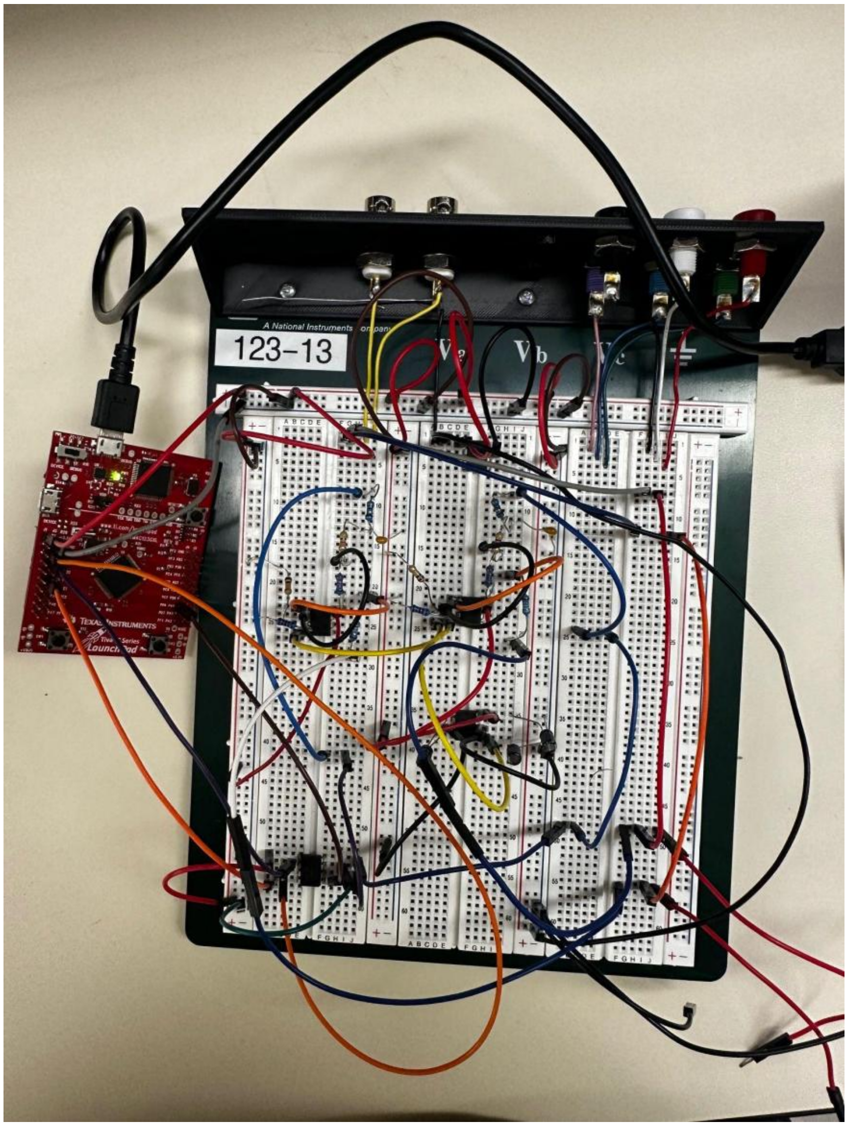
Wireless Lab Report

Lab #6

CSE 4377 – Dr. Losh

KOPIL SHARMA (1001845041)



Introduction

In this lab, we extended our work from Lab 1 to implement a QPSK transmitter that includes a preamble sequence (0xCCCCCCCCCCCCCCCC) before transmitting a QPSK-encoded message ("HELL"). The signal was transmitted using the Lab 1 circuit connected to TM4C123GH6PM which was further connected to a signal generator and received using an RTL-SDR USB receiver. We analyzed and visualized the received IQ data using MATLAB to observe the demodulated signal and identify frequency offsets.

Process Overview

Modulation Preparation:

- The original Lab 1 code was modified to include a preamble before transmitting the message.
- The QPSK message sent was "HELL".

Hardware Setup:

- We used our TM4C123GH6PM microcontroller with the circuit to generate the signals and connected the I and Q inputs to the IQ modulation input of the RF signal generator.
- The RF generator was configured to be transmitted at 1431 MHz.
- The output power was set so that the signal level observed on the spectrum analyzer was approximately –84.5 dBm.
- Signal Captures: We collected three types of IQ recordings using rtl_sdr on the Raspberry Pi:
    1. **DC signal (constant 0.5 + j0)**
    2. **1 kHz tone**
    3. **QPSK-modulated signal with preamble**

Each signal was captured for 1 second using the command:

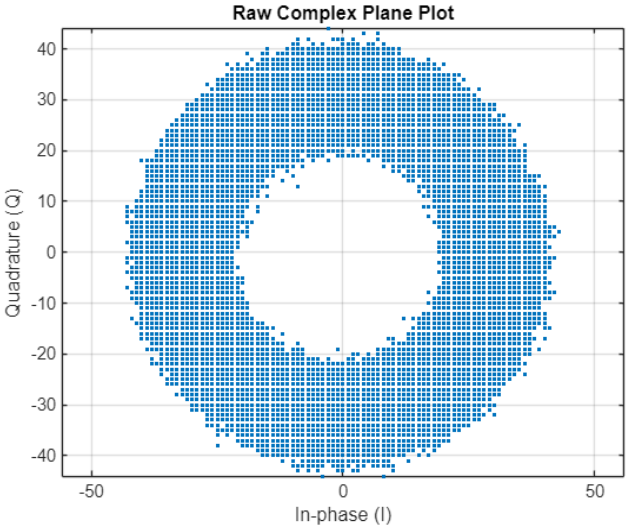
```
rtl_sdr -f 1431000000 -s 2048000 -n 2048000 AlexKopilQPSK.iq
```

Results and Analysis

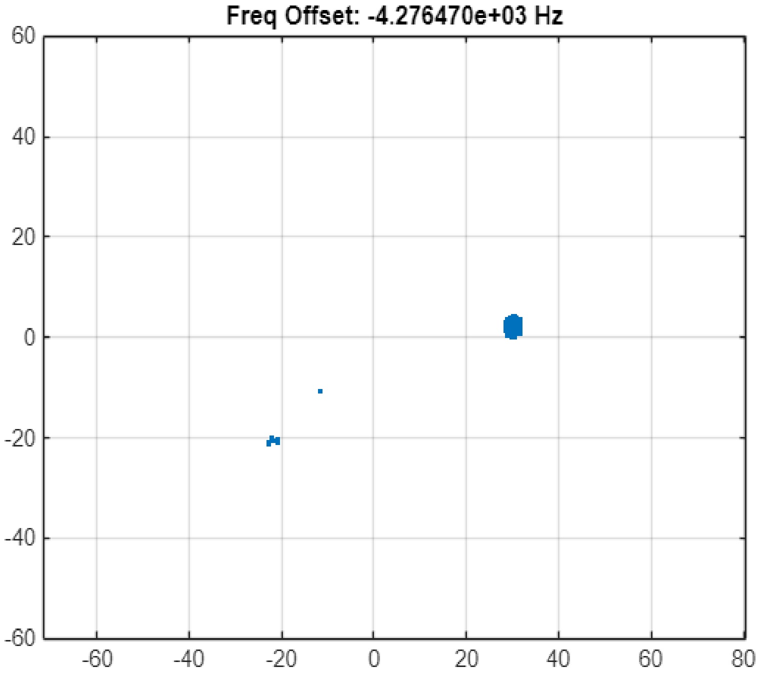
DC Signal:

Expected: A single point in the complex plane.

Observed: A rotating donut due to frequency offset between the generator and SDR.



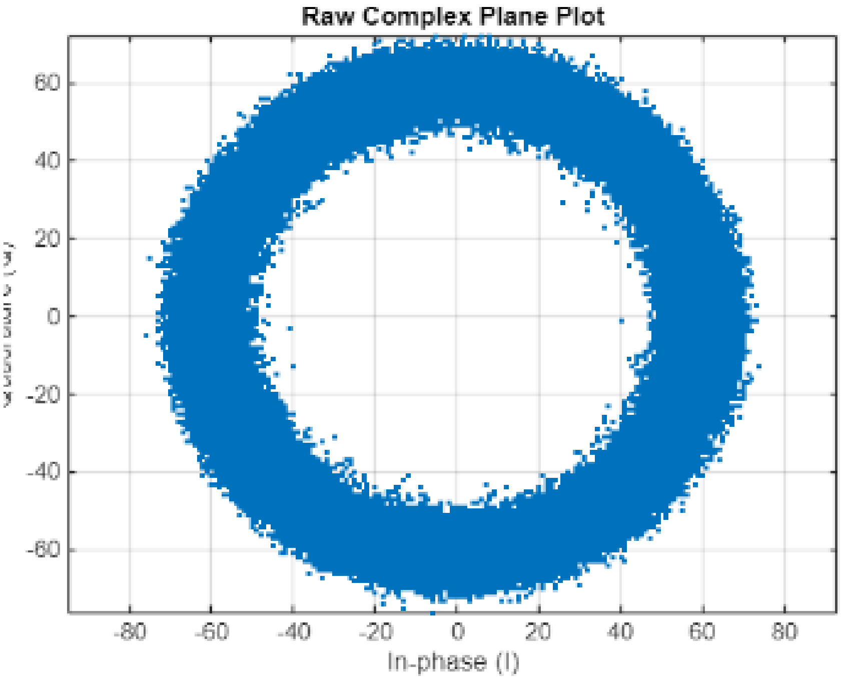
After applying rotation correction (offset: –4276.47 Hz), the plot collapsed into a tight dot.



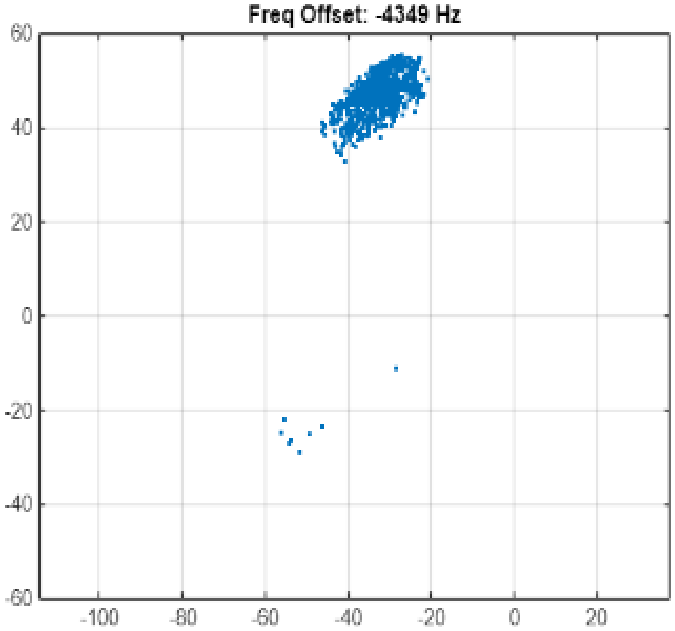
1 kHz Tone:

Expected: A circular trace in the IQ plot.

Observed: Clear circular rotation.



Frequency offset found: -4349.01 Hz. It also converged to a fuzzy dot after correction.

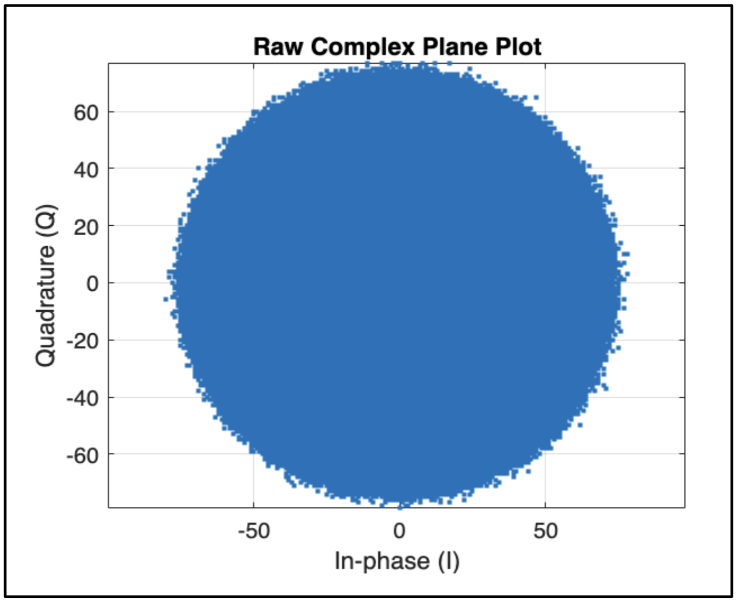


QPSK Signal:

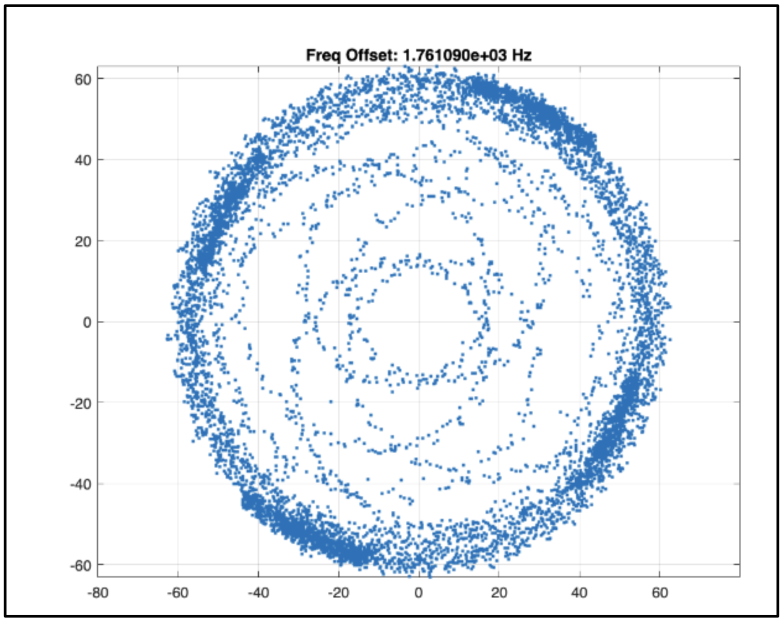
Expected: Four constellation points (QPSK) visible and stable.

Observed:

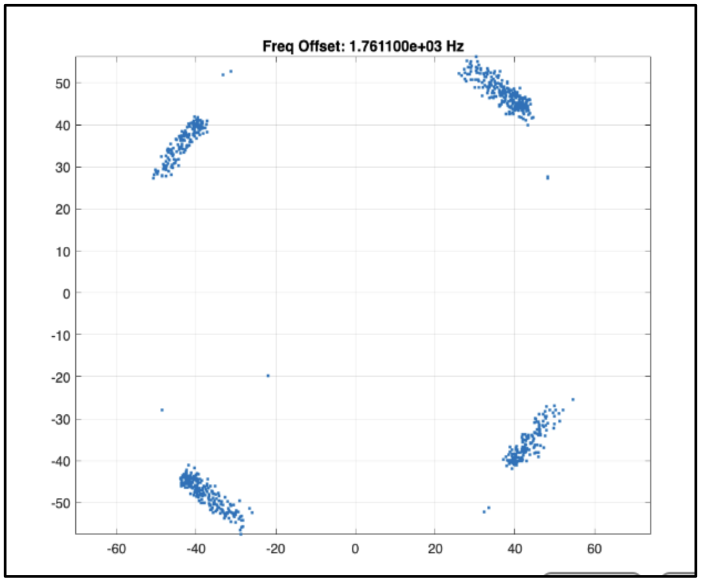
Unfiltered: A Full circle with noise all over the area.



Filtered: A circular figure with less noise in the center.



After frequency offset correction **(+1761.1 Hz),** a stable QPSK constellation appeared.



MATLAB Processing Steps

We first plotted the raw value before and after they are filtered.

- rxFilt = conv(rx, h8000);
- rxFilt = rxFilt(1:2048000/8000:end);

Then we applied the **frequency offset correction** using this part of the submitted code:

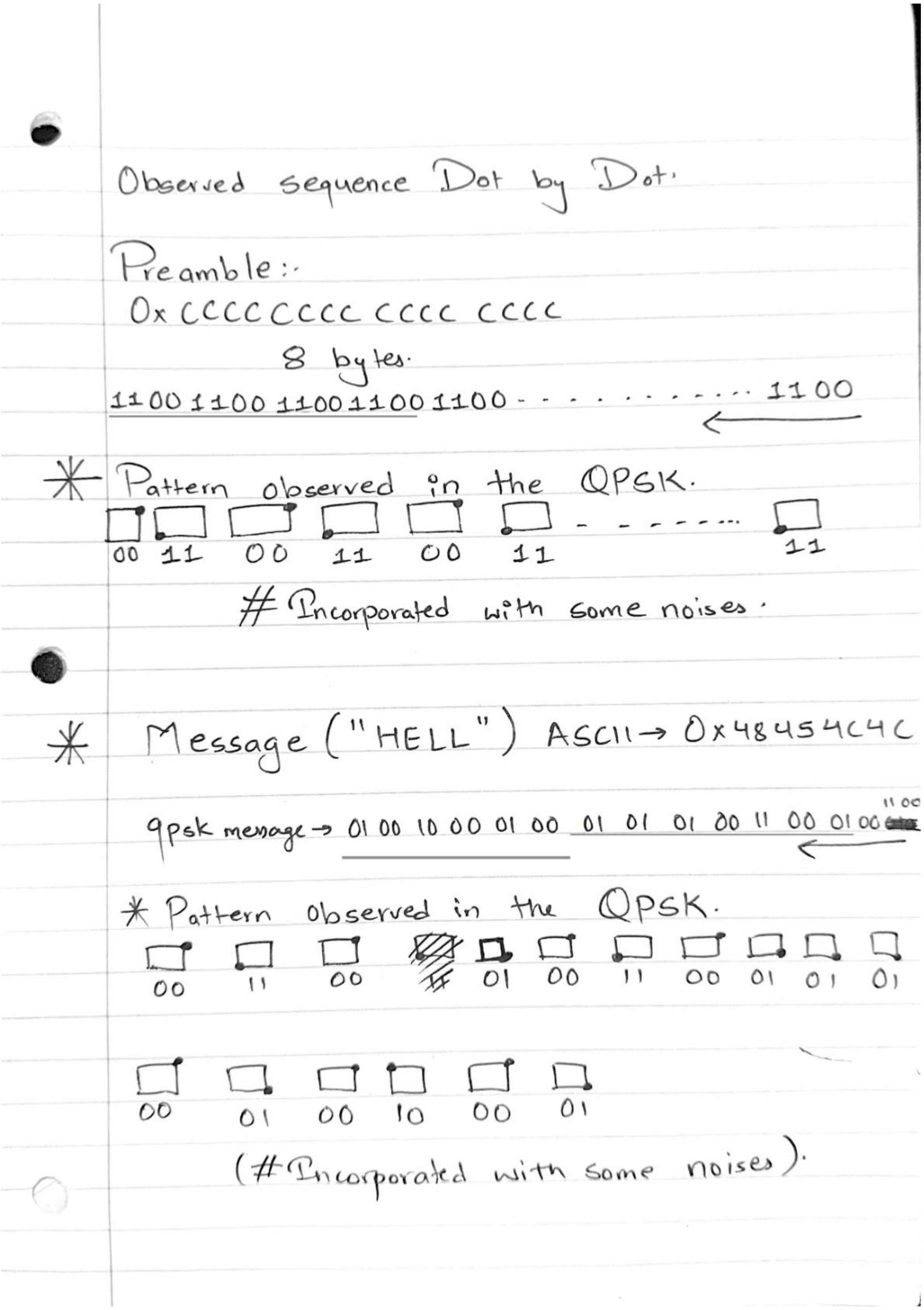
- complex(cos(2*pi*i/2048000*f), sin(2*pi*i/2048000*f))

Preamble Detection:

We had a hard time applying the correlation and finding the right preamble sequence. A preamble sequence was detected but it was at the 7969 sample and the remaining sample wasn't enough for us to detect the correct message that should have been received. It was probably because of **noise** or faulty **raspberry-pi.**

Instead, we decided to go **dot by dot** from the very beginning and we tried to find out our preamble sequence which is basically 0XCCC....(A diagonal fluctuation of the dot in the QPSK plane). We plotted the points dot by dot, in a for loop from the beginning, with a pause of 0.3 sec until we found our exact similar looking pattern.

When we went through the dot by dot process we were able to see a **similar pattern** of our preamble but it was incorporated by the noises and we couldn't correctly figure out the message but we were able to observe similar QPSK pattern that was in match to our preamble and message: Here is a quick description in the picture below of how it looked in the matlab with noises:



Observed sequence Dot by Dot.

Preamble:·

0x CCCCCCCC CCCC CCCC

8 bytes.

1100 1100 1100 1100 1100 - - - - - - - - - ... 1100 ←

* Pattern observed in the QPSK.

00 11 00 11 00 11 - - - - - - ... 11

Incorporated with some noises.

* Message ("HELL") ASCII → 0x48454C4C

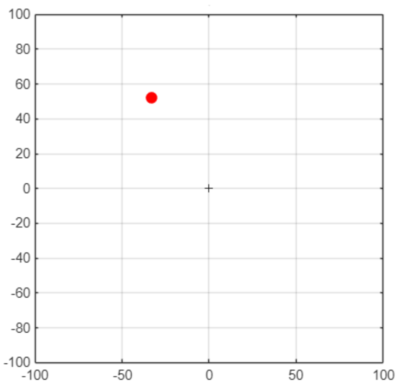
qpsk message → 01 00 10 00 01 00 01 01 01 00 11 00 01 00 11 00 ←

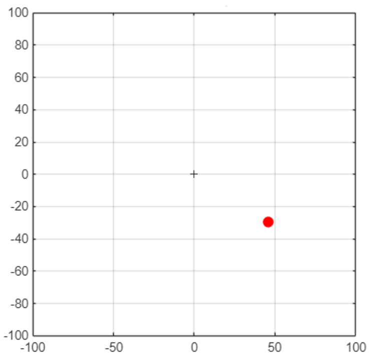
* Pattern observed in the QPSK.

00 11 00 # 01 00 11 00 01 01 01

00 01 00 10 00 01

(# Incorporated with some noises).





Observation: We can see the preamble going diagonally when we started looking dot by dot.

BER: For a random byte of preamble, we received these bits:

Theoretical: 11001100

Actual: 11011101

BER = 2/8

Since two bits received were wrong the bit error rate was One Quarter **(1/4)** for this byte sample which should be similar to all the samples.

Conclusion

In this lab, we successfully transmitted and received QPSK data using a preamble to aid in synchronization. We captured IQ data and applied filtering, frequency correction, and manual inspection of QPSK to analyze the results in MATLAB.

We observed:

- Clear rotation in the DC and tone signals, corrected via frequency offset.
- Stable QPSK constellation after correcting for a +1761.1 Hz offset.
- Dot by dot detection of the preamble using manual visuals.
- Calculated BER from the random byte sample.
- This lab helped demonstrate the importance of:
    1. Proper timing and offset correction,
    2. Using preambles for synchronization,
    3. And understanding how modulation translates into observable IQ data.