**2.2.3 CODE FOR Tx-**

% Transmitter Code using USRP B210

% Define the audio sample rate and samples per frame

audioSampleRate = 200e3; % Higher sample rate for FM transmission (200 kHz)

samplesPerFrame = 1024; % Number of samples per frame for live audio

% Create an audioDeviceReader object to capture live audio

audioReader = audioDeviceReader('SampleRate', audioSampleRate, 'SamplesPerFrame', samplesPerFrame);

% Create a figure for live plotting

figure;

hPlot = plot(zeros(samplesPerFrame, 1)); % Initialize the plot with zeroed data

ylim([-1, 1]); % Set amplitude limits for the plot

xlabel('Sample Number');

ylabel('Amplitude');

title('Live Audio Data (Voice)');

% USRP B210 Serial Number (replace with your actual serial number)

usrpSerialNumber = '3166CCB'; % Replace with your actual serial number

% Set FM transmission frequency (choose a valid FM frequency within the range)

fmFrequency = 88.9e6; % FM frequency (e.g., 98.3 MHz for Radio Mirchi)

% Set initial MasterClockRate and calculate InterpolationFactor

masterClockRate = 20e6; % Use 20 MHz master clock rate for both transmitter and receiver

interpolationFactor = round(masterClockRate / audioSampleRate); % Round to nearest integer

% Adjust MasterClockRate if necessary to ensure a valid InterpolationFactor

while (interpolationFactor > 512 || interpolationFactor < 1) && masterClockRate > 5e6

masterClockRate = masterClockRate - 1e6; % Reduce clock rate step-by-step

interpolationFactor = round(masterClockRate / audioSampleRate); % Recalculate factor

end

% Create a USRP B210 transmitter object

usrp = comm.SDRuTransmitter(...

'Platform', 'B210', ...

'SerialNum', usrpSerialNumber, ... % USRP serial number

'CenterFrequency', fmFrequency, ... % Set FM center frequency (98.3 MHz)

'Gain', 30, ... % Transmission gain

'MasterClockRate', masterClockRate, ... % Set same master clock rate for both transmitter and receiver

'InterpolationFactor', interpolationFactor); % Use rounded interpolation factor

% Create FM modulator for audio (Remove Stereo property)

fmModulator = comm.FMModulator(...

'SampleRate', audioSampleRate, ... % Audio sample rate

'FrequencyDeviation', 75e3); % Frequency deviation for FM radio (standard 75 kHz)

% Start the timer

disp('Starting live voice transmission...');

startTime = tic; % Record the starting time

% Transmit for 30 seconds

transmitDuration = 30; % Transmission duration in seconds

try

while toc(startTime) < transmitDuration

% Read audio data from the microphone

audioData = audioReader();

% Update the plot with live audio data

set(hPlot, 'YData', audioData);

drawnow; % Refresh the plot immediately

% FM modulate the audio data

fmSignal = fmModulator(audioData);

% Transmit the FM modulated signal via USRP

usrp(fmSignal);

end

catch ME

% Display error message in case of failure

disp('An error occurred during transmission:');

disp(ME.message);

end

% Release system resources

release(audioReader);

release(fmModulator);

release(usrp);

% Display total time elapsed

disp(['Total time elapsed: ', num2str(toc(startTime), '%.2f'), ' seconds']);

disp('Live voice transmission finished.');

**Code For Rx-**

% Parameters

serialNumber = '3166CFB'; % Replace with your USRP B210 serial number

sampleRate = 200e3; % Sampling rate (200 kHz typical for FM)

audioSampleRate = 48e3; % Audio playback rate

frameDuration = 0.1; % Frame duration (in seconds)

centerFrequency = 88.9e6; % Replace with your desired station frequency (e.g., 98.3 MHz for Radio Mirchi)

% Derived parameters

samplesPerFrame = sampleRate \* frameDuration;

% Verify USRP availability

try

% Create USRP receiver object

radio = comm.SDRuReceiver(...

'Platform', 'B210', ...

'SerialNum', serialNumber, ...

'MasterClockRate', 20e6, ...

'DecimationFactor', round(20e6 / sampleRate), ...

'Gain', 30, ...

'SamplesPerFrame', samplesPerFrame, ...

'CenterFrequency', centerFrequency); % Set the specific station frequency

catch ME

disp('Error initializing USRP Receiver:');

disp(ME.message);

return;

end

% Create audio player

audioPlayer = audioDeviceWriter('SampleRate', audioSampleRate);

% FM demodulator

fmDemod = comm.FMDemodulator(...

'SampleRate', sampleRate, ...

'FrequencyDeviation', 75e3);

disp(['Listening to live radio at ', num2str(centerFrequency / 1e6), ' MHz...']);

% Start the reception

try

while true % Continuous listening

% Receive data from the USRP

rxData = radio();

if ~isempty(rxData)

% Convert to floating-point

rxDataFloat = double(rxData); % Ensure data is in floating-point format

% FM Demodulation

demodulatedSignal = fmDemod(rxDataFloat);

% Resample to match audio playback rate

audioSignal = resample(demodulatedSignal, audioSampleRate, sampleRate);

% Play audio

audioPlayer(audioSignal);

else

disp('No data received. Adjust antenna or gain settings.');

end

end

catch ME

disp('An error occurred during FM reception:');

disp(ME.message);

end

% Release resources

release(radio);

release(audioPlayer);

disp('FM Radio reception finished.');