# FAUXY Lady

## Task

* Category: Can’t Stop the Signal, Mal
* Points: 165
* Solves: 19

Description:

A university needs your help collecting their cubesat’s telemetry. We’ve captured a .wav recording of the satellite signal and the university has published the telemetry definition. The recorded signal has the following characteristics:

BPSK modulated Differentially encoded 44.1k samples per second

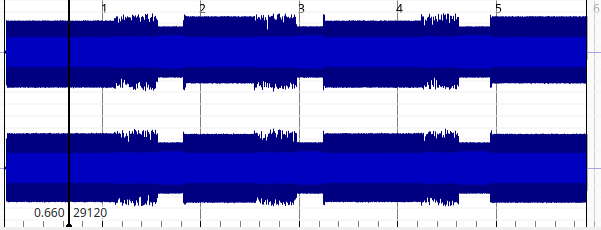
Can you reconstruct the telemetry packet?

Attachements:

* tlm\_def.pdf, a description of the packet format
* signal.wav, a wav file containing the signal. 2 Channels, one I one Q.

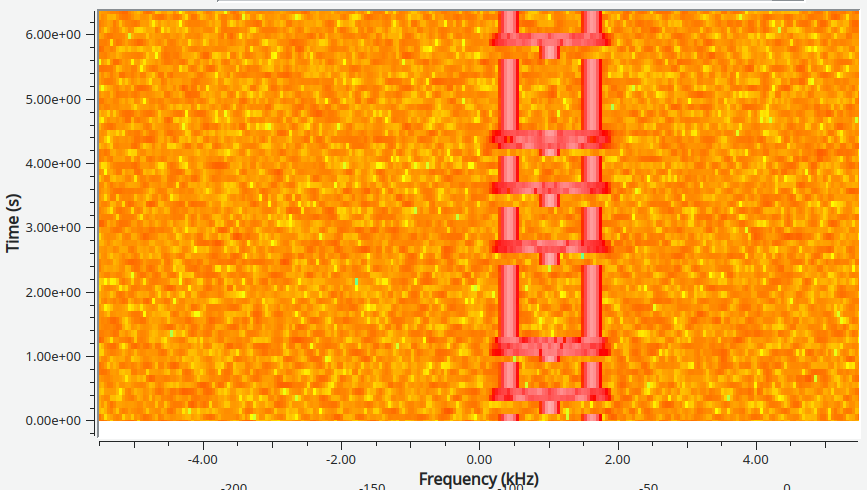
## Solution

First, we opened the wav file in sonic-visualizer to get a look at the waveform:

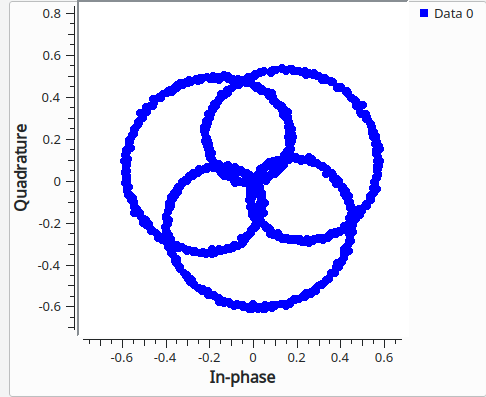


As you can see, there seem to be three packets.

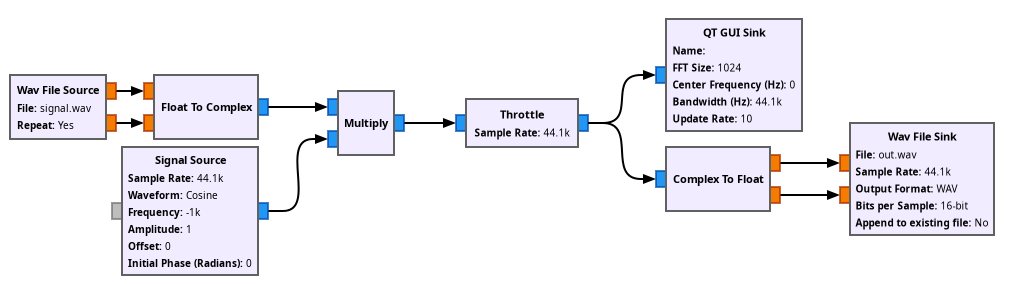
Next, we visualized the signal in a GNURadio Waterfall and Constellation Sink:



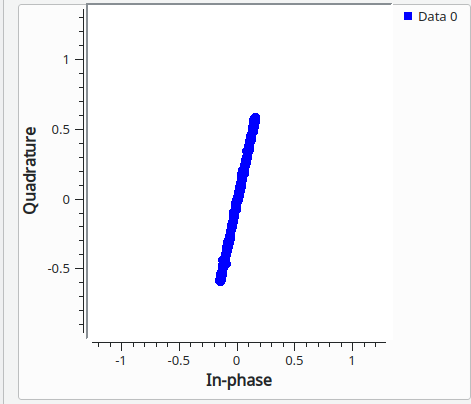
The signal is NOT perfectly in the baseband yet, but shifted to exactly +1000Hz. This makes our constellation display “spin”:



We undo the shift by multiplying with the appropriate complex cosine:



This corrects the constellation:

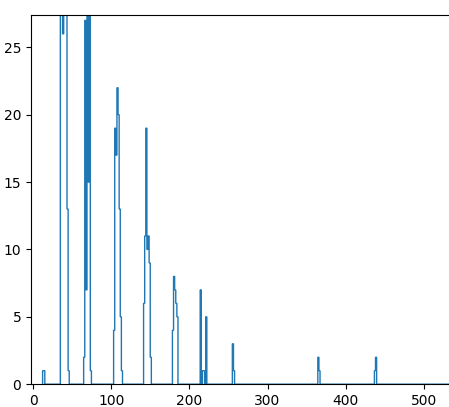


Working with GNURadio and digital signal is always as bit annoying, and we found it easier to do decode in python. We therefore write the frequency-shifted signal back to a file (out.wav).

From the description of the telemetry packet, we know that it will start with 00011010110011111111110000011101 and end on 01111110.

We know it is BPSK Differential Encoded. We have two constellation points. By educated guess we choose to use the sign of the I component of the signal as symbol, which turns out to work good enough.

As we don’t know the baudrate or samples per symbol, we run-length-encode the resulting symbol-stream and plot a histogram of the lengths of the buckets:



We find very distinct peaks with lots of empty in between. They corrospond to “one same symbol in a row”, “two same symbols in a row” etc. We take a rough guess at ~35 samples per symbol. Now, optimally you’d want to have some kind of resynchronization mechanism, but we are lazy and this is a CTF afterall. Sooo we just divide the run-length of each symbol by 35 and continue.

The differential encoding part is simple. If we observe two same symbols in a row we emit a 0, otherwise a 1. We tried both ways and this one was correct.

This now gives us a bit-stream we can compare to the expected packet format. This guess of samples-per-symbol turned out to be slightly wrong, but the known bits of packet start and end gave us enough info to simply try some values. We ended up with 36.75 samples per symbol, which worked great.

Each of the three telemetry packets held a part of our flag. Putting it together, we get:

Flag: flag{juliet422165hotel4:GEHcyMz0-lG5ECral59sCkgHSIsw9wapbs-Dh3wGAN3V9pF2vsRYtXyJctNBfoDD7CFVmeV-xBIusRqEyPMud38}