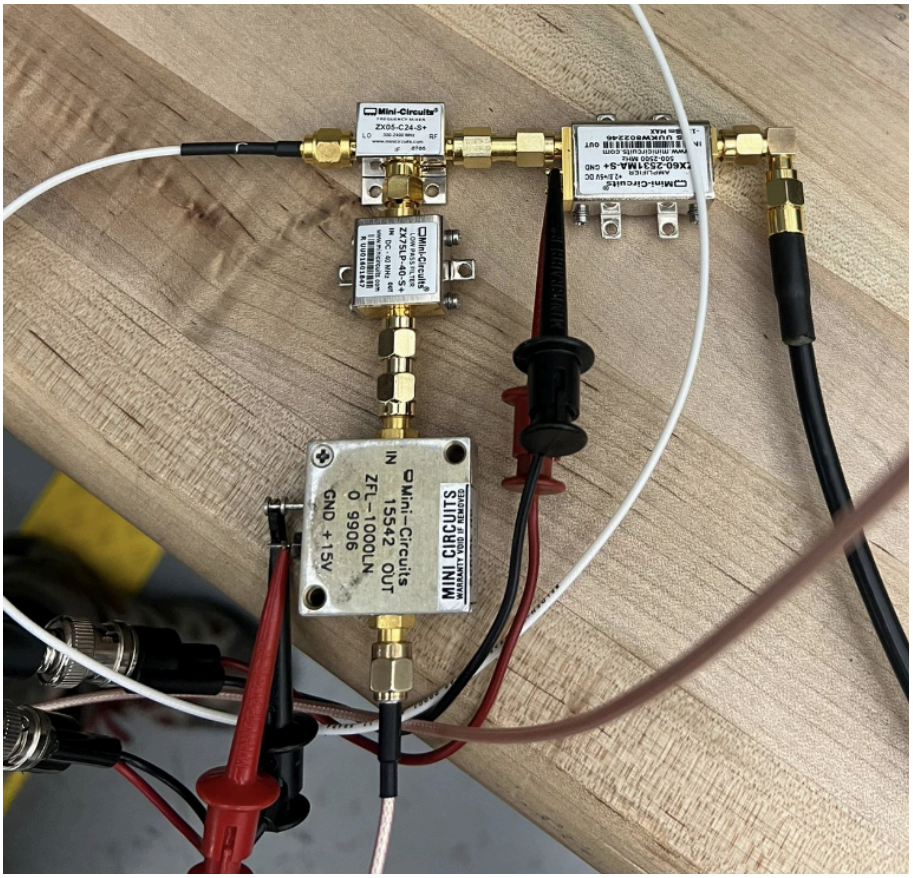

Figure 13: Receiver after receiver antenna

The receiver is a direct downconversion + direct sampling receiver built. We take the TP-link patch antenna, feed into a ~35 dB amplifier, mix with a 2.25 GHz LO (so that the transmit tones end up at about 6 MHz and 46 MHz), and feed through a 40 MHz low-pass filter to cut out the WiFi signals that end up around 140 MHz and above. Finally, we feed through a low noise ~20 dB amplifier, and then feed the output into the digital filter (explained in the previous page what it does).

Schematic of your receiver drawn in Powerpoint or some other illustration program

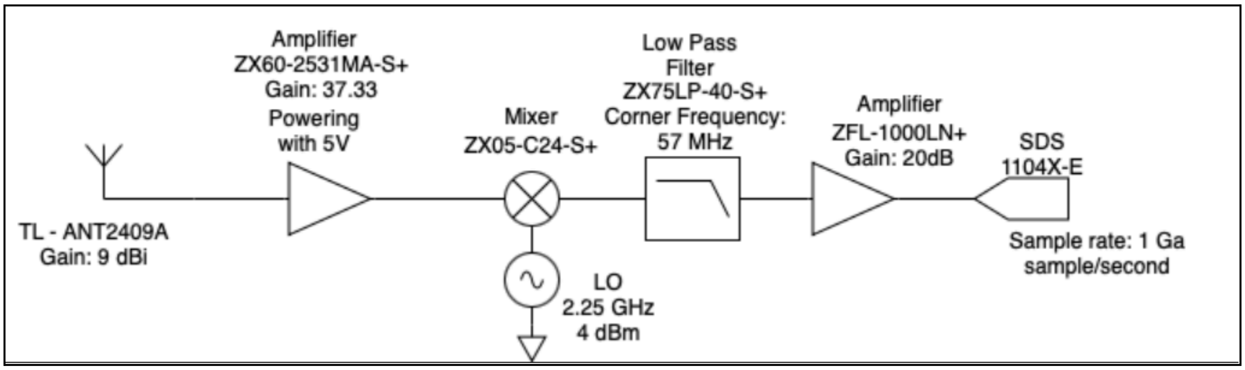

Figure 14: Schematic of receiver

Output spectrum from each stage of your receiver with cursors showing signal power, noise floor, and distortion products. You may include multiple spectra on one page if you can fit and label them well.

NEW TRACES FOR THE SIGNAL POWER: 3 meters range

Input to mixer: Freq 2.25GHz and 4dBm

Type	Trace
Amp + Mixer +LPF + **Low Noise amp.** output:	33
LPF output	34
mixer	35
amp	36
Antenna alone	37

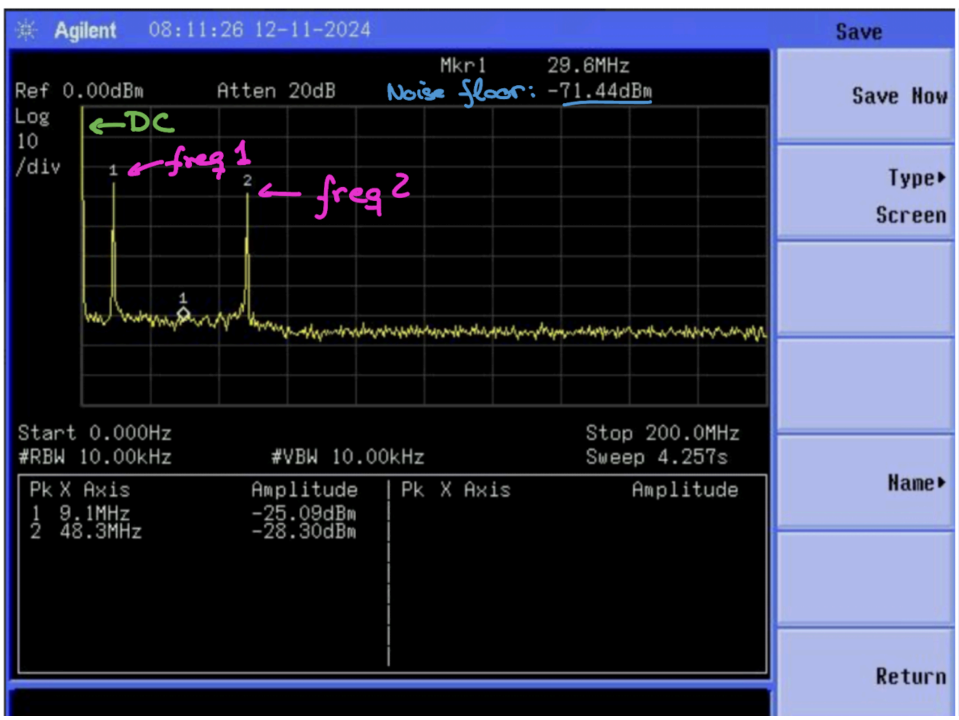

Figure 15: Trace 33 - The spectrum of the Low Noise Amplifier output

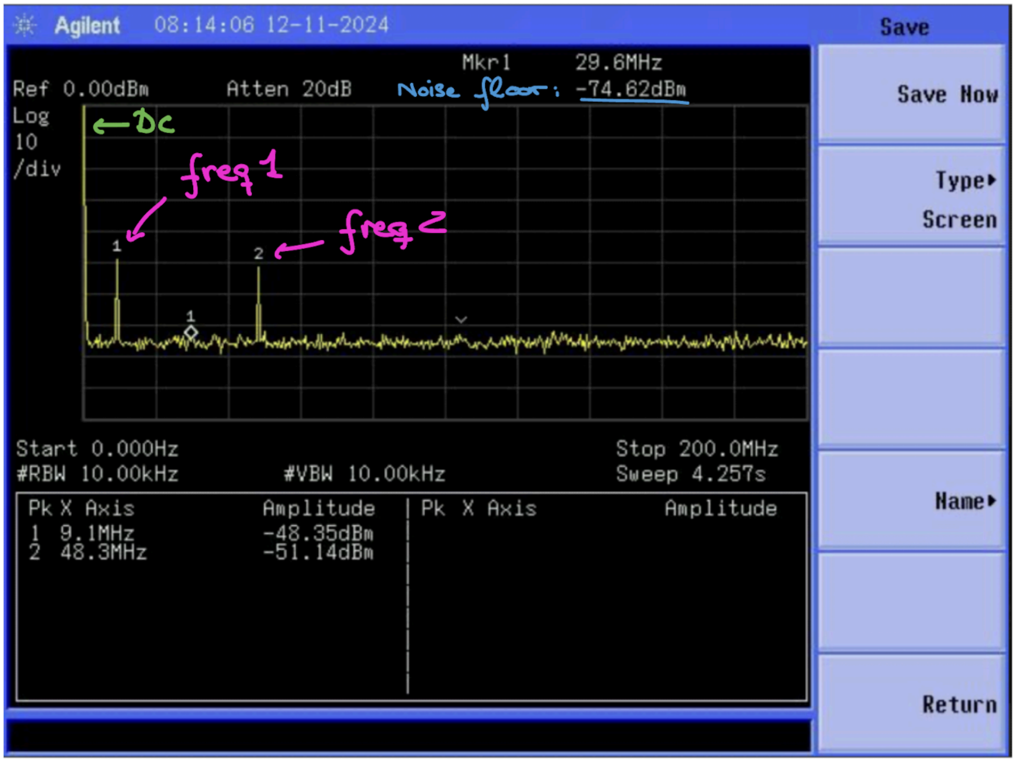

Figure 16: Trace 34 - The spectrum of the Low Pass Filter Output

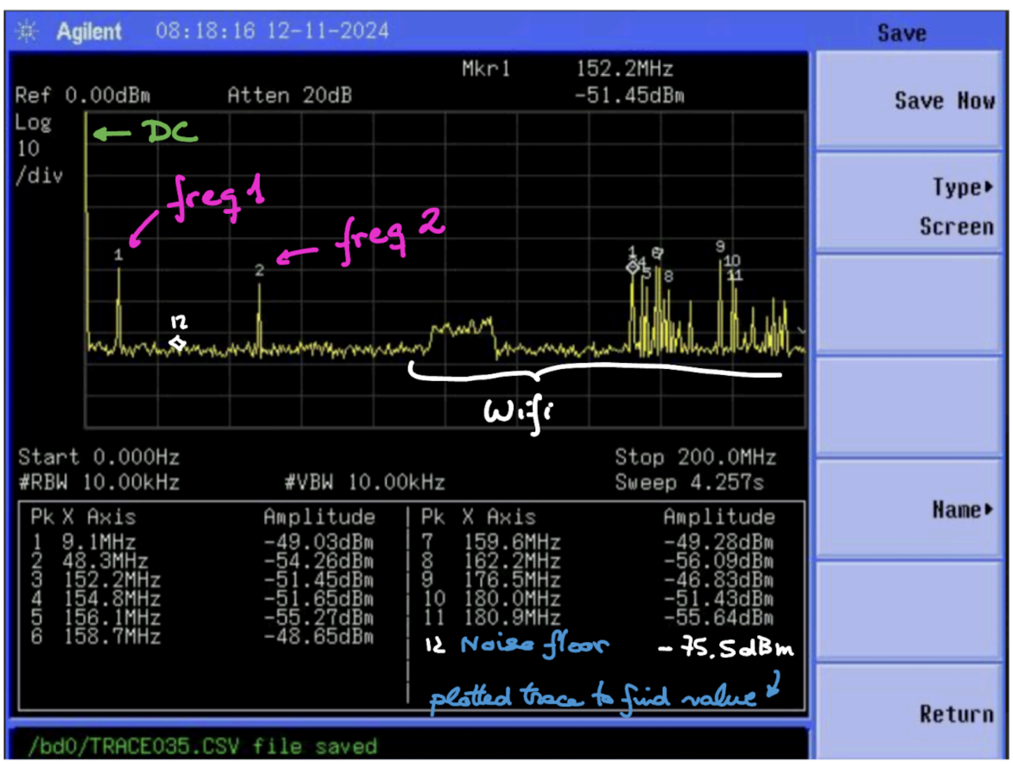

Figure 17: Trace 35 - The spectrum of the Mixer Output

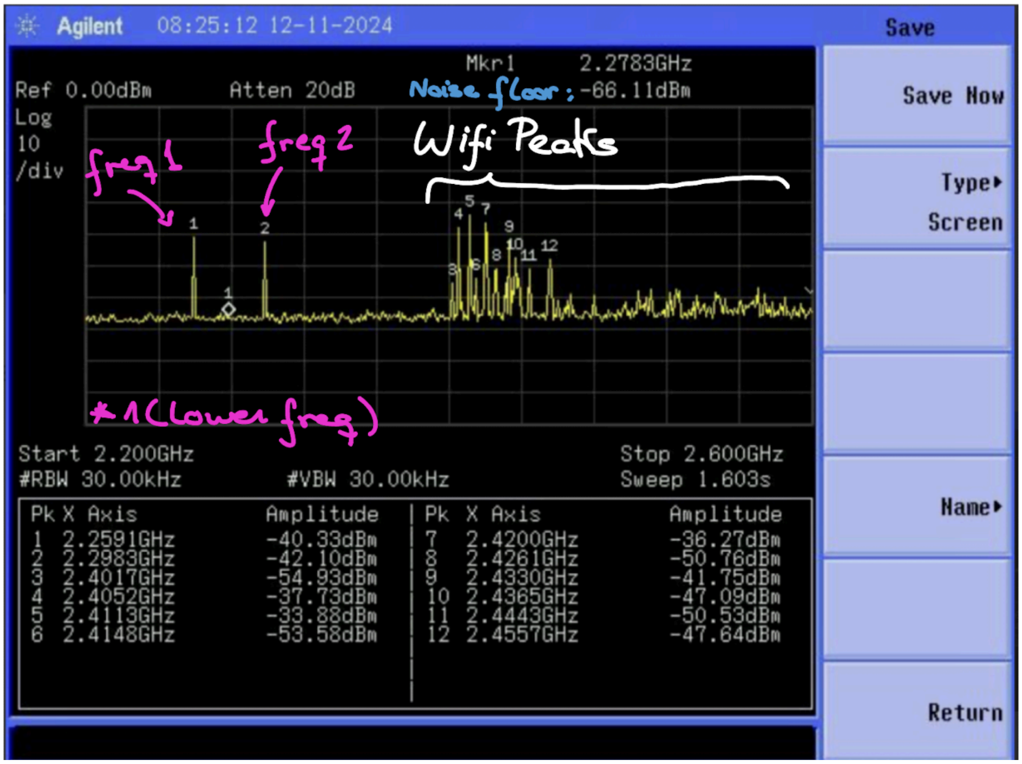

Figure 18: Trace 36 - The spectrum of the Amplifier output

Figure 19: Trace 37 - The spectrum of the Antenna Output

DISTORTION PRODUCTS:

(make sure to show an example trace) and grew 2.296GHz power of -35dBm -> dual tone

Type	Trace
1st **amplifie**r output:	17
Amp + **Mixer o**utput	21
Amp + Mixer +**Low Pass Filter** output:	25
Amp + Mixer +LPF + **Low Noise amp.** output:	31

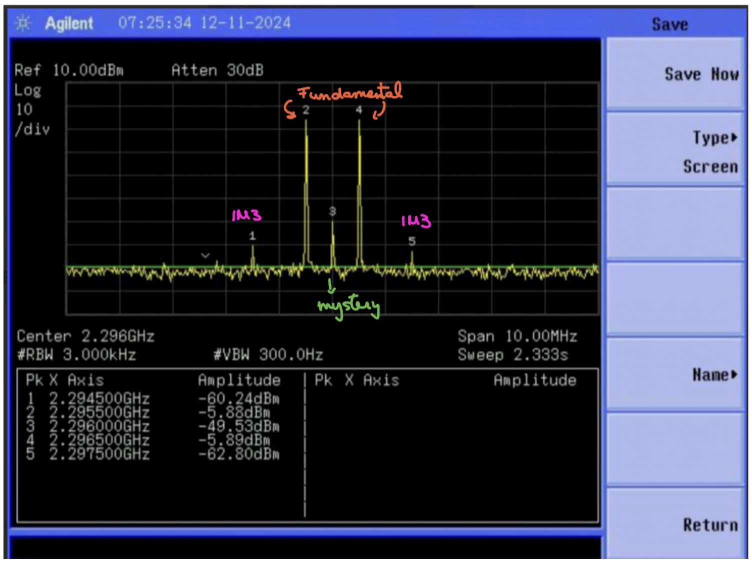

Figure 20: Trace 17 - The spectrum of the Amplifier output

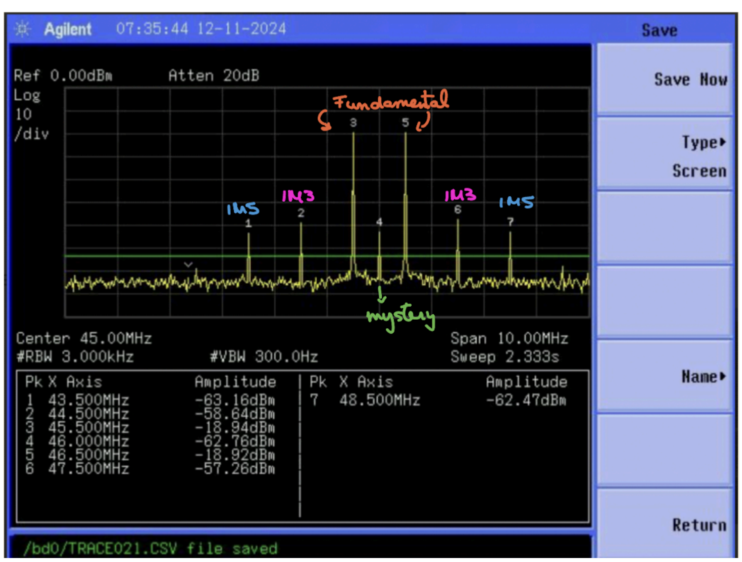

Figure 21: Trace 21- The spectrum of the Mixer Output

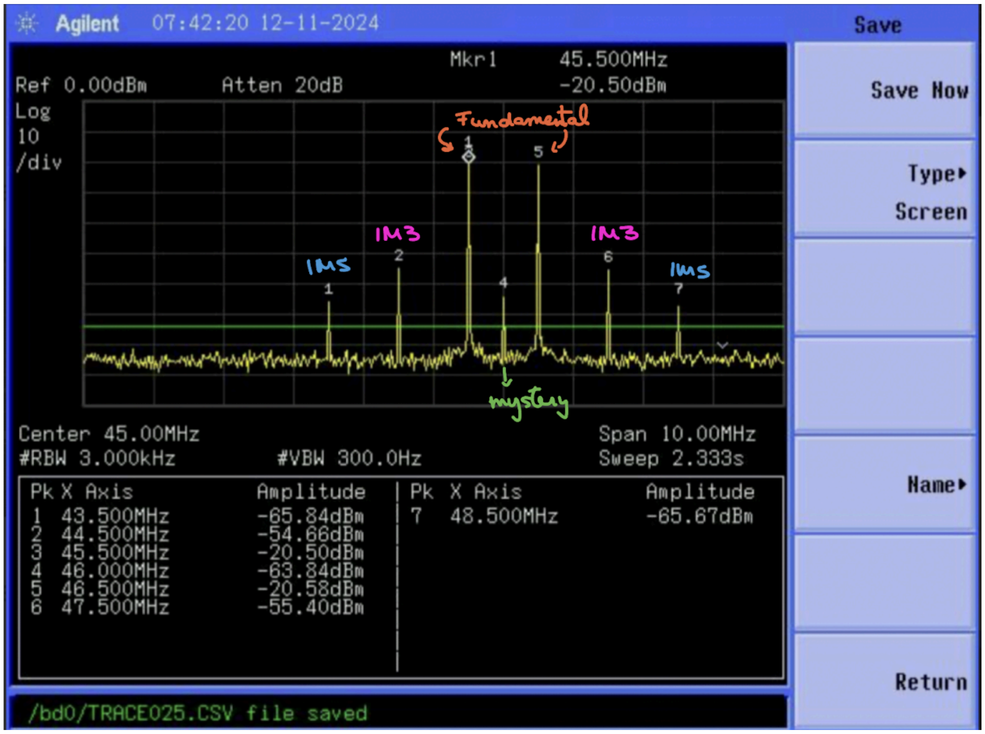

Figure 22: Trace 25 - The spectrum of the Low Pass Filter Output

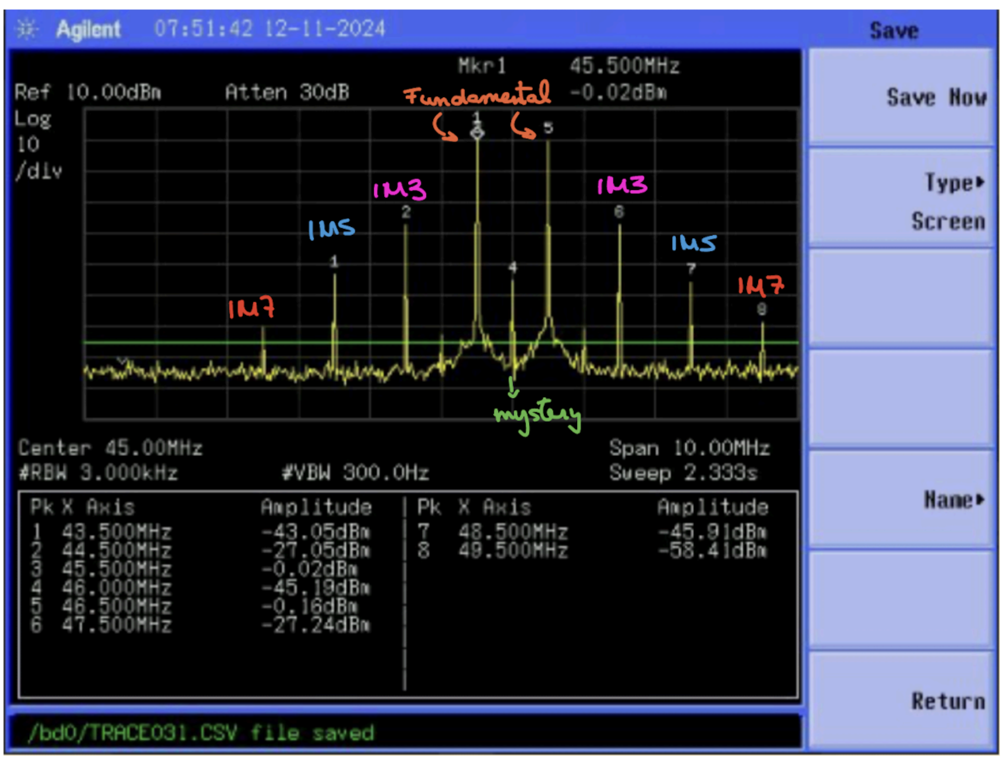

Figure 23: Trace 31- The spectrum of the Low Noise Amplifier output

S11 of your antenna. You don't need to extract the antenna gain using the anechoic, but extra credit is available if you do (and document the process).

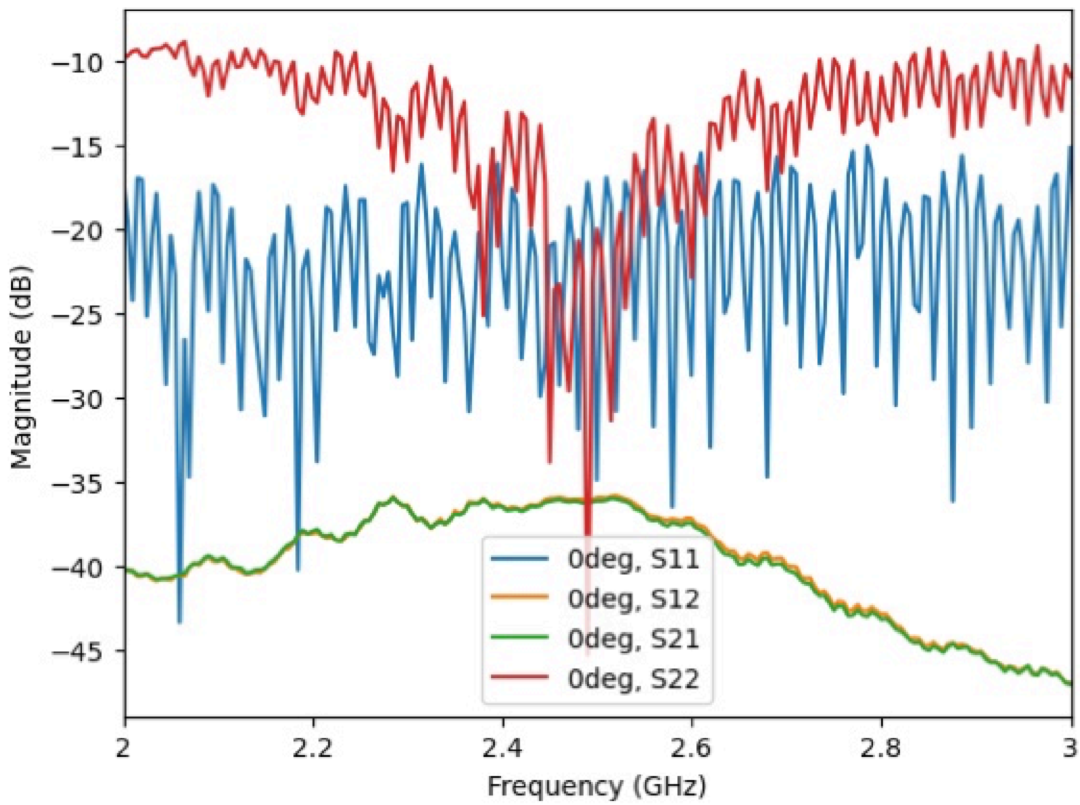

Figure 24: Graph from lab 5 where we characterized the TP-link antenna.

The S22 of the graph is the S11 of the TP link.

In figure 24, the S22 represents the S11 of the wifi patch antenna. This is because during set up the TP link was set in the second port. Nevertheless, we get the same information as if it had been attached to port 1. We selected 2.3 Ghz as it is very close to both of our frequencies and found its S11 to be around -15.98dB.

By observing the graph, the resonant frequency of the TP is at around 2.5GHz instead of 2.4GHz which was the expected value in the datasheet. This means that at our frequencies the antenna should be less efficient at capturing the signal due to a larger impedance mismatch and less energy being transmitted to the receiver. This seems to agree with the value we obtained for S11 as it is different from the value at resonant frequency.

One page explanation of how you found the theoretical signal, noise and distortion levels for your receive chain. Also upload an Excel spreadsheet that includes these calculations for my review. Include both your analysis and your measured data in the spreadsheet.

Calculations spreadsheet

Signal Power:

To find the theoretical signal power we used the log version of the Friis Equation:

$$P_{RX} = P_{TX} + G_{TX} + G_{RX} + 20log(\frac{\lambda}{4\Pi r}) = \text{-70.203 dBm}$$

Equation 1

For equation 1, the transmitter has -29dBm of Output Power, poor antenna matching (~-10dB return loss), and modest (9dBi) directionality. The distance we are calculating for is 3 meters and the frequency used was 2.296 GHz. With the Prx value obtained we then added all of the gains and losses in our receiver. The gains are found in the datasheet and also schematic of the receiver (Figure 14) and the loss was found in the datasheet. The mixer has an insertion loss of 6.47 dB at a frequency of 2292.4MHz and the Low Pass filter has an insertion loss of 1.62dB at 50MHz. We looked at trace 33 to observe the measured output power of the receiver.

To find out the Grx, we look at the Antenna Datasheet. We observe that it is right around 5dB but it is a higher frequency than ours, so it will be slightly different. Another team obtained 3db in lab 5, so we will be using that value.

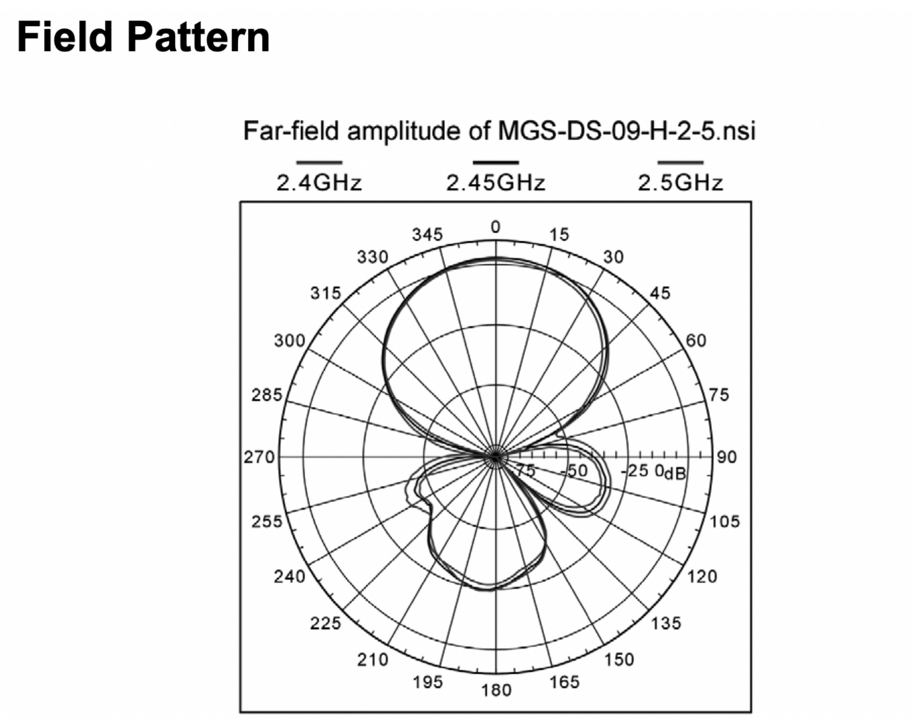

Figure 25: Field pattern for our antenna from the datasheet.

Finding the power of the receiver antenna*			
	Grx	3	dBi **
	Gtx	6	dBi, ***
	Ptx	-29.15689365	dBm
	lambda	0.1304347826	Using the c/f equation
Friis equation	Prx	-69.37564765	dBm

*Using the high frequency value instead of the low frequency value
** This is the value that other teams obtained in lab 5 and this makes more sense than the -5 which the number shouldn't be negative
*** Conflicting with the datasheet and the instructions. We choose the datasheet value but if we were to go with instructions we would use 6 instead.

Finding the Expected output power of the receive chain				
	Gain (dB)	Loss (dB)	Gain unlogged ratio of powers	Loss unlogged ratio of powers
1st amplifier	37.33		5407.543229	
Mixer		6.43		0.2275097431
Lowpass Filter (6MHz)		0.54		
Lowpass Filter (46 MHz)		1.62		0.6886522963
2nd amplifier	23.55		226.4644308	
Expected output power (dBm)		**-16.54564765**		
Expected Corrected output power (dBm)		**-28.95564765**		
Measured Output Power (dBm)		**-28.3**		
Measured Gain of the 1st Amplifier:				
32.37	1725.837892			

When comparing the expected to measured output power, we observe discrepancies. This is most likely due to operating in heavily gained compressed regions of the amplifier, so probably our gains are not right. Thus, we calculated our new expected output power (corrected) with the gain measured in the lab and we obtained an almost exact output power. This leads us to believe that that might be the cause of the discrepancies

Noise levels:

To find the theoretical noise level we first found the noise factor or loss for each component in their datasheet. Then we found the temperature at each component accounting for the gain and losses at each stage and then used this data to calculate the total Noise Voltage Variance [V^2].

$$TA = (nf - 1)T$$

Equation 2: Noise temperature of an Amplifier

$$Tp = (\frac{1}{L-1})T$$

Equation 3: Lossy passives add noise with a temperature of Tp

$$System\ Temp = (G_{amp1}G_{mix}G_{LPF}G_{amp2}T_{in} + G_{amp1}G_{mix}G_{LPF}G_{amp2}T_{amp1} + G_{mix}G_{LPF}G_{amp2}T_{mixer} + G_{LPF}G_{amp2}T_{LPF} + G_{amp2}T_{amp2} + Tspectrum)$$

Equation 4:

To compare with our spectrum traces we will calculate the displayed noise level/ power for each stage:

$$P_n = kT_{system}\ \Delta f$$

Equation 5: Noise power

		THEORETICAL :		**Measured:**	Trace
Stage	SYSTEM TEMP AT EACH STAGE	Noise level [W]	Noise level in dBm		
Stage 1: antenna temp	1.15E+06	1.59E-14	**-107.9988722**	**-71.44**	Trace 33
Stage 2: after 1st amp	4.21E+06	1.74E-12	**-87.58935004**	**-74.62**	Trace 34
Stage 3: after the mixer	1.68E+06	2.32E-13	**-96.34242074**	**-75.5**	Trace 35
Stage 4: after the low pass filter	1.88E+06	2.60E-13	**-95.85228002**	**-66.11**	Trace 36
Stage 5: 2nd amplifier	4.27E+08	5.90E-11	**-72.29422989**	**-79.15**	Trace 37

Discrepancies between them: Being dominated by the spec until the last stage, not going to see our noise floor, we are not the problem! :)

Distortion Level:

To find the theoretical distortion we first point out that only the last stage was acting non-linear and under normal operating conditions the 1st amplifier, mixer and low pass act linear. Moreover, the second amplifier's distortion products will be much bigger than the first and we can check the 1st amplifier is linear by looking at its intermodulation products and checking that it is smaller than the fundamental .

From the datasheet we find the OIP3 for the 1st amplifier, and use the relationship between OIP3 and IIP3 to find IIP3 -> OIP3 Gain of the amplifier = IIP3. We also did the same for the 2nd amplifier.

	power in -70	
	IIP3 (dBm)	OIP3 (dBm)
1st amplifier	-10	27.33
2nd amplifier	-9.55	14

IIP3 and OIP3 values for the 2 amplifiers in our receiver system

Using IIP3 and the OIP3 we can calculate the backoff using the following equation:

$$IIP3\ -\ Input\ Power\ =\ Backoff$$

Equation 6

We later use the backoff to find the power of the intermodulation products using the following equation:

$$P3IMO\ =\ OIP3\ -\ 3\ *\ Backoff$$

Equation 7

If power of the intermodulation products was below the noise floor then we could disregard it.

1st Amplifier										
	backoff	40	db	IIP3 -PRX	no GAIN OR LOSS for the first amplifier					
	Power in the intermodulation products	-92.67	Acting linear because it is sucha a small number, therefore there will be no spikes visible in the spectra						IMproducts	
2nd amplifier stage										
	backoff	11.17			What is the input power in the 2nd amp relative to the iip3				16.13	
	Power in the intermodulation products	-19.51	at the output	This is a small answer that means the last 3 stages are linear	Power of the intermodulation products at the output				-19.51	
					Trying to find the backoff					
	System						CORRECTED THEORETICAL:		Using measured gain to find IIP3	
theoretical	System IIP3	-38.83							-18.37	
	~~backoff~~	~~31.17~~								
	~~Power in the intermodulation products~~	~~-79.51~~								
measured	System IIP3	-22.12								

Calculations for the 1st, 2nd and system stages for distortion products.

When doing all of the calculations, we realised we were operating in the amplifiers heavily gained compressed regions, which means our gain was probably off from the datasheet value, and that is why our theoretical system IIP3 (-38.83 dBm) is so far of from the measured system IIP3 (-22.12 dBm). In order to get a more accurate theoretical value, we used the measured gain (32.37dBm), and with it found our corrected theoretical value (-18.37 dBm), which is much closer to the measured system IIP3.

```
Marker 1 linear slope is: 0.8540000000000001
Marker 2 linear slope is: 2.8080000000000003
Marker 1 linear intercept is: 30.01666666666669
Marker 2 linear intercept is: 71.11666666666673
```

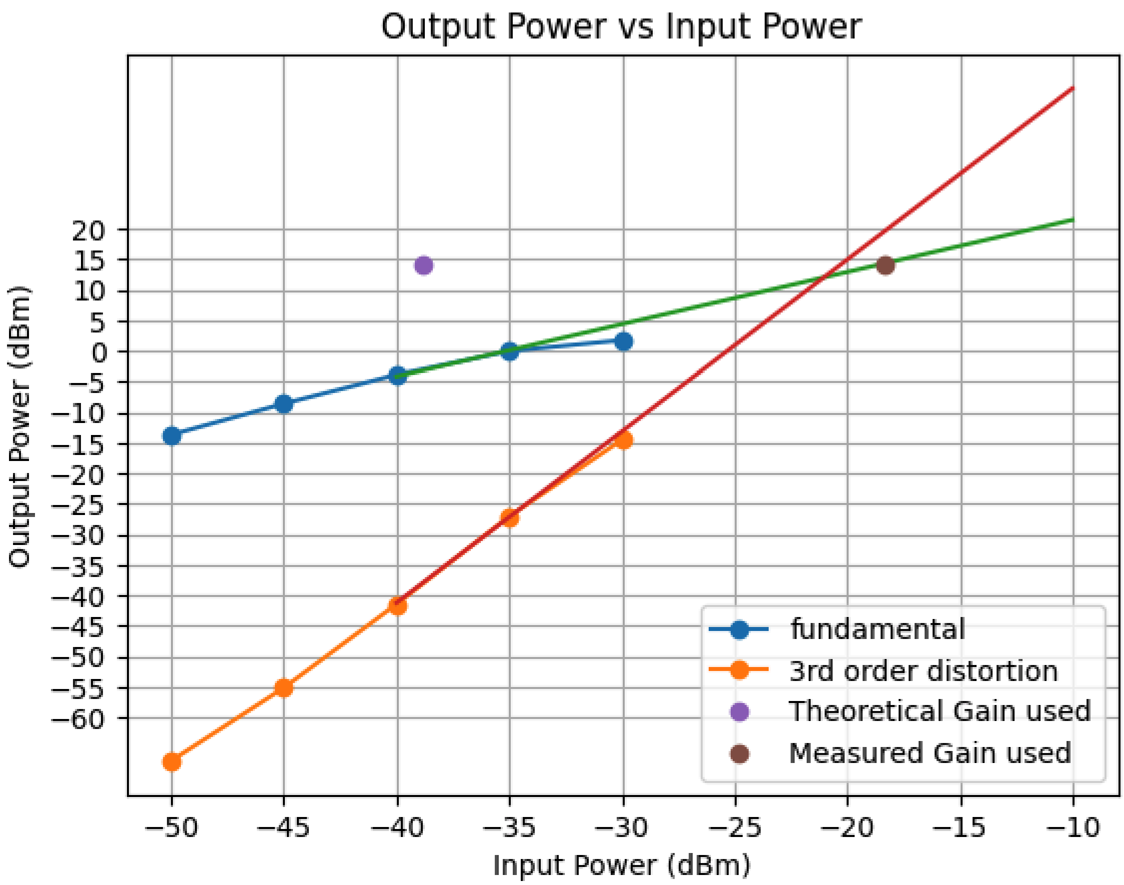

Figure 26: Graph obtained by plotting Pin vs Pout for the distortion products.

APPENDIX 1: CODE FOR THE DIGITAL FILTER

DP2.ipynb

Python

```python
import numpy as np
from scipy import signal
import matplotlib.pyplot as plt

fs = 1e9

# SDS00001.csv
# transmitter at 3 meters
# amplifier powered with 5 V
# signal generate at 2.25 GHz and 4 dBm
#
# SDS00002.csv
# now with lna ZFL-1000LN after the lpf
#
# SDS0003.csv
# down the hallway, 14.097 m
#
# SDS0004.csv
# SDS0005.csv (duplicate, oops)
# jacobs 104 outlet doorstop
#
# SDS0006.csv
# end of physics hallway, 3060 inches = 77.724 meters
# /content/SDS00002.csv
data4 = np.genfromtxt('/content/SDS00006.csv', delimiter=',',
skip_header=12)

t = data4[:,0]
v = data4[:,1]
```

```
# convert real signal to IQ
v_iq = signal.hilbert(v)

# quadrature demod: extract phase of adjacent samples
demod = np.angle(v_iq[:-1] * np.conj(v_iq[1:]))  #taking pairs of
adjacent angles and determining the phase angle between adjacent
samples
#frequency demodulation
tdemod = t[:len(demod)]

# lowpass the demodulated to clean it up
b, a = signal.butter(4, 50e3, btype="lowpass", fs=fs)
filtered = signal.filtfilt(b, a, demod)

# plt.plot(tdemod, filtered, linewidth=0.5)
# plt.show()

# # binary threshold
# threshed = filtered < filtered.mean()

# plt.plot(tdemod, threshed, linewidth=0.5)
# plt.show()
plt.plot(t,v, linewidth=0.5)
xmin, xmax = plt.xlim(-0.004, -0.0039)
plt.xlabel('Time (s)')
plt.ylabel('Amplitude')
plt.title('Oscilloscope data zoomed in')
plt.show()

# plt.plot(tdemod, demod, linewidth=0.5)
# xmin, xmax = plt.xlim(-0.004, -0.0039)
```

```python
# plt.xlabel('Time (s)')
# plt.ylabel('Amplitude')
# plt.title('Demodulated data zoomed in')
# plt.show()

plt.plot(tdemod, filtered, linewidth=0.5)
plt.xlabel('Time (s)')
plt.ylabel('Amplitude')
plt.title('Filtered Data')
plt.show()

# binary threshold
threshed = filtered < filtered.mean()

plt.plot(tdemod, threshed, linewidth=0.5)
plt.xlabel('Time (s)')
plt.ylabel('Amplitude')
plt.title('Cleaned Filtered Data')
plt.show()
```

APPENDIX 2: CODE FOR THE DIGITAL FILTER:

DP2.ipynb

Python

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

#real data
IIP_data = np.array([
   # input Fund[dBm] 3rd
   [-50, -13.65, -67.11], #trace 28
   [-45, -8.56, -55.13 ], #trace 29
   [-40, -3.85, -41.43 ], #trace 30
   [-35, -0.02, -27.05 ], #trace 31
   [-30, 1.77,  -14.51 ]  #trace 32
])

input_powerIIP = IIP_data[:, 0]
marker1 = IIP_data[:, 1]
marker2 = IIP_data[:, 2]
plt.plot(input_powerIIP, marker1, 'o-', label='Marker 1')
plt.plot(input_powerIIP, marker2, 'o-', label='Marker 2')
plt.legend()
plt.grid()
plt.xlabel('Input Power (dBm)')
plt.ylabel('Output Power (dBm)')
plt.title('Output Power vs Input Power')

#finding the slope linear lines

slopeM1 = (marker1[3] - marker1[1]) / (input_powerIIP[3] -
input_powerIIP[1])
```

```
print("Marker 1 linear slope is: " + str(slopeM1))
slopeM2 = (marker2[3] - marker2[1]) / (input_powerIIP[3] -
input_powerIIP[1])
print("Marker 2 linear slope is: " + str(slopeM2))
linearInput = [input_powerIIP[1],
input_powerIIP[2],input_powerIIP[3]]
linearOutputM1 = [marker1[1], marker1[2], marker1[3]]
linearOutputM2 = [marker2[1], marker2[2], marker2[3]]
slope_intercept1 = np.polyfit(linearInput,linearOutputM1,1)
slope_intercept2 = np.polyfit(linearInput,linearOutputM2,1)
print("Marker 1 linear intercept is: " + str(slope_intercept1[1]))
print("Marker 2 linear intercept is: " + str(slope_intercept2[1]))


x1 = np.linspace(-40, -10, 30)
y1 = slopeM1*x1 + slope_intercept1[1]
plt.plot(x1,y1)
y2 = slopeM2*x1 + slope_intercept2[1]
plt.plot(x1,y2)
plt.gca().set_yticks(np.arange(-60, 25, 5))
plt.plot(-38.83, 14, 'o')
plt.show()


#loosing a 10db gain somewhere
#35 db of gian
#theoretical gain does not much bc we are running out calculation
with our
```