Real-Time Spectrum Analysis and STFT Visualization using USRP B210 SDR

Sai Krishna Shanigarapu
Indian Institute of Technology, Hyderabad
(ee23btech11054@iith.ac.in)

***Abstract*—In this project, we implement a programmable, real-time spectrum analyzer using the Software Defined Radio (SDR). We capture the complex base-band IQ samples in the 2.4 GHz band and process them using a Short-Time Fourier Transform (STFT) to visualize the time domain signal. By utilizing the UHD Python API, the system achieves controllable center frequency, channel width, time duration, STFT window size and overlap.**

I. Introduction and Related Work

The growth of wireless devices in 2.4 GHz ISM band has created a need for tools to monitor the spectral usage. The traditional hardware spectrum analyzers are highly expensive and they also lack the flexibility for custom signal processing.

As technology advanced, now we can implement the same traditional hardware components (mixers, filters, modulators) in software using much cheaper Software Defined Radio (SDR) boards [1]. Previously, low-cost SDR's like RTL-SDR were used in this domain. However, it has a very limited bandwidth (around 2MHz). [2]

This project utilizes the Ettus Research USRP B210, a high-performance transceiver. We focus on the receiver chain design, specifically the mathematical transformation of complex IQ data into time-frequency spectrograms using Python.

II. Problem Statement

The objective of this independent project is to design a software-based receiver system with the following constraints:

1) **Hardware Control:** The system must interface with the USRP B210 via the UHD driver, ensuring the Transmitter (Tx) path is blocked to prevent interference.
2) **Programmability:** The user should be able to define the center frequency, channel width, dwell time, window size and overlap. $i.e$, these parameters must be programmable.
3) **Signal Processing:** System should capture the In-phase (I) and Quadrature (Q) samples and then perform an STFT to visualize the power spectral density over time.

III. System Design and Contributions

A. Mathematical Formulation

The main contribution of this project is the handling of complex base-band signals. The USRP down-converts the received signal $r(t)$ to a discrete-time complex sequence $x[n]$:

$$x[n] = I[n] + jQ[n] \tag{1}$$

where $I[n]$ is the real (In-phase) component and $Q[n]$ is the imaginary (Quadrature) component.

To visualize the frequency content change over time, we implemented the discrete Short-Time Fourier Transform (STFT). For a signal $x[n]$ and a sliding window function $w[n]$ (Hann window), the STFT is computed as:

$$X(m,k) = \sum_{n=0}^{N-1} x[n+mH]w[n]e^{-j\frac{2\pi}{N}kn} \tag{2}$$

where:

- m is the time index of the window.
- k is the frequency index.
- H is the hop size (overlap).
- N is the FFT size (set to 1024).

The spectrogram $S(m,k)$, representing power in dB, is derived as:

$$S(m,k) = 20\log_{10}|X(m,k)| \tag{3}$$

B. Hanning Window

To reduce spectral leakage during the STFT computation, we use Hann window which is applied to each frame of IQ samples prior to the FFT. The Hanning (or Hann) window is defined as:

$$w[n] = 0.5\left(1-\cos\left(\frac{2\pi n}{N-1}\right)\right), \qquad 0 \le n \le N-1 \tag{4}$$

where:

- N is the total window length (equal to the STFT FFT size, here, $N = 256$),
- n is the discrete time index of the windowed segment.

C. Parameters Used

TABLE I: Experimental Parameters

Parameter	Value
Center Frequency	2.432 GHz
Sample Rate	5 MHz
Dwell Time	0.2 s
STFT Window Size	256 samples
STFT Overlap	128 samples (50% of window)
Window Type	Hann

D. Results

The system was tuned to a center frequency of $f_c = 2432$ MHz (between Wi-Fi Channels 5 and 6) with a bandwidth of 5 MHz and $f_c = 2402$ MHz for Bluetooth.

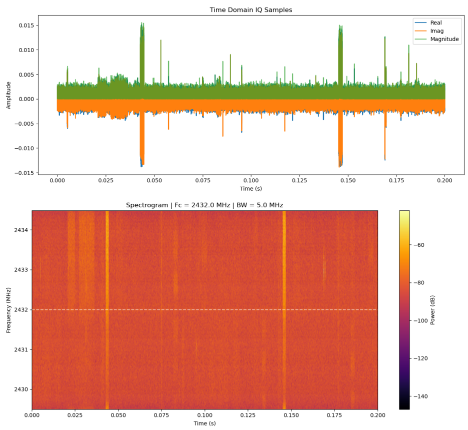

Fig. 1: Time domain and STFT plot for Wi-fi Hotspot

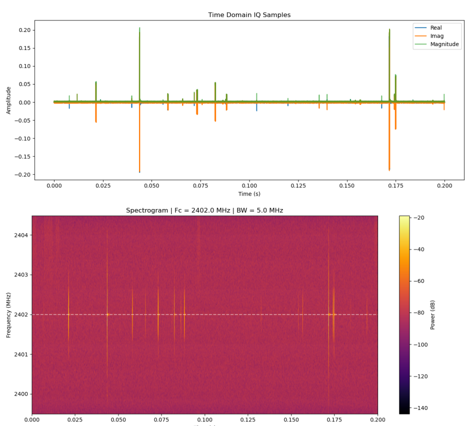

Fig. 2: Time domain and STFT plot for Bluetooth

IV. Conclusion

We have successfully captured the signals using USRP B210 and UHD Python API. The system was able to capture complex base-band IQ samples in the 2.4 GHz ISM band and process them into time–frequency representations using the Short-Time Fourier Transform (STFT). From the Figures 1 and 2, it is evident that the tall vertical lines are repeating approximately after 100ms which represent the Wi-Fi Hotspot. In contrast, the Bluetooth capture demonstrated rapid non-periodic frequency hops spread across the 2.4 GHz band.
Due to the short dwell time of 0.2 s, only a limited portion of signals were captured.

The complete Python source code used for USRP control, IQ capture, and STFT visualization is provided here as part of this project for reproducibility and further experimentation.

V. UHD Python API Installation Steps

To install the UHD Python API, one must have Ubuntu 22.04 version and Python 3.8 or greater (this is mandatory). Follow the steps accordingly.

```
sudo apt update
sudo apt install -y \
    git cmake g++ python3 python3-dev python3-pip \
    libboost-all-dev libusb-1.0-0-dev libncurses5-dev \
    python3-mako libaio-dev
cd ~
git clone https://github.com/EttusResearch/uhd.git
mv uhd uhd_repo
cd ~/uhd_repo/host
mkdir build
cd build
cmake \
  -DENABLE_PYTHON_API=ON \
  -DPYTHON_EXECUTABLE=$(which python3) \
  -DPYTHON3_SITEPKG=/usr/local/lib/python3.10/dist-packages \
  ..
make -j$(( $(nproc) / 2 ))
sudo make install
sudo ldconfig
cd ~/uhd_repo/host/build/python
sudo python3 setup.py install
```

If there are no errors, do the following to update the USRP USB permissions. USB 3.0 or 3.1 is must.

```
# Copy Ettus USRP udev rules
sudo cp ~/uhd_repo/host/utils/uhd-usrp.rules /etc/udev/rules.d/

# Reload udev rules
sudo udevadm control --reload-rules
sudo udevadm trigger

# Add current user to required groups
sudo usermod -aG plugdev $USER
sudo usermod -aG dialout $USER

# Reboot the system
sudo reboot now
```

References

[1] J. Mitola, ”The software radio architecture,” in *IEEE Communications Magazine*, vol. 33, no. 5, pp. 26-38, May 1995.
[2] T. A. Kazi and M. Noman, “Implementation of Wide Band FM Receiver on RTL-SDR,” *International Journal of Engineering Research and Technology (IJERT)*, vol. 5, no. 5, pp. 580–584, 2016.
[3] Ettus Research, ”USRP B200/B210 Knowledge Base,” National Instruments, 2024.
[4] B. E. Ward, “USRP Tutorial,” *PySDR: A Practical Guide to SDR*, 2025. [Online]. Available: Here. [Accessed: Nov. 30, 2025]