**Lab 02 - Decoding Signals**

# Part I: Demodulate Audio

analyze and demodulate provided FM RF data file

## Instructions

1. Setup a system with GNU Radio and GNU Radio Companion or use the VM provided in D2L
   1. GNU Radio is typically happiest on Linux
   2. If you fell in love with BSD in 840, please consider breaking up
2. On your canvas, load the data file and demodulate it using the demodulate module.
   1. You’re welcome to investigate demodulating it manually as well
   2. Many tutorials exist online for demodulating the signal of a Software Defined Radio. You can use those, just swap the file source for the RTL-SDR source (you may need to add a throttle with the signal rate too)

## Findings and Analysis

1. Who is the sweepstakes open to which age groups?

*The sweepstakes is open to South Dakota teens, age 14-19.*

1. There’s a commercial with a big-wig CEO, who is it or what is he the CEO of?

*A diagram of a computer

Description automatically generatedThe commercial features Jay Farner, CEO of Rocket Mortgage*

Demodulate FM (WBFM Receive)

in order to reduce the computational load and focus processing on the relevant information for audio playback, I selected an Audio Decimation rate of 40. Since the quadrature rate is 2M, I chose to decimate it by a factor of 40 to produce an audio sample rate of 50k.

Resample (Rational Resampler)

Next, I selected the "Rational Resampler" to match the sample rate of the demodulated signal to the desired audio sample rate for playback (46kHz). I solved for the Resampling Factor by solving for the ratio of the output rate (L) to the input rate (M). At this point, our sample rate is 50kH because we just demodulated FM, and we know that 48kH is the desired output, therefore, this can be solved mathematically as such:

|  |  |  |
| --- | --- | --- |
| interpolation factor output rate |  |  |
| decimation factor input rate |  |  |
| resampling factor |  |  |

Audio Sink

*A screenshot of a computer

Description automatically generated*It was then hooked up to an Audio sink at our target 48kHz sample rate.

# Part II: Demodulate Digital Data

discover, analyze, and decode transmissions

## Instructions

1. Use SDR# to connect to the radio from the first lab (sdr://138.247.12.20:5555)
2. Tune into ISM band where this mystery device is transmitting
3. Identify frequency of the device using the provided FCCID (M3N-A2C31243300)
4. Capture transmission from this mystery device
5. Open transmissions in Audacity (or another appropriate tool for waveform analysis)
6. Identify waveform
7. Decode digital data and convert to hexadecimal format

## Findings and Analysis

Look up device using Federal Communications Commission Identification (FCC ID)

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Tune to Device

When searching around 902-903mHz, I located the signal at 90.1.9375mHz

A screen shot of a graph

Description automatically generated

Configure Settings and Record

After setting bandwidth to 30k and modulation to AM, I was able to effectively capture and record the signal

A screenshot of a computer

Description automatically generated

Open Recording in Audacity

A screenshot of a computer

Description automatically generated

As shown from this image, the signal is repeating. I isolated the target signal and zoomed in to get a better view of the amplitude modulation wavelength.

A screenshot of a computer

Description automatically generated

Analyze

A screenshot of a computer

Description automatically generated

Looking back at the zoomed in image, it can be seen that there are a series of consistent repeating patterns (shown in red), but then it varies (shown in green). This is because the red section is the Preamble, which the transmitter uses to wake up a receiver. A Preamble has a distinctive pattern or structure that the receiver can quickly recognize . The section in green are the Data frames which contain the target information.

Decode

This is a common modulation type, where the peaks represent 1’s and the valleys represent 0’s

To decode this capture, I uploaded the file into Universal Radio Hacker (URH).

A diagram of data frames

Description automatically generated

This capture contains the following binary sequence:

|  |  |
| --- | --- |
| Preamble | 1010 1010 1010 1010 1010 1010 1010 1010 |
| Header/Control Flags | 0000 |
| Data Frames | 1100 0100 1110 1101 1110 1100 0000 1101 1110 |

Convert to Hex

I then copied these values into the Binary to Hex editor at [Tools a Day](https://toolsaday.com/number-tools/binary-to-hex) to find the hexadecimal equivalent.

A number and text on a white background

Description automatically generated