Beyond 5G, Spring 2024
HD Radio Side Band Isolation and Symbol Synchronization Project
Jack Willard, Pryce Tharpe, Troy Madden

Meeting 2/20/24:

Meeting Progress:

- **HD Radio Side Band Isolation**
  - Pryce made a slideshow for presentation during 2/22 meeting with Professor
  - Further brainstorming was completed by project members to plan out project, and it was determined
  - Troy found some resources to help with the project including, https://kb.ettus.com/Building_and_Installing_the_USRP_Open-Source_Toolchain_(UHD_and_GNU_Radio)_on_Linux -> To download gnu and uhd radio on a linux machine
  - In the documentation above it is determined v3.7.13.4 of would be the appropriate direction to move in to install gnu radio software
  - Introductory usrp slides supplied by Pryce: https://kb.ettus.com/Building_and_Installing_the_USRP_Open_Source_Toolchain_(UHD_and_GNU_Radio)_on_Windows
  - https://kb.ettus.com/images/4/47/Workshop_GnuRadio_Slides_20190507.pdf
- **Symbol and Frame Recovery**
  - This will be worked on this project following the HD radio project, the specific people assigned to each is still up in the air
  - Initial plans for the project include literature review
  - Search keywords **matched filter/correlator** as well as **preamble** and **auto correlation function** (Start here)
- Jack updated the notebook

Road Bumps:

- The team does not have an ethernet to usb adapt, it is believed having one would be advantageous, so we will ask Prof. Wang during the meeting Thursday 2/22
- It was determined during download specific versions of UHD may be better than others, per this web page https://files.ettus.com/manual/page_semver.html - found by Jack. This pushes back the download of UHD software

Objectives Before Next Meeting:

- Everyone should download Ubuntu and set up gnu radio, per https://kb.ettus.com/Building_and_Installing_the_USRP_Open-Source_Toolchain_(UHD_and_GNU_Radio)_on_Linux
- Literature should be reviewed on the topic to develop a background set of knowledge and plan of attack for the problem

Following the meeting with professor Wang on Thursday 2/22/24 it was determined the team would need to split work and research separate but related topics. So the team broke down the HD radio sideband isolation into the following research topics with the assigned lead next to them.

Literature Review:

Jack Willard - Device Streaming

- UHD Development Manual - https://files.ettus.com/manual/page_uhd.html
- Hardware Specs - chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.ettus.com/wp-content/uploads/2019/01/07495_Ettus_N200-210_DS_Flyer_HR_1.pdf
- Hardware Driver and USRP manual - https://files.ettus.com/manual/page_usrp2.html
- UHD information - https://kb.ettus.com/UHD
    - Will GNU radio be needed? - Maybe, due to signal processing tools

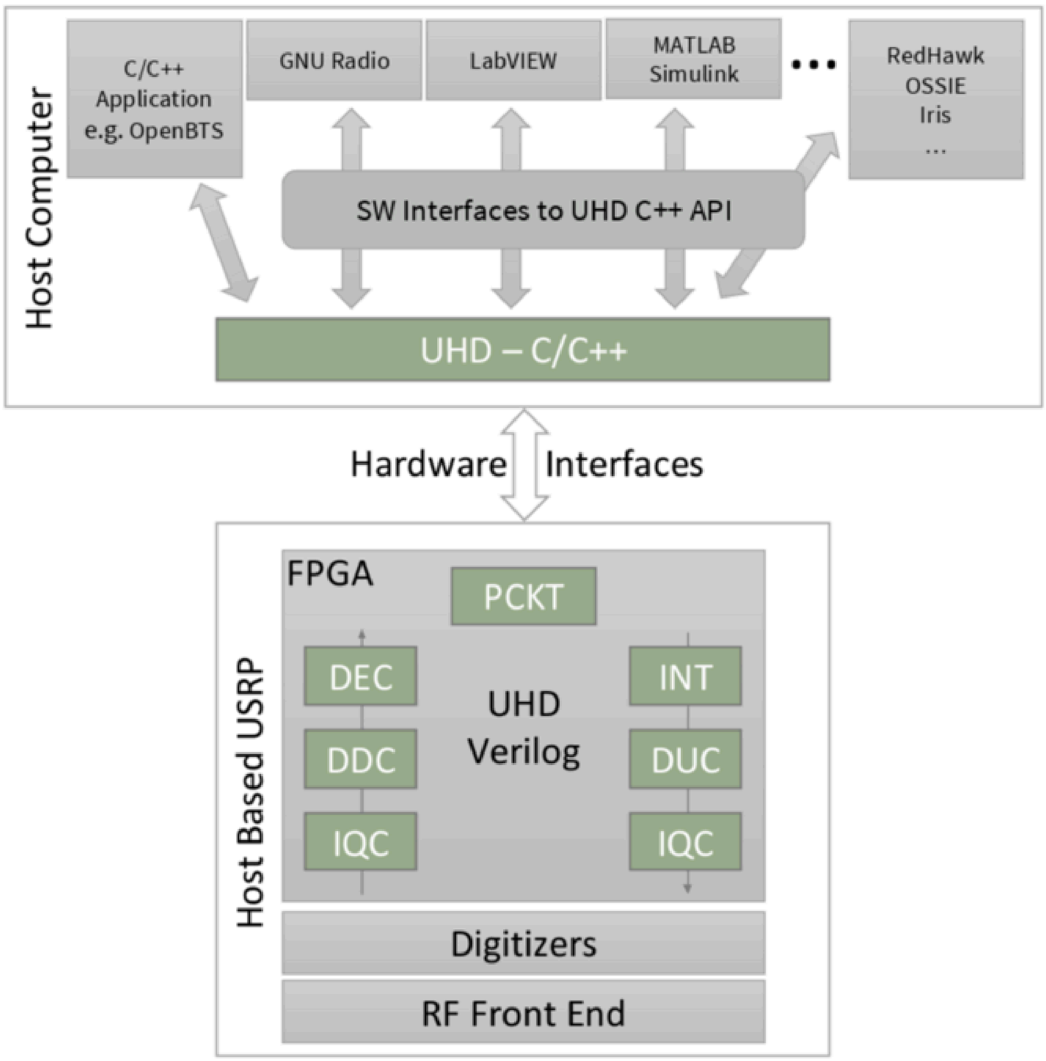


Figure 2. UHD components

    - Working with Network series hardware (N210)
    - UHD is C/C++ API
    - Most UHD code is open source
    - Supports Python 3
    - UHD controls transfer of I and Q samples

- UHD allows for digital up and down conversion to help with have sub Hz RF frequency step size, fast tuning inside bandwidth, and mitigating DC problem that exists
- UHD provides decimation and interpolation blocks to perform sample rate translations for higher bandwidths, if the computers which do not have large enough bandwidth
- IQ corrections -> the "HW Impairments" block can be utilized to simulate disruptions in the in-phase and quadrature signals

uhd_cal_rx_iq_balance: - mimimizes RX IQ imbalance vs. LO frequency

uhd_cal_tx_dc_offset: - mimimizes TX DC offset vs. LO frequency

uhd_cal_tx_iq_balance: - mimimizes TX IQ imbalance vs. LO frequency

- Self-Calibration of USRP - https://files.ettus.com/manual/page_calibration.html
  - The daughter board we are using naturally puts calibration data into csv file format, and running a command can put it into binary format
  - The DC offset can be tweaked and accounted for if known

UHD software comes with the following calibration utilities:

- **uhd_cal_rx_iq_balance: -** minimizes RX IQ imbalance vs. LO frequency
- **uhd_cal_tx_dc_offset: -** minimizes TX DC offset vs. LO frequency
- **uhd_cal_tx_iq_balance: -** minimizes TX IQ imbalance vs. LO frequency

  - Single point IQ imbalance can be accounted for in the equation by tweaking values of A and B listed on this website below

$$\begin{pmatrix} I' \\ Q' \end{pmatrix} = \begin{pmatrix} A/64+1 & 0 \\ B/64 & 1 \end{pmatrix} \begin{pmatrix} I \\ Q \end{pmatrix}$$

- Tuning of USRP - https://files.ettus.com/manual/page_general.html#general_tuning_process
  - IF -> intermediate frequency, LO -> Local Oscillator
  - When integer-N tuning requested, the UHD will pick up the LO frequency closest to the desired RF frequency and tune the rest digitally
  - RF front-end needs time to settle into a usable state as LO needs time to lock
  - "It is important to understand that strictly-integer **decimation** and **interpolation** are used within USRP hardware to meet the requested sample-rate requirements of the application at hand. That means that the desired sample rate must meet the requirement that master-clock-rate/desired-sample-rate be an integer ratio. Further, it is strongly desirable for that ratio to be **even**."
  - The N210 will likely have a fixed master clock leaving fewer effective sample rates available on the hardware
  - A "D" prints to stdout when the host can't consume data fast enough

- Underrun occurs when not enough data is produced, “U” is printed
- Information on threading

- GNU Radio on Wikipedia - https://en.wikipedia.org/wiki/GNU_Radio
- Other notes, I downloaded UHD v4.5.0.0 and Ubuntu version 18.04.6, downloaded gnu v3.7.13.4, however only 95% of my cases passed for gnu. Install was unsuccessful for gnu v3.7.13.4
- How do we access band?
- What does software interface look like?

Pryce Tharpe - Converters

- General stuff on converters - https://files.ettus.com/manual/page_converters.html
- Programming bank for converters - https://files.ettus.com/manual/classuhd_1_1convert_1_1converter.html
- “Most FPGAs use integer data types, the most common being complex 16-bit integers (16-bit for I and Q, respectively)”... Ours is this 16-bit integer type

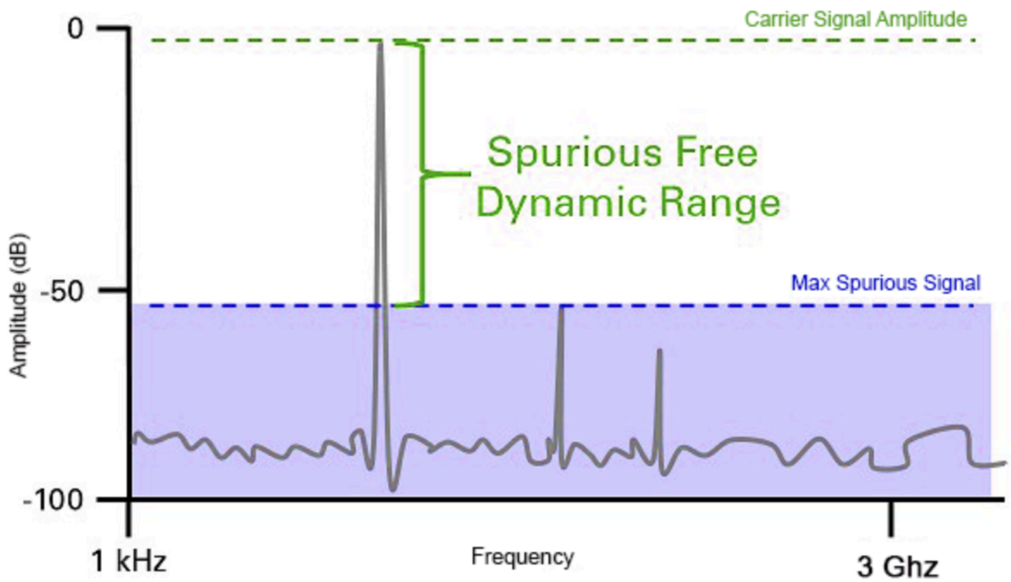
◆ conv()

```
UHD_INLINE void uhd::convert::converter::conv ( const input_type &  in,
                                                const output_type & out,
                                                const size_t        num
                                              )
```

The public conversion method to convert inputs -> outputs.

CONVERSION PERFORMANCE AND CLOCKS		
ADC Sample Rate	100	MS/s
ADC Resolution	14	bits
ADC Wideband SFDR	88	dBc
DAC Sample Rate	400	MS/s
DAC Resolution	16	bits
DAC Wideband SFDR	80	dBc
Host Sample Rate (8b/16b)	50/25	MS/s
Frequency Accuracy	2.5	ppm
w/ GPSDO Reference	0.01	ppm

ADC - Analog to digital converter
DAC - Digital to analog converter
SFDR is a Spurious-free dynamic range - decibels relative to carrier - in simple terms, it measures the purity of a signal.
GPSDO is a GPS disciplined oscillator - a gps clock - steers oscillator output to GPS reference with a ppm accuracy of .01(without this reference, the internal clock is accurate to 2.5 ppm)



This is just a stack exchange from a different person using the n210 to receive FM radio - https://ham.stackexchange.com/questions/9473/viewing-the-broadcast-fm-spectrum-using-a-usrp-n210 * There are a few of these to browse to find fixes for issues.
Also a 10-minute guide to programming a USRP for FM radio - https://youtu.be/KWeY2yqwVA0?si=N97LeBYtBJGur8ti * Plenty of fixes for errors in comments of video.

GPT SUM:
“USRP (Universal Software Radio Peripheral) is a flexible and widely used platform for software-defined radio (SDR) applications. Converters in the context of USRP generally refer to the analog-to-digital converter (ADC) and digital-to-analog converter (DAC) components within the device. These converters are crucial for interfacing between the analog radio frequency (RF) signals and the digital processing performed by the software-defined radio system.

Here's a brief explanation of each:

Analog-to-Digital Converter (ADC):

The ADC is responsible for converting the analog RF signals received by the USRP's RF front-end into digital samples that can be processed by the digital signal processing chain.
The ADC's characteristics such as resolution (bits), sampling rate, and dynamic range are critical for capturing the fidelity of the received signals accurately.
Higher resolution ADCs can capture finer details of the received signals, while higher sampling rates allow capturing wider bandwidths of RF spectrum.
Digital-to-Analog Converter (DAC):

The DAC converts digital signals from the processing chain into analog signals for transmission through the USRP's RF front-end.
Similar to the ADC, the DAC's resolution, sampling rate, and dynamic range are essential for accurately reconstructing the analog signals from digital samples.
The quality of the DAC affects the fidelity and purity of the transmitted signals.
In USRP devices, these converters are integrated into the hardware, and their specifications vary depending on the specific model and intended use case. Users typically choose USRP models based on factors such as ADC/DAC performance, frequency range, bandwidth, and available interfaces.

Advanced users may sometimes need to consider additional factors like linearity, spurious-free dynamic range (SFDR), phase noise, and other parameters depending on the specific application requirements, such as wireless communications, radar, spectrum monitoring, or research purposes."

Troy Madden - Receiving FM radio, GNU flowgraphs, and preliminary API research

Flowgraphs: "The basic data structure in GNU Radio is the flowgraph, which represents the connections of the blocks through which a continuous stream of samples flows. The concept of a flowgraph is an acyclic directional graph with one or more source blocks (to insert samples into the flowgraph), one or more sink blocks (to terminate or export samples from the flowgraph), and any signal processing blocks in between.

A program must at least create a GNU Radio 'top_block', which represents the top-most structure of the flowgraph. The top blocks provide the overall control and hold methods such as 'start,' 'stop,' and 'wait'."

https://wiki.gnuradio.org/index.php/Handling_Flowgraphs

A flowgraph lets you arrange signal processing blocks in a logical sequence to receive, demodulate, and process the radio signals.

```python
from gnuradio import gr, blocks, filter, analog

class my_topblock(gr.top_block):
    def __init__(self):
        gr.top_block.__init__(self)

        amp = 1
        taps = filter.firdes.low_pass(1, 1, 0.1, 0.01)

        self.src = analog.noise_source_c(analog.GR_GAUSSIAN, amp)
        self.flt = filter.fir_filter_ccf(1, taps)
        self.snk = blocks.null_sink(gr.sizeof_gr_complex)

        self.connect(self.src, self.flt, self.snk)

if __name__ == "__main__":
    tb = my_topblock()
    tb.start()
    tb.wait()
```

Example of flowgraph used with a low pass filter to filter out noise

To set the maximum number of output items, we pass a value into the 'start' or 'run' method of the gr_top_block:

```
tb.start(1000)
tb.wait()
```

or

```
tb.run(1000)
```

For receiving am/fm signals, here are basic instructions from chatGPT:

1. Install GNU Radio and UHD: Make sure you have GNU Radio and the UHD (USRP Hardware Driver) installed on your system. You can install them using package managers like apt or by building from source.

2. Set up the Flowgraph: Open GNU Radio Companion (GRC), the graphical interface for creating GNU Radio flowgraphs.

3. Add USRP Source: Drag and drop a "USRP Source" block from the "Sources" category onto the canvas. This block allows you to receive samples from the USRP.

4. Configure USRP Source: Double-click on the USRP Source block to open its properties. Configure parameters such as the sampling rate, center frequency, and gain. For A. For FM, set it to the FM station frequency.
5. M, the center frequency should be set to the frequency of the AM station you want to receive
6. Demodulation: Add appropriate demodulation blocks after the USRP Source. For AM, use an "AM Demod" block from the "Analog" category. For FM, use an "FM Demod" block.

7. Audio Output: Add an "Audio Sink" block from the "Audio" category to hear the demodulated audio. Connect the output of the demodulation block to the input of the Audio Sink.

8. Connect Blocks: Connect the output of the USRP Source block to the input of the demodulation block.

9. Save and Run: Save your flowgraph and click the "Execute" button in GNU Radio Companion to run the flowgraph.

10. Tune In: Tune in to the desired AM or FM station using the center frequency you set in the USRP Source block.

11. Adjust Parameters: You may need to adjust parameters such as the gain and filter settings to optimize reception quality.

12. Listen: Once everything is set up correctly, you should be able to listen to the audio from the AM/FM radio station.

After this signal is received, we can work on using band pass filter to isolate the HD radio portion.

API: https://www.gnuradio.org/doc/doxygen/page_filter.html

For demodulation, “HD Radio employs various modulation schemes such as COFDM (for FM) or OFDM (for AM) for the digital sideband. You would need to use demodulation blocks appropriate for these modulation schemes to extract the digital signal.”

API: https://www.gnuradio.org/doc/doxygen/page_digital.html

After demoding the digital signal, the signal needs to be synchronized to the symbol timing and carrier frequency.

Once the signal is properly found, the signal can be decoded, but the encoding is often proprietary, so it may not be that possible.

(Not sure how this is possible without knowing the encoding methods)

Beyond 5G, Spring 2024
HD Radio Side Band Isolation and Symbol Synchronization Project
Jack Willard, Pryce Tharpe, Troy Madden

Meeting 2/28/24:

Meeting Progress:

Progress was discussed in this meeting. GPSDO was defined as a GPS disciplined oscillator. This allows for better frequency accuracy.

Then specifics of the UHD were discussed while GNU wasn't.

Road Blocks:

- GNU radio not downloading correctly delaying access to the USRP
- USRP not detected

Objectives before Next meeting:

- Get USRP operational and begin plan of attack for isolation of waves
- Research more about the preamble and for symbol and frame recovery.

The team split up to work on separate topic research again.

Jack Willard - USRP and Preamble Progress

USRP progress

- Began with "uhd_find_devices" statement, however, no devices were found, so what was the problem?
    - Found, https://stackoverflow.com/questions/33304828/when-trying-to-use-my-usrp-in-gnu-radio-i-get-a-no-devices-found-for, and followed process
    - UHD version is acceptable
    - USB-c 1Gbit capable per network information in Ubuntu
    - Changed computer IP network (in ubuntu, to 192.168.10.5)
    - All the above didn't work, so what it the IP address that was incorrect?
        - Downloaded wireshark -> https://www.geeksforgeeks.org/how-to-install-and-use-wireshark-on-ubuntu-linux/
        - Ran the software and found no IP address for the USRP
        - So, how to find the IP address?
        - Followed the following procedure but in the network window https://www.wikihow.com/Assign-an-IP-Address-on-a-Linux-Computer
        - Then followed this process for the IP https://files.ettus.com/manual/page_usrp2.html
- Set IP of USRP to 198.162.10.3 per https://files.ettus.com/manual/page_usrp2.html however ran into Destination Host Unreachable Error
    - Found the following link https://www.quora.com/Why-is-Network-is-unreachable-when-attempting-to-ping-Google-but-internal-addresses-work-Linux-networking-DNS-routing

Beyond 5G, Spring 2024
HD Radio Side Band Isolation and Symbol Synchronization Project
Jack Willard, Pryce Tharpe, Troy Madden

Preamble Progress

Searched keywords **matched filter/correlator** as well as **preamble** and **auto correlation function** (Start here)

- Matched filter/correlator
    - https://cpb-us-w2.wpmucdn.com/sites.gatech.edu/dist/e/466/files/2016/11/sp14_notes3084_10_correlation.pdf

$$x(t) = s_k(t) + n(t),$$

        - Wave form
        - Squared Error Metrics
            - This process

$$\int_{-\infty}^{\infty} |x(t) - s_k(t)|^2 dt \quad = \quad \int_{-\infty}^{\infty} [x(t) - s_k(t)][x(t) - s_k(t)]^* dt$$

            - Find k that minimizes:
            - $$2\Re e \left\{ - \int_{-\infty}^{\infty} x(t) s_k^*(t) \right\},$$
            - Or maximizes:
            - $$\Re e \left\{ \int_{-\infty}^{\infty} x(t) s_k^*(t) \right\} dt.$$
            - Assumes noise distribution is gaussian
            - Referred to as template matching because there are certain assumed signal templates (known possible wave forms), and the one with the least error is matched
            - Does star mean conjugated? Why?
        - Matched filter implementation
            - This accounts for possible time shifts (convolution) Particularly useful when the time shift is unkown
            - $$y(t) = \int_{-\infty}^{\infty} x(\tau) h_k(t - \tau) d\tau.$$
        - Delay Estimation
            - The match filter implementation is beneficial when the time delay is unknown.

$$\begin{aligned} y(t) &= \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau = \int_{-\infty}^{\infty} s(\tau-\Delta)h(t-\tau)d\tau \\ &= \int_{-\infty}^{\infty} s(\tau-\Delta)s^*(-(t-\tau))d\tau = \int_{-\infty}^{\infty} s(\tau-\Delta)s^*(\tau-t)d\tau. \end{aligned}$$

time to introduce the Schwarz inequality, which, in this context, says

$$\left|\int_{-\infty}^{\infty} f(t)g^*(t)dt\right|^2 \leq \left[\int_{-\infty}^{\infty} |f(t)|^2 dt\right]\left[\int_{-\infty}^{\infty} |g(t)|^2 dt\right]$$

ıd only if $f(t) = \alpha g(t)$, for some α. If we correspond $f(t)$ with $s(\tau$ -
at

$$\left|\int_{-\infty}^{\infty} s(\tau-\Delta)s^*(\tau-t)d\tau\right|^2$$

- This value is largest when delta = t
- If there is no time delay it is the autocorrelation of the signal
    - The resolution of the system is indicated by the broadness of the peak of the signal
    - Accounts for noisy conditions
    - “If we can find a waveform with a lengthy time extent, but whose autocorrelation has a narrow mainlobe, match filtering essentially compresses the energy over that long time extent into a narrow pulse in the matched filter output, so we can enjoy both good resolution and good signal-to-noise ratio.”
- Causal concerns?

- https://en.wikipedia.org/wiki/Matched_filter

  $$h = \frac{1}{\sqrt{s^{\mathrm{H}} R_v^{-1} s}} R_v^{-1} s.$$

    -
    - For finding the signal, with Rv being the noise auto-correlation matrix. S being the signal
    - Otherwise discusses same phenomena as above
- https://physics.stackexchange.com/questions/123665/is-there-a-difference-between-correlation-processing-and-matched-filter-processi
    - Just a cool article

- Preamble
    - https://en.wikipedia.org/wiki/Syncword#:~:text=In%20digital%20communication%2C%20preamble%20is,for%20automatic%20baud%20rate%20detection.

        - Ethernet preamble consists of 56 bits
        - USB preamble is 8 bits at low speed and 32 at high
    - https://www.youtube.com/watch?v=Td2PeutVq7I&ab_channel=EdD
        - Need alternating 1s and 0s to get bit synchronizer and band pass filter started
            - A series of 1s until first 0 as it indicates first bit of transmitted message
            - Then indicator of code starting (0), read 8 bits and (1) to notify end of byte
            - Steady mark region needs to be long enough to sufficiently detect, but short enough that the bandpass filter continues oscillations
    - Recommendations for approaching preamble design?
- Auto correlation function
    - https://en.wikipedia.org/wiki/Autocorrelation
        - Largely repeats the earlier article
    - https://www.sciencedirect.com/topics/chemistry/autocorrelation-function
        - Future reference, primarily research

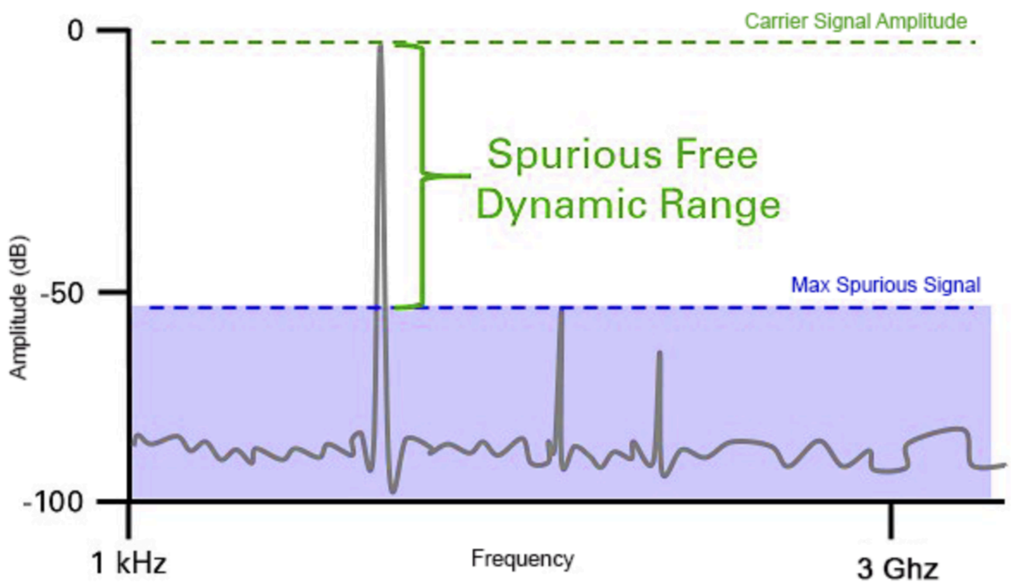
Troy Madden

- Recieving/Transmitting HD radio (NRSC-5)
- https://www.nrscstandards.org/standards-and-guidelines/documents/standards/nrsc-5-d/nrsc-5-d.asp
    - Recieving with an RTL-SDR standalone dongle: https://github.com/theori-io/nrsc5/
        - Purchase required, separate from USRP project
    - Transmitting with GNU Radio: https://github.com/argilo/gr-nrsc5?tab=readme-ov-file
        - Includes HDC encoder (High definition coding)
        - PSD encoder (Program Service Data) - conveys info about the audio such as song title and artist
        - SIG encoder (Station Information Guide) - Album art and station logo
        - SIS encoder (Station Information Service) - Messaging information about the station

Pryce Tharpe

- ADC - Analog to digital converter
- DAC - Digital to analog converter
- SFDR is a Spurious-free dynamic range - decibels relative to the carrier - in simple terms, it measures the purity of a signal.
- GPSDO is a GPS disciplined oscillator - a GPS clock - steers oscillator output to GPS reference with a ppm accuracy of .01(without this reference, the internal clock is accurate to 2.5 ppm)

Beyond 5G, Spring 2024
HD Radio Side Band Isolation and Symbol Synchronization Project
Jack Willard, Pryce Tharpe, Troy Madden



Helped USRP setup and reinstalled and setup UHD on personal device.

Meeting 3/20/24:

Meeting Progress:

Progress was discussed in this meeting. Further more guidance was provided on where the team should place its efforts next. Turns out GNU radio may be able to function on its own.

The team plans to transition Ubuntu to version 22.04 to see if GNU will download correctly there.

How determine what output is satisfactory? - dependent on noise level, higher the output, more confident in detecting what is transmitted

Correlator -> determines witch signal is most likely the one being received and where it is shifted
Matched Filter -> determines filter that will pick up desired signal

Road Blocks:

- UHD not detecting USRP, could be result of information not translating through ethernet line, or unknown USRP IP address
- Setting up GNU radio and reading HD signals.

Objectives before Next meeting:

- Jack - Continue to work on getting the USRP working with Fabio or someone else. For matched filter, work on seeing if we can emulate with python or matlab file the sharp peak. Also, Look into wireless and cellular preambles -> channel estimation, channel state information.
- Pryce - setup GNU radio and get it operational on 22.04

Beyond 5G, Spring 2024
HD Radio Side Band Isolation and Symbol Synchronization Project
Jack Willard, Pryce Tharpe, Troy Madden

- Troy - continue to research HD radio and how to pick up the signal

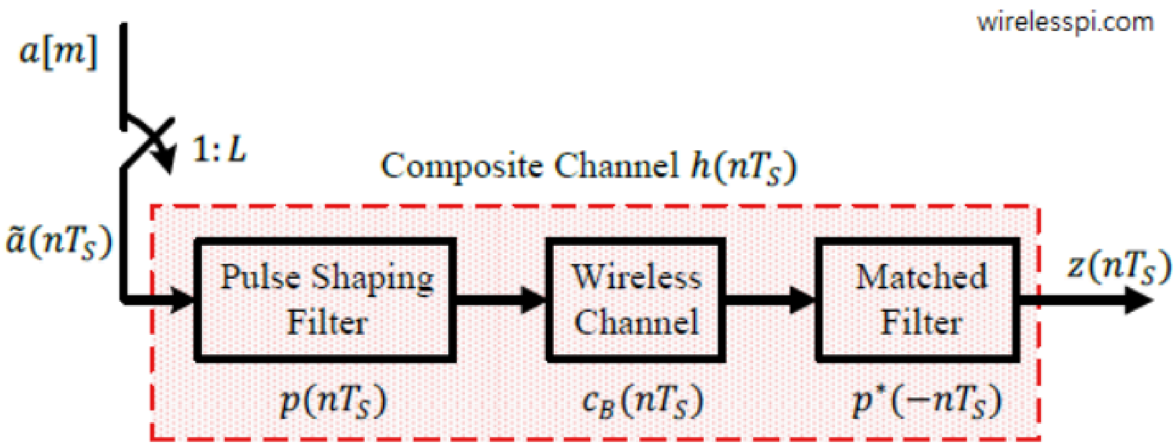
<u>Jack Willard</u>

- Met with Fabio to interface with the USRP.
    - Turns out, our approach and coding ideas were correct, however the ethernet cable we were using was faulty. We replaced the cable and power cord. Successfully interfaced with USRP and received its information (IP address etc.)

```
[INFO] [UHD] linux; GNU C++ version 7.5.0; Boost_1
f98f
--------------------------------------------------
-- UHD Device 0
--------------------------------------------------
Device Address:
    serial: F2A017
    addr: 192.168.10.4
    name:
    type: usrp2
```

- So the idea is that the signal forms being received are known. How does reception of multiple varying signals? Like how are images with information on color received?
- Goal of symbol/frame recovery project (See if can emulate with python or matlab file the sharp peak)
- Look into wireless and cellular preambles
    - https://www.speedguide.net/faq/what-is-preamble-type-when-do-i-need-to-change-it-300
        - Wireless devices not using same preamble types have trouble connecting
        - Overview of IEEE length of cyclic redundancy check (CRC) for communication between access point and roaming wireless adapter (CRC common for detecting transmission errors)
    - https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4970172/
        - Processing time is reduced by code division multiplexing technique where beams are transmitted simultaneously with corresponding Tx beam IDs in preamble
        - Paper discusses increasing number of CBIDs using preamble sequence constructed by a combination of two Zadoff-Chu sequences, a new technique for detecting these is also described
    - https://www.rcrwireless.com/20220216/network-infrastructure/wi-fi/what-is-preamble-puncturing-and-what-does-it-have-to-do-with-wi-fi-7
        - Preamble puncturing optional feature introduced in Wi-Fi 6, will likely be the standard in Wi-Fi 7.
- Channel Estimation
    - https://www.hs-osnabrueck.de/fileadmin/HSOS/Forschung/Recherche/Laboreinrichtungen_und_Versuchsbetriebe/Labor_fuer_Hochfrequenztechnik_und_Mobilkommunikation/Forschung/digitale_Funksysteme/Channel-Estimation.pdf
        - Channel Estimation 3 steps: 1.) Mathematical model for channel is created. This model correlates sent and received signal using channel matrix. 2. A signal known by both sender and receiver is sent by sender over the channel. 3.

Receiver compares the received signal with original signal and figures out the values in channel matrix.
    - Hermitian?
- https://www.mathworks.com/help/lte/ug/channel-estimation.html
    - LTE toolbox in matlab uses orthogonal frequency division multiplexing
    - I believe this tries to estimate the data signal?
- https://wirelesspi.com/channel-estimation-in-wireless-communication/
    - Discusses using least squares and matched filters to identity the channel for single-carrier systems



- Channel State Information
    - https://en.wikipedia.org/wiki/Channel_state_information#:~:text=In%20wireless%20communications%2C%20channel%20state,and%20power%20decay%20with%20distance.
        - CSI is the known channel properties of a communication link (describes how a signal propagates from transmitter to receiver and represents the combined effect of scattering, fading, and power decay with distance)
        - Provides a mathematical description
- Capturing a Preamble
    - https://www.mathworks.com/help/wireless-testbench/gs/triggered-capture-using-preamble-detection.html
        - Shows how to use SDR to capture data from air using preamble detection
    - https://www.mathworks.com/help/wireless-testbench/ug/triggered-wlan-waveform-capture-using-preamble-detection.html
        - shows how to use a software-defined-radio (SDR) to capture a WLAN waveform from the air by detecting the legacy long training field (L-LTF).

Troy Madden

- Continued research on using USRP to receive and decode HD radio
    - https://www.youtube.com/watch?v=BT82rdPAYOg&ab_channel=DoubleALabs
    - ^Tutorial for installation^
    - https://airspy.com/download/

    - https://dotnet.microsoft.com/download…
    - https://drive.google.com/drive/folder…
    - https://github.com/zefie/nrsc5-gui/re...
    - https://sourceforge.net/projects/mingw/
    - People online have shared great resources including drivers, GUI's for tuning radio and displaying HD radio info, etc.
- Installed Ubuntu 22.04 because GNU radio wasn't installing properly on 18.04
    - Installed GNU radio, working on UHD 4.6.0 install
    - Jack got USRP to finally work with uhd find device, next step for that project is get antenna from Ashwin
        - Sent followup email to Ashwin regarding antenna
- Next steps:
    - Crucial -> Need antenna before we can progress much further on USRP/HD radio project
    - Pivot towards frame recovery w/ Jack until we get the antenna

Meeting 3/31 Facetime call

Meeting Progress:

Troy

Talked about symbol sync:

Decided to use Python, Numpy

Looking to implement matched filter by generating np arrays and comparing them by subtracting. Accuracy would be evaluated by summing the difference of their arrays (-1,1 represent differences, 0 means correct bit). Array would be sent back to calling function and bit in the beginning is removed, bit added to the end, everything else shifts.

Jack:

https://ieeexplore.ieee.org/document/9363693 - on IEEE see page 2774 also 2753

Signal Processing occurs at physical layer

15.3.2 PPDU format

Figure 15-1 shows the format for the PPDU including the DSSS PHY preamble, the DSSS PHY header, and the MPDU. The PHY preamble contains the following fields: SYNC and SFD. The PHY header contains the following fields: Signaling (SIGNAL), Service (SERVICE), length (LENGTH), and CRC-16 (CRC). Each of these fields is described in detail in 15.3.3.

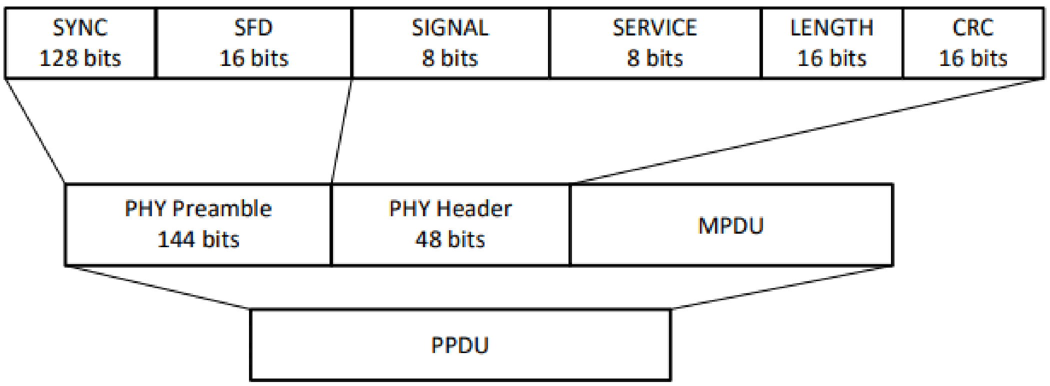


Figure 15-1—PPDU format

Long PPDU format

Long SFD - 1111 0011 1010 0000

16.2.4 PHY/HR/DSSS PHY data scrambler and descrambler

The polynomial $G(z) = z^{-7} + z^{-4} + 1$ shall be used to scramble all bits transmitted. The feedthrough configuration of the scrambler and descrambler is self-synchronizing, which requires no prior knowledge of the transmitter initialization of the scrambler for receive processing. Figure 16-5 and Figure 16-6 show typical implementations of the data scrambler and descrambler, but other implementations are possible.

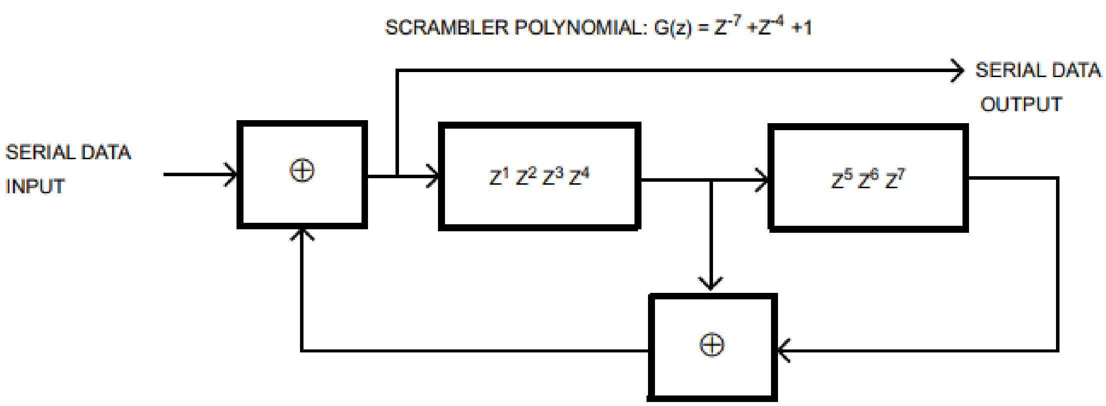


Figure 16-5—Data scrambler

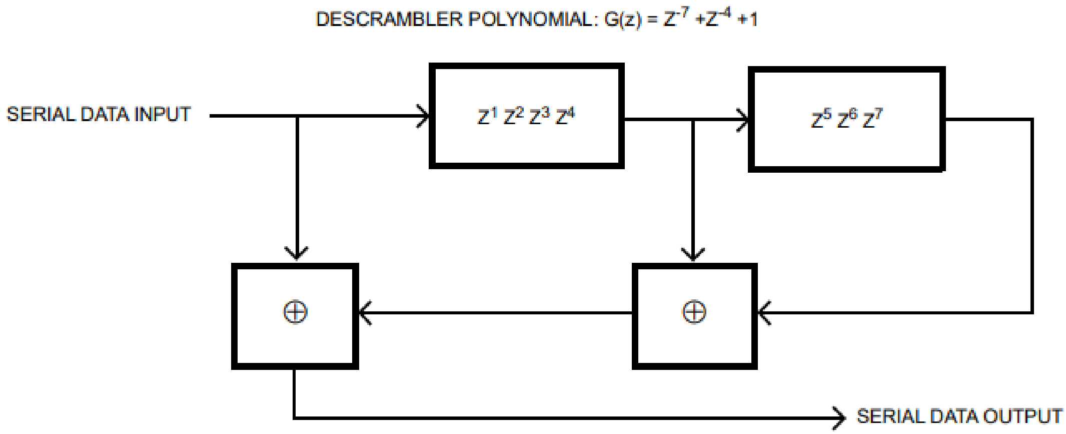


Figure 16-6—Data descrambler

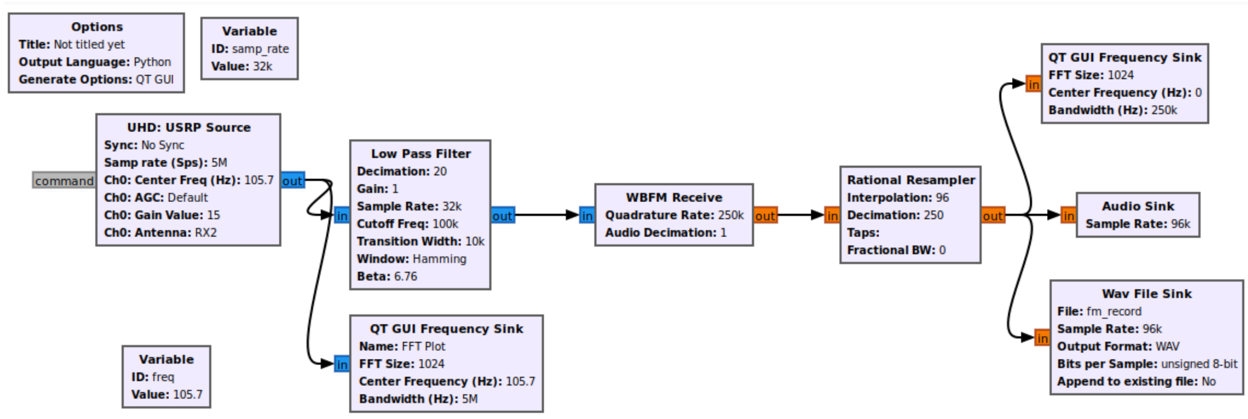
The scrambler shall be initialized as specified in 16.2.3.9 for the short PPDU format and 16.2.3.2 for the long PPDU format. For a long preamble, this shall result in the scrambler registers Z^1 to Z^7 in Figure 16-5 having the data pattern [1101100] (i.e., Z^1= 1... Z^7= 0) when the scrambler is first started. The scrambler shall be initialized with the reverse pattern [0011011] when transmitting the optional short preamble.

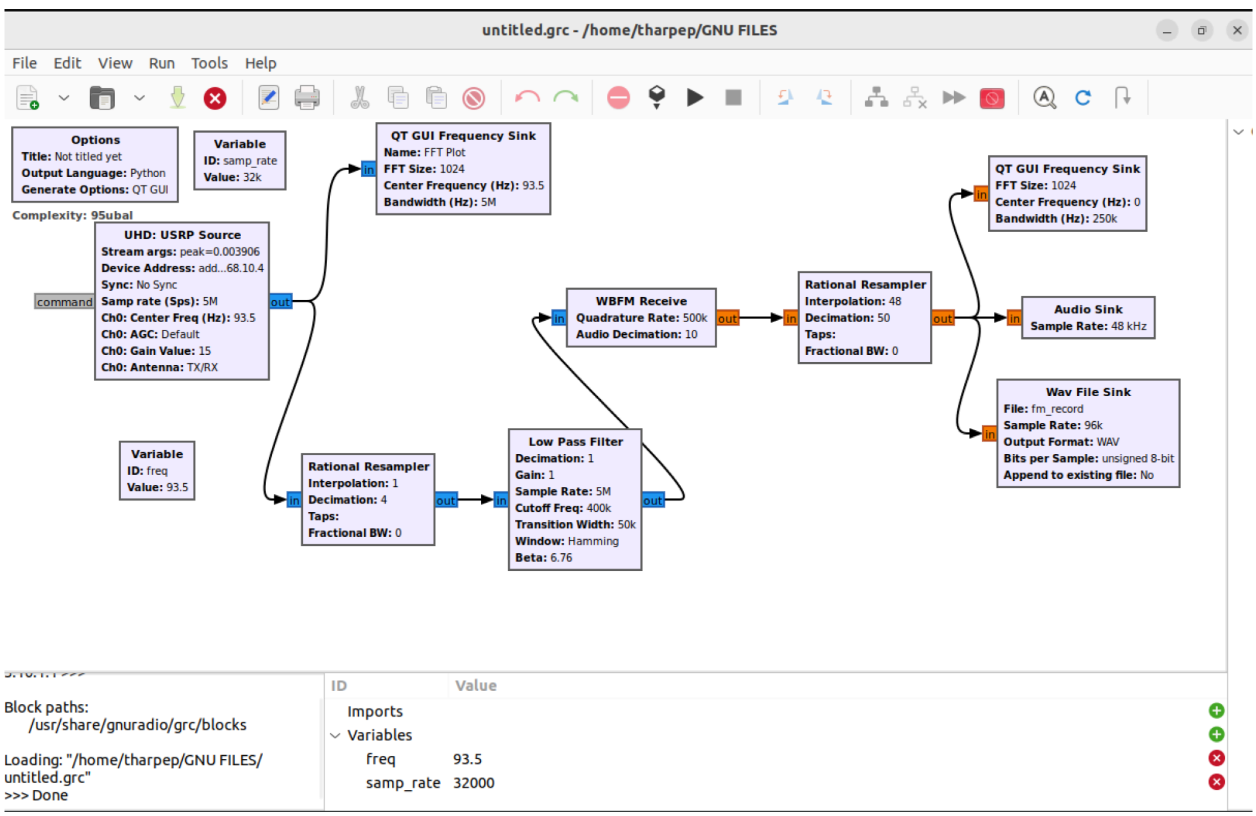
USRP -> Create program to detect energy for smaller bandwidth
For USRP -> use AM radio to transmit 0 vs 1

ON-OFF-Keying (send something oscillating for 1, for 0 send something oscillating with much smaller amplitude)

Map 0 -> 1.0 + noise (noise is Gaussian, could be +0.2 or -0.6), and 1 -> -1.0
How account for continuous noise?

Beyond 5G, Spring 2024
HD Radio Side Band Isolation and Symbol Synchronization Project
Jack Willard, Pryce Tharpe, Troy Madden
Pryce:
https://www.youtube.com/watch?v=KWeY2yqwVA0





Actual python:
https://drive.google.com/file/d/1duGNj9PSkYYo6H33TdVjtIwIGbWGD_0Y/view?usp=drive_link
* Run without errors, but not attempted setup with antenna. When attempted, it should simply play from radio station channel 105.7. Will also show two frequency graphs.

Rational Resampler:

The rational resampler employs a polyphase filterbank architecture to achieve resampling.

It takes an input signal at a specific rate (fs) and outputs a signal at a different rate (fr).

The resampling ratio is defined by a rational number: interpolation / decimation.

Interpolation: Factor by which the sample rate is increased (e.g., interpolation of 2 doubles the rate).
Decimation: Factor by which the sample rate is decreased (e.g., decimation of 4 reduces the rate by a factor of 4).

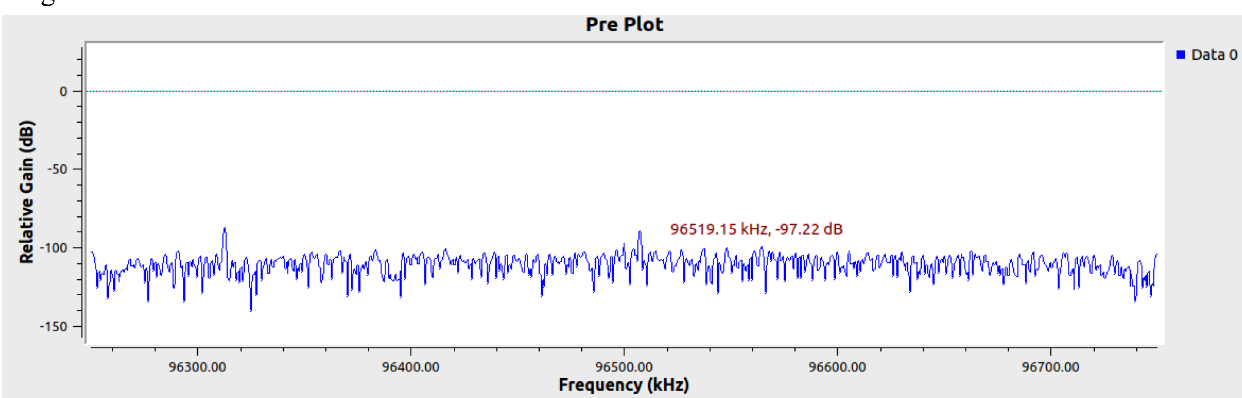
Low Pass Filter
Block Parameters:

The block requires several key parameters to define its filtering behavior:
Sample Rate (Fs): The sampling rate of the input signal (number of samples per second).
Cutoff Frequency (Fc): The frequency above which the filter starts attenuating signals. It's typically specified as a normalized value between 0 and 1 relative to the sample rate (e.g., Fc = 0.5 represents a cutoff frequency at half the sample rate).
Transition Width (Tw): This parameter controls the sharpness of the filter's transition band between the passband (frequencies allowed to pass) and the stopband (frequencies attenuated). A narrower transition width results in steeper attenuation but potentially introduces ringing (oscillations) in the filtered signal.

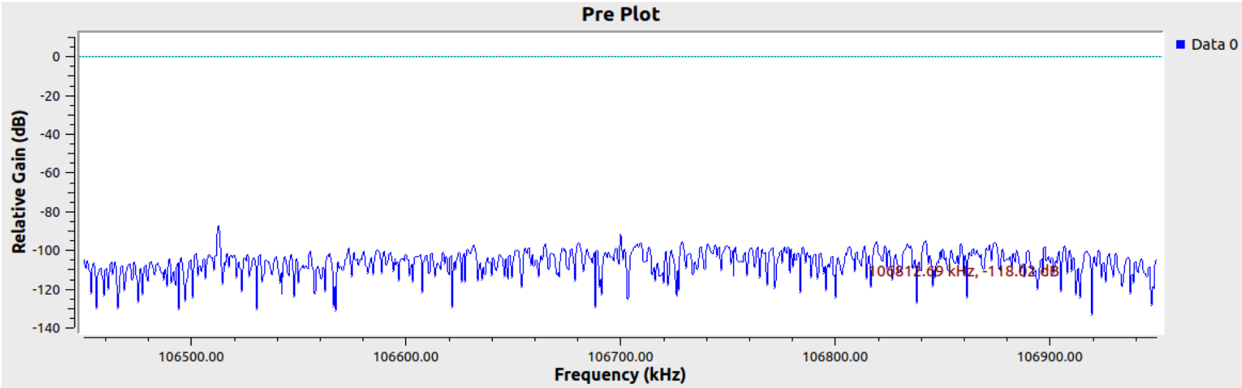
WBFM

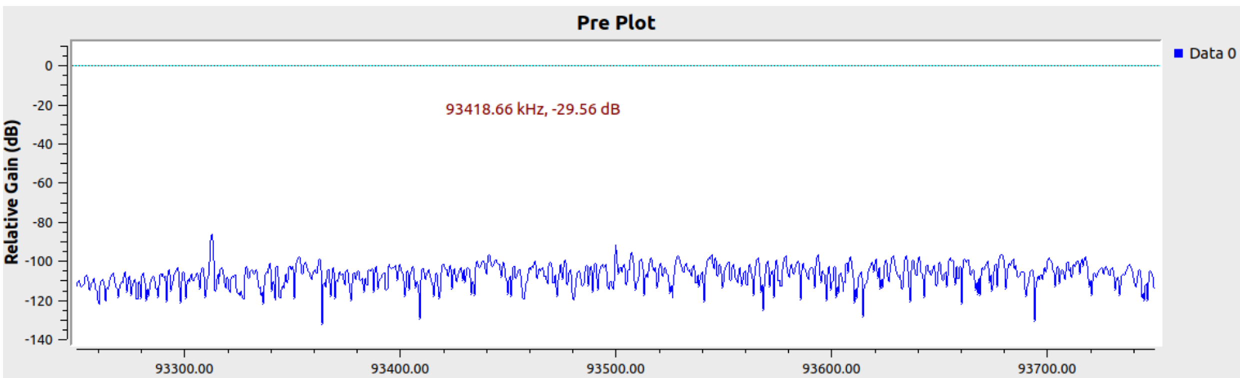
Extracts the audio information embedded within an FM radio signal.
The WBFM Receive block performs the demodulation process, converting the received signal from a complex baseband format (containing both carrier and information) to a recovered audio signal

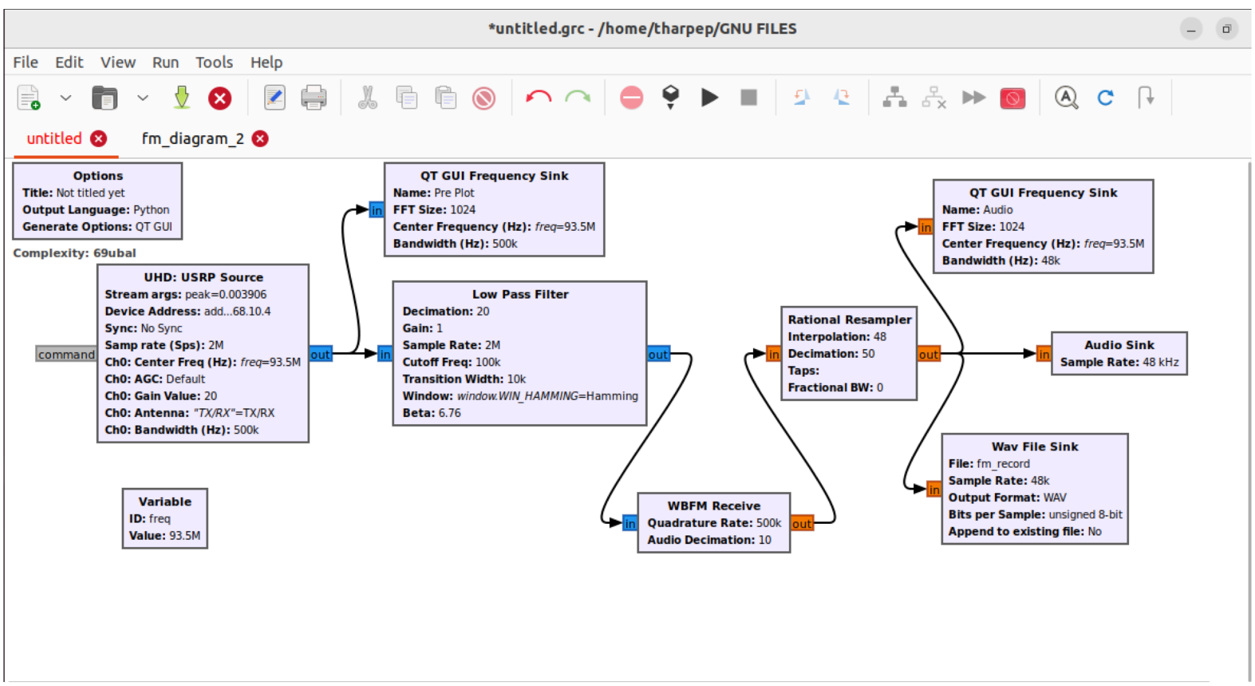
Diagram 1:



Beyond 5G, Spring 2024
HD Radio Side Band Isolation and Symbol Synchronization Project
Jack Willard, Pryce Tharpe, Troy Madden







<u>**Citations Used in Presentation:**</u>

Synchronization Algorithm ->

Beyond 5G, Spring 2024
HD Radio Side Band Isolation and Symbol Synchronization Project
Jack Willard, Pryce Tharpe, Troy Madden

https://cpb-us-w2.wpmucdn.com/sites.gatech.edu/dist/e/466/files/2016/11/sp14_notes3084_10_correlation.pdf

Lanterman, A. (Ed.). (2014, March 8). Correlation and matched filtering - CDN. https://cpb-us-w2.wpmucdn.com/sites.gatech.edu/dist/e/466/files/2016/11/sp14_notes3084_10_correlation.pdf

https://ieeexplore.ieee.org/document/9363693

"IEEE Standard for Information Technology--Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks--Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," in *IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016)* , vol., no., pp.1-4379, 26 Feb. 2021, doi: 10.1109/IEEESTD.2021.9363693.

https://en.wikipedia.org/wiki/Autocorrelation

Wikimedia Foundation. (2024, March 3). *Autocorrelation*. Wikipedia. https://en.wikipedia.org/wiki/Autocorrelation

Generate random binary. Cross-browser testing. (n.d.). https://www.browserling.com/tools/random-bin

https://www.browserling.com/tools/random-bin

USRP ->

Ettus Research. (2024, February 22). *Building and installing the USRP open-source toolchain (UHD and GNU Radio) on linux*. Building and Installing the USRP Open-Source Toolchain (UHD and GNU Radio) on Linux - Ettus Knowledge Base. https://kb.ettus.com/Building_and_Installing_the_USRP_Open-Source_Toolchain_(UHD_and_GNU_Radio)_on_Linux

https://kb.ettus.com/Building_and_Installing_the_USRP_Open-Source_Toolchain_(UHD_and_GNU_Radio)_on_Linux

Beyond 5G, Spring 2024
HD Radio Side Band Isolation and Symbol Synchronization Project
Jack Willard, Pryce Tharpe, Troy Madden

***SBX*. SBX - Ettus Knowledge Base. (2019, November 13). https://kb.ettus.com/SBX**

https://kb.ettus.com/SBX

Muller, M. (2015, October 23). *When trying to use my USRP in GNU Radio, I get a " no devices found for ----->" error*. Stack Overflow. https://stackoverflow.com/questions/33304828/when-trying-to-use-my-usrp-in-gnu-radio-i-get-a-no-devices-found-for

https://stackoverflow.com/questions/33304828/when-trying-to-use-my-usrp-in-gnu-radio-i-get-a-no-devices-found-for

Download_Old_Builds_7_0 – oracle VM virtualbox. (n.d.). https://www.virtualbox.org/wiki/Download_Old_Builds_7_0

https://www.virtualbox.org/wiki/Download_Old_Builds_7_0

***Guided Tutorial Hardware Considerations*. Guided Tutorial Hardware Considerations - GNU Radio. (2023, April 27). https://wiki.gnuradio.org/index.php/Guided_Tutorial_Hardware_Considerations**

https://wiki.gnuradio.org/index.php/Guided_Tutorial_Hardware_Considerations

admin. (2013, November 4). *Tutorial: Creating an FM receiver in GNURADIO using an RTL-SDR source*. RTL-SDR. https://www.rtl-sdr.com/tutorial-creating-fm-receiver-gnuradio-rtl-sdr/

https://www.rtl-sdr.com/tutorial-creating-fm-receiver-gnuradio-rtl-sdr/

Ettus Research. (2012, April 2). *How to build an FM receiver with the USRP in less than 10 minutes*. YouTube. https://www.youtube.com/watch?v=KWeY2yqwVA0

https://www.youtube.com/watch?v=KWeY2yqwVA0

Github Public Repository Link - https://github.com/Jack-Willard/Beyond5GSymbolSync.git

Beyond 5G, Spring 2024
HD Radio Side Band Isolation and Symbol Synchronization Project
Jack Willard, Pryce Tharpe, Troy Madden