SDR Based Spectrum Sensing

Amitesh Singh [B.tech ECE]
211010214

Bhavesh Balendra[B.tech ECE]
211010221

***Abstract*—In the given problem we are detecting the unused band using usrp as transmitter and RTL SDR to detect the signal. Using the SDR Sharp to detect and identify the frequency in the given spectrum range.**

Keywords—USRP, SDR-Sharp, Spectrum Range

I. Introduction

The demand for higher data rates is increasing as more multimedia applications are being used, but the natural frequency spectrum has limitations that make it difficult to accommodate this demand with current static frequency allocation schemes. To address this, innovative techniques are needed to make the most of the available spectrum. As interest in wireless communication continues to rise, the demand for radio spectrum is expected to keep increasing. Traditional spectrum management, which assigns exclusive licenses to operate in specific frequency bands, has become inflexible and finding available bands for new services or enhancing existing ones is difficult. Additionally, licensed spectrum is often underutilized due to inefficient fixed frequency allocations. To address this, regulatory bodies are allowing secondary (unlicensed) systems to opportunistically utilize unused primary (licensed) bands, referred to as white spaces. The IEEE has formed a group (IEEE 802.22) to develop an air interface for efficient secondary access to the spectrum. Various techniques have been implemented to facilitate this, such as database registries and beacon signals, which enable primary systems to provide secondary users with current spectrum usage information either by broadcasting it on regional beacons or registering it in a centralized database.

II. SDR

Joseph Mitola is credited with inventing Software Defined Radio (SDR) in the early 1990s. He defined SDR as a type of radio that can be reprogrammed and reconfigured through software. Mitola envisioned an ideal SDR system with an antenna and Analog to Digital Converter (ADC) as the physical components on the receiver side, and a Digital to Analog Converter (DAC) and transmitting antenna on the transmitter side. The remaining functions would be held by reprogrammable processors.

III. SDR

SDR# (SDR Sharp) is a popular software application for software-defined radio (SDR) systems. It is a free, open-source software that allows users to receive and decode a wide range of radio signals using compatible SDR hardware..

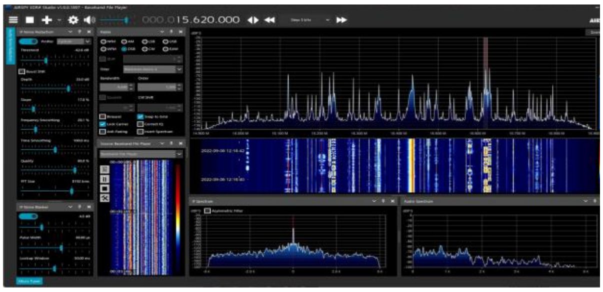

Fig.1 SDR #

IV. Coginitive

Cognitive radio is a type of wireless communication system that uses artificial intelligence techniques to automatically detect available frequencies in the radio spectrum and adapt to changing conditions in the environment. The cognitive radio technology aims to efficiently utilize radio spectrum by allowing secondary (unlicensed) users to use the idle channels of primary (licensed) users when they are not in use. The technology works by continuously sensing the radio frequency spectrum and using machine learning algorithms to identify idle or underutilized frequency bands. The cognitive radio then uses these available frequency bands to establish a communication link, while avoiding interfering with primary users.Cognitive radio technology has several potential benefits, including improved spectrum utilization, increased network capacity, and enhanced reliability and security. It also enables dynamic spectrum access, which allows multiple users to share the same spectrum in real-time without causing interference to one another.

V. RTL-SDR

RTL-SDR (Software Defined Radio) is a type of low-cost radio communication system that uses a TV tuner dongle based on the Realtek RTL2832U chipset to receive and decode various radio signals RTL-SDR technology allows users to receive and decode a wide range of radio signals, including FM radio, digital audio broadcasting (DAB), and even aircraft and satellite transmissions. By connecting the TV tuner dongle to a computer or mobile device and using dedicated software, users can tune into various radio frequencies, visualize signals, and demodulate them into digital formats.

The popularity of RTL-SDR is due to its low cost, versatility, and ease of use. It has become a popular tool for hobbyists, amateur radio enthusiasts, and security researchers interested in exploring the radio frequency spectrum and experimenting with various radio technologies.

Fig.2 RTL-Sdr

VI. USRP

USRP (Universal Software Radio Peripheral) is a type of software-defined radio (SDR) system that allows users to transmit and receive radio signals using configurable software running on a programmable hardware platform. It was developed by Ettus Research, which is now a part of National Instruments. USRP hardware consists of a motherboard with one or more daughterboards that can be configured for different frequency ranges and modulation schemes. The hardware platform is programmable, allowing users to customize the signal processing and control the data flow between the hardware and software layers. The USRP system is typically used by researchers, engineers, and hobbyists who need to test and develop new wireless communication systems or algorithms. It is a versatile and flexible tool that supports a wide range of applications, including wireless standards testing, cognitive radio, and signal intelligence. USRP technology has become popular due to its flexibility, high-performance capabilities, and ease of use. It has been used in a wide range of research and development projects in fields such as wireless communications, aerospace, defense, and education.

VII. Implementation

As we require a USRP transmitter, RTL-sdr, Ethernet cable, RTL-Sdr Antenna, PC/Laptop. Here we will use two laptops, one for connecting the USRP transmitter other for visualizing the signal using software (SDR#).For the first laptop, install SDR# , You must have the Microsoft .NET 6.0 x86 Desktop Runtime installed to use SDRSharp. Most modern PCs will have automatically updated to this already, unless you have specifically blocked these updates on your PC. download sdrsharp-x86.zip. Install by double click on install -rtlsdr.bat after extracting the file. After this download zadig for manual driver installation .Plug in your dongle .Open Zadig >>goto options >>list all devices .Then, Select "Bulk-In, Interface (Interface 0)" from the drop down list. Then replace the driver . Open SDRSharp.exe and set the "Source" drop down box to 'RTL-SDR USB'. This "Source" tab is on the lower left menu bar by default. On newer version of SDR# you may need to scroll down in the drop down box a little to find the RTL-SDR USB entry. Adjust RF gain setting . To receive HF signals below 24 MHz's Now for the second laptop connect the ethernet cable with USRP and laptop . Make the IP address of laptop same with the IP address written on the USRP transmitter .Ping the IP address in the command prompt to ensure that IP address of the laptop does not get changed to it's default value . Now open the MATLAB ,check whether the USRP is connected to the system or not by typing the command "findsdru" in the command window of MATLAB .Here we are generating the sinusoidal signal by entering some of the parameter of USRP . After running the code USRP transmitter will generate the signal and it will be captured by the RTL-Sdr antenna and the signal is visualized by the SDR# software .Here we have are using loop for generating continuous signal . Signal is seen after a fixed interval of time . On newer version of SDR# you may need to scroll down in the drop down box a little to find the RTL-SDR USB entry. Adjust RF gain setting . To receive HF signals below 24 MHz. Now for the second laptop connect the ethernet cable with USRP and laptop . Make the IP address of laptop same with the IP address written on the USRP transmitter .Ping the IP address in the command prompt to ensure that IP address of the laptop does not get changed to it's default value .Now open the MATLAB ,check whether the USRP is connected to the system or not by typing the command "findsdru" in the command window of MATLAB .Here we are generating the sinusoidal signal by entering some of the parameter of USRP . After running the code USRP transmitter will generate the signal and it will be captured by the RTL-Sdr antenna and the signal is visualized by the SDR# software .Here we have are using loop for generating continuous signal . Signal is seen after a fixed interval of time .

VIII. SIMULATION, RESULT AND ANALYSIS

Fig. 3 Setup

```
%% Create sine wave
fs = 1e6;                  % Sample rate (Hz)
f = 100e3;                 % Frequency of sine wave (Hz)
t = 0:1/fs:1;              % Time vector
x = sin(2*pi*f*t);         % Sine wave signal

%% Setup USRP object
tx = comm.SDRuTransmitter(Platform="N200/N210/USRP2",IPAddress="192.168.10.2
tx.CenterFrequency = 0.7e9;                % Transmit center frequency (Hz)
tx.InterpolationFactor = 100;              % Interpolation factor
tx.Gain = 30;                              % Transmit gain (dB)
tx.ClockSource = 'Internal';               % Use internal clock

%% Transmit signal
for i = 1 : 100
    tx(x');                     % Transmit signal
    release(tx);               % Release USRP object
end
```

Fig.4 Code

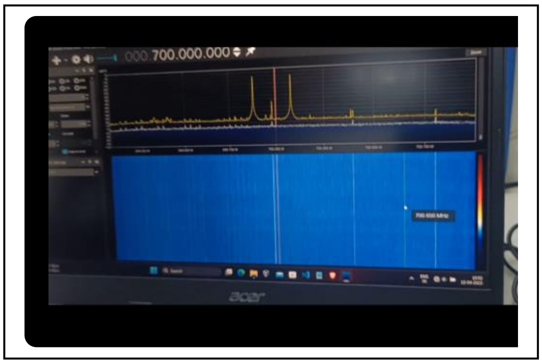

Fig. Output

IX. CONCLUSION

In this work, an implementation of simple SDR Spectrum sensing using the RTL SDR has been done using SDR Sharp for analying the frequecy. Using this RTL SDR can detect the sigal in the range of 24MHz to 0.7GHz un used bands. Using this we can implement a cheap way of detecting frequecy and identify the un used bands and use that band effeciency.

X. REFERENCES

[1] https://www.rtl-sdr.com/sdrsharp-users-guide/.

[2] https://www.rtl-sdr.com/rtl-sdr-quick-start-guide/.

[3] https://www.youtube.com/watch?v=xQVm-YTKR9s&ab_channel=AndreasSpiess.