```
clear, clc;

%The Cosine signal passes through a Turner Gain of 10dB, bandpass filter,
%Mixer of 3.5MHz, Lowpass filter BW=5 MHz, and Variable gain amplifier.
%For the Demodulator the signal then passes through an ADC 28.8MHz, splits
%into both a cos/sine mixer, lowpass filter, and decimation downsapling
%filter. The output is two signals (I & Q) which can be added to create a
%complex signal.

%Below I created a Cosine signal with a 4 GHz sampling rate and 100 MHz
%Frequency. I created a frequency vs domain graph. I use the 'plotter'
%function to call it.
Fs = 4e9;
dt = 1/Fs;
t =  (0:dt:0.0002-dt)';
Fm = 100e6;
sig1 = cos(2*pi*Fm*t);
figure(1);
plotter(sig1, 'Initial Cosine Signal')


%RAFAEL R820T2 TUNER SECTION

%Turner gain of 10dB (lna)
agc1 = comm.AGC('AveragingLength',10);
sigAGC = agc1(sig1);
plotter(sigAGC, 'Turner Gain')

%RF Filter of 6MHZ(bandpass filter)
hRF = firpm(100,[0 Fm-50e6 Fm-3e6 Fm+3e6 Fm+50e6 (Fs/2)]*2/Fs, [0 0 1 1 0 0]);
figure(2);
freqz(hRF);
RF = filter(hRF,1,sigAGC);
figure(3);
plotter(RF, 'RF Filter')

%Mixer of 3.5MHz
Fif =  3.57e6;
Flo = Fm - Fif;
mix = RF.*cos(2*pi*Flo*t);
figure(4);
plotter(mix, 'Cosine Mixer')

%IF Filter of BW=5MHz (lowpass filter)
hIF = firpm(150,[Fif-2.5e6 Fif+2.5e6 Fif+30e6 (Fs/2)]*2/Fs, [1 1 0 0]);
figure(5);
freqz(hIF);
IF = filter(hIF,1,mix);
plotter(IF, 'IF Filter')

%Variable Gain Amplifier
%vga1 = serdes.VGA('Gain', 10)   defaults to one
vga = IF * 10;
```

```matlab
figure(5);
plotter(vga, 'VGA')

%RTL2832U DEMODULATOR

%Decimate by a factor of 275
f = vga(1:275:end);

%ADC of 8-bit and 28.8MHz range
B = 8;
N = 2^(B-1);
adcout  = (round(f .* N)) ./ N;
error = f-adcout;
plotter(adcout, 'ADC')

%Cosine and Sine Mixers
n = (1:length(adcout)).';
cos1 = adcout.*cos(2*pi*(3.57e6/28.8e6)*n);
sin1 = adcout.*sin(2*pi*(3.57e6/28.8e6)*n);
plotter(cos1, 'Cosine Mixer')
plotter(sin1, 'Sine Mixer')

%Decimation/downsample filter for every 10
real = decimate(cos1, 10);
img = decimate(sin1, 10);
plotter(real, 'Real Output')
plotter(img, 'Imaginary Output')

%Combining both signals (I & Q) to get original signal
out = real + 1j * img;
plotter(out, 'Final Complex Signal')
```

```matlab
function plotter(input, name)
%DOCUMENTATION
% Plots the input signal in the frequency domain.
% Gets the appropriate length for plotting.
% Then plots the graph.
f = -(length(input)/2)+1:1:(length(input)/2);
Spectrum = 20*log10(abs(fft(input)));
Spectrum = fftshift(Spectrum);
plot(f,Spectrum);
title(name);
xlabel('Frequency (Hz)');
ylabel('Amplitude (dB)');
end
```