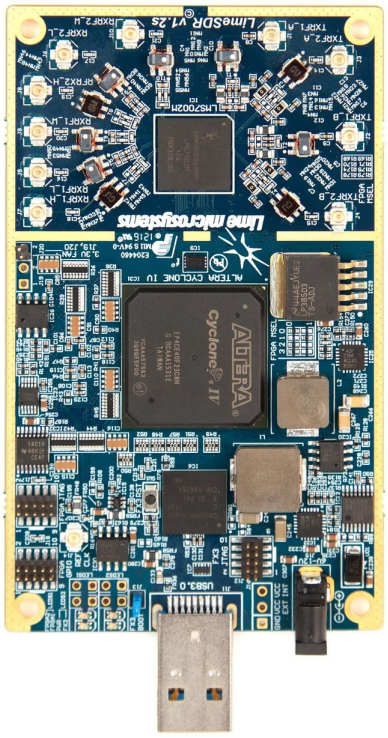
RADIOCOMMUNICATIONS: TRANSMITTERS AND RECEIVERS

LIMESDR-USB HARDWARE DESCRIPTION

LimeSDR is a low cost, open source, apps-enabled software defined radio (SDR) platform that can be used to support just about any type of wireless communication standard.

Features & Specifications

- **RF Transceiver:** Lime Microsystems LMS7002M MIMO FPRF (Datasheet)
- **FPGA:** Altera Cyclone IV EP4CE40F23 – also compatible with EP4CE30F23
- **Memory:** 256 MBytes DDR2 SDRAM
- **USB 3.0 controller:** Cypress USB 3.0 CYUSB3014-BZXC
- **Oscillator:** Rakon RPT7050A @30.72MHz (Datasheet)
- **Continuous frequency range:** 100 kHz – 3.8 GHz
- **Bandwidth:** 61.44 MHz
- **RF connection:** 10 U.FL connectors (6 RX, 4 TX)
- **Power Output (CW):** up to 10 dBm
- **Multiplexing:** 2×2 MIMO
- **Power:** micro USB connector or optional external power supply
- **Status indicators:** programmable LEDs
- **Dimensions:** 100 mm x 60 mm

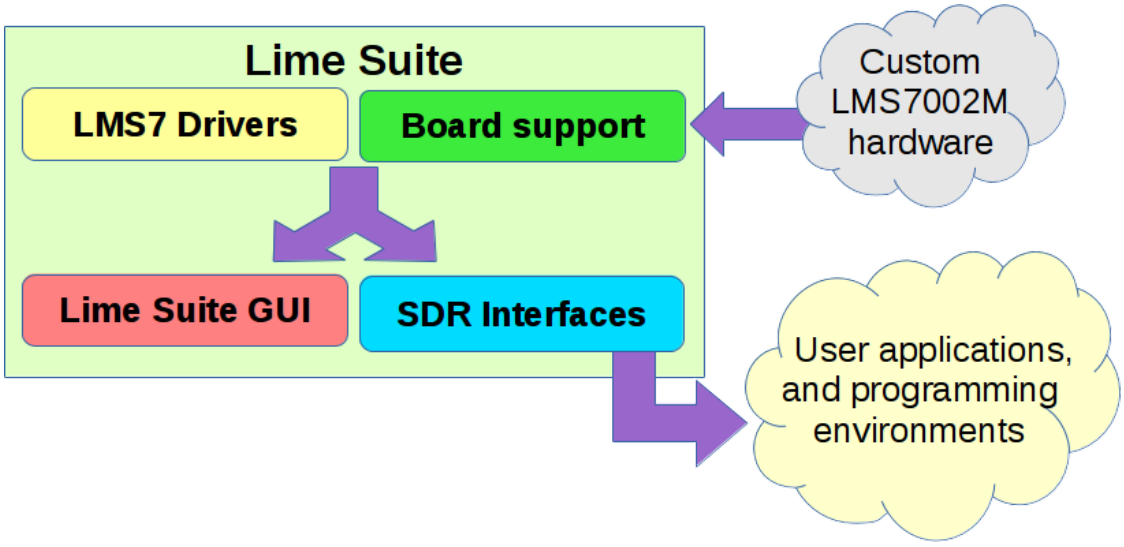


LIME SUITE

Lime Suite is a collection of software supporting several hardware platforms based on the LMS7002M transceiver RFIC, such as LimeSDR family. It contains the following components:

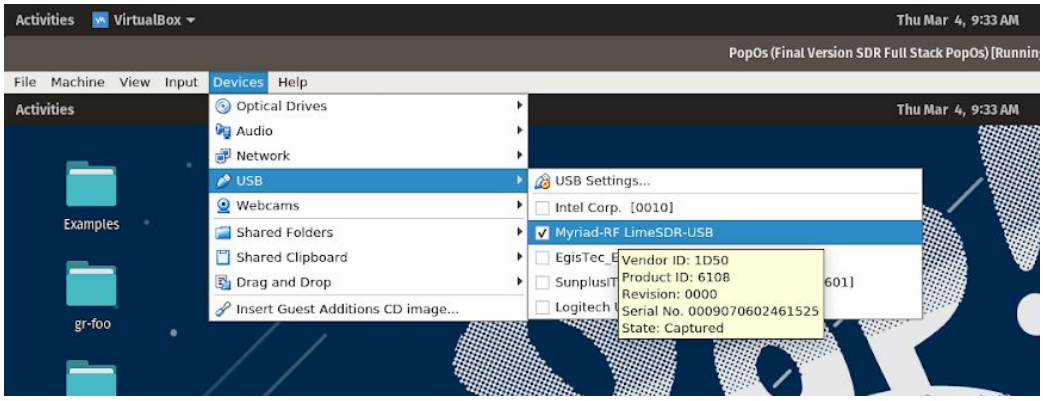
- **LimeSuite library** that provides C-style API;
- **LimeSuiteGUI** application for accessing low-level chip and board settings, displaying FFT, updating firmware and more;
- **SoapyLMS** plugin for LimeSDR support in SoapySDR;
- **LimeUtil** command line tool for listing LimeSDR devices and updating firmware;
- **LimeQuickTest** application to run some basic tests;
- **LimeSuite API examples** (basicRX, basicTX, singleRX, dualRXTX, gpio_example);
- **Octave plugin** (provides some basic functionality only);

LimeSuite is the glue that binds Lime RFIC based hardware designs with the SDR application ecosystem. Hardware developers can create hardware with the aide of debugging tools and driver APIs. Users can interact with LimeSDR and other devices in existing SDR applications. And application developers can create software in a variety of APIs and programming environments.



TESTING YOUR LIMESDR

1. Check that you VirtualBox machine has the USB Devices activated:
   Select Devices -> USB -> Myriad-RF LimeSDR-USB



2. Check which version of LimeUtil is installed:
   Open a Linux Terminal and run `LimeUtil --info`

You should see a message similar to this

```
popuser@pop-os:~$ LimeUtil --info
######################################################
## LimeSuite information summary
######################################################

Version information:
  Library version:      v20.01.0-myriadrf1~bionic
  Build timestamp:      2020-01-28
  Interface version:    v2020.1.0
  Binary interface:     20.01-1

System resources:
  Installation root:    /usr
  User home directory:  /home/popuser
  App data directory:   /home/popuser/.local/share/LimeSuite
  Config directory:     /home/popuser/.limesuite
  Image search paths:
     - /home/popuser/.local/share/LimeSuite/images
     - /usr/share/LimeSuite/images

Supported connections:
   * FT601
   * FX3
   * PCIEXillybus
```

3. Use LimeUtil to check if your device is being recognized:
   Open a Linux Terminal and run `LimeUtil --find`

You should see a message similar to this

```
popuser@pop-os:~$ LimeUtil --find
  * [LimeSDR-USB, media=USB 3.0, module=FX3, addr=1d50:6108, serial=0009070602461525]
```

4. To develop SDR applications SoapySDR is used. SoapySDR is an open-source generalized API and runtime library for interfacing with SDR devices. To check if we can use SoapySDR we can use SoapySDRUtil.
   Open a Linux Terminal and run `SoapySDRUtil --info`

You should see a message similar to this

```
popuser@pop-os:~$ SoapySDRUtil --info
######################################################
##     Soapy SDR -- the SDR abstraction library     ##
######################################################

Lib Version: v0.7.1-myriadrf1~ubuntu18.04
API Version: v0.7.1
ABI Version: v0.7
Install root: /usr
Search path:  /usr/lib/x86_64-linux-gnu/SoapySDR/modules0.7
Search path:  /usr/local/lib/x86_64-linux-gnu/SoapySDR/modules0.7            (missing)
Search path:  /usr/local/lib/SoapySDR/modules0.7                             (missing)
Module found: /usr/lib/x86_64-linux-gnu/SoapySDR/modules0.7/libLMS7Support.so (20.01.0)
Available factories... lime
Available converters...
 -  CF32 -> [CF32, CS16, CS8, CU16, CU8]
 -  CS16 -> [CF32, CS16, CS8, CU16, CU8]
 -  CS32 -> [CS32]
 -   CS8 -> [CF32, CS16, CS8, CU16, CU8]
 -  CU16 -> [CF32, CS16, CS8]
 -   CU8 -> [CF32, CS16, CS8]
 -   F32 -> [F32, S16, S8, U16, U8]
 -   S16 -> [F32, S16, S8, U16, U8]
 -   S32 -> [S32]
 -    S8 -> [F32, S16, S8, U16, U8]
 -   U16 -> [F32, S16, S8]
 -    U8 -> [F32, S16, S8]
```

5. Now let's check that SoapySDR can detect our LimeSDR
   Open a Linux Terminal and run `SoapySDRUtil --find="driver=lime"`

You should see a message similar to this

```
popuser@pop-os:~$ SoapySDRUtil --find="drive=lime"
######################################################
##     Soapy SDR -- the SDR abstraction library     ##
######################################################

Found device 0
  addr = 1d50:6108
  driver = lime
  label = LimeSDR-USB [USB 3.0] 9070602461525
  media = USB 3.0
  module = FX3
  name = LimeSDR-USB
  serial = 0009070602461525
```

REFERENCES

[1] https://limemicro.com/products/boards/limesdr/

[2] https://wiki.myriadrf.org/LimeSDR-USB_hardware_description

[3] https://wiki.myriadrf.org/LimeMicro:LMS7002M_Datasheet

[4] https://github.com/myriadrf/LimeSuite

[5] https://myriadrf.org/news/limesuite-driver-architecture/

LABORATORY 2
TESTING LIMESDR HARDWARE

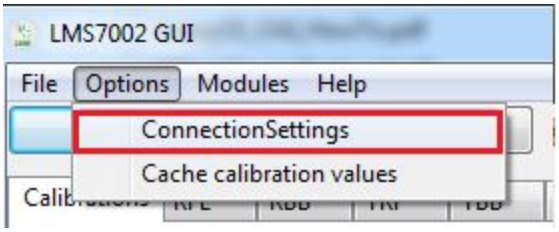
1. LOOPBACK TEST

In this test:

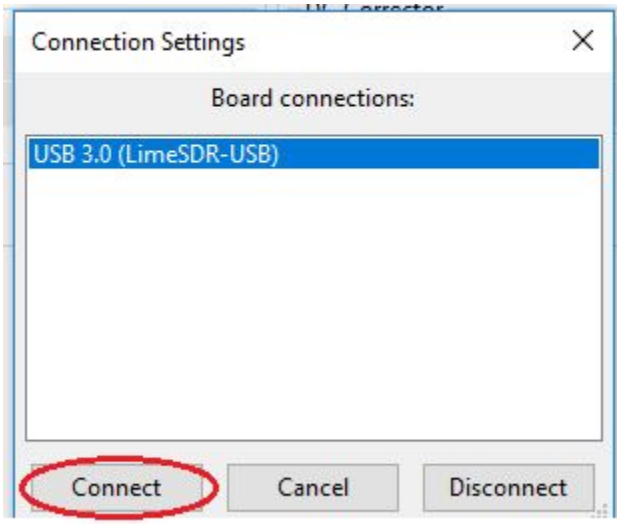
- A waveform is sent from the host computer to the LimeSDR-USB board
- WCDMA signal is generated in the FPGA
- An on-board RF switch connects TX to RX
- The received signal is displayed by the FFT viewer

1.1 Connect to board

1. Launch Lime Suite GUI (LimeSuiteGUI.exe on Windows or just ‘LimeSuiteGUI’ on Linux)
2. From menu bar select: ‘Options->ConnectionSettings’

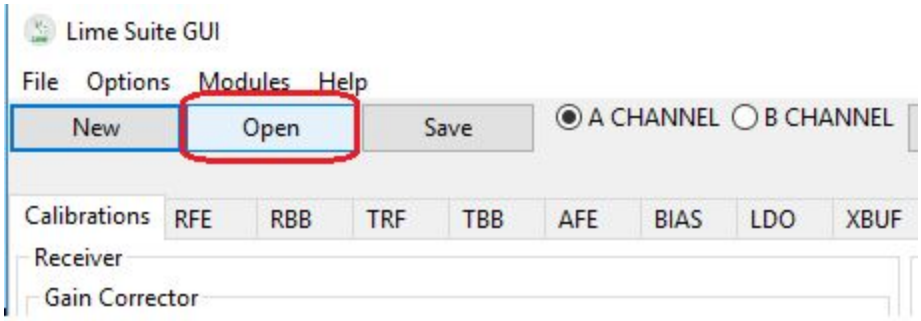


3. Select the device and click the “Connect” button



1.2 Load the configuration file

1. In LimeSuite GUI click the ‘Open’ button



2. Navigate and select self test INI file ‘self_test.ini’
3. Lime Suite GUI should be updated with values loaded from INI file.
4. Select GUI->Chip

1.3 Configure SXT

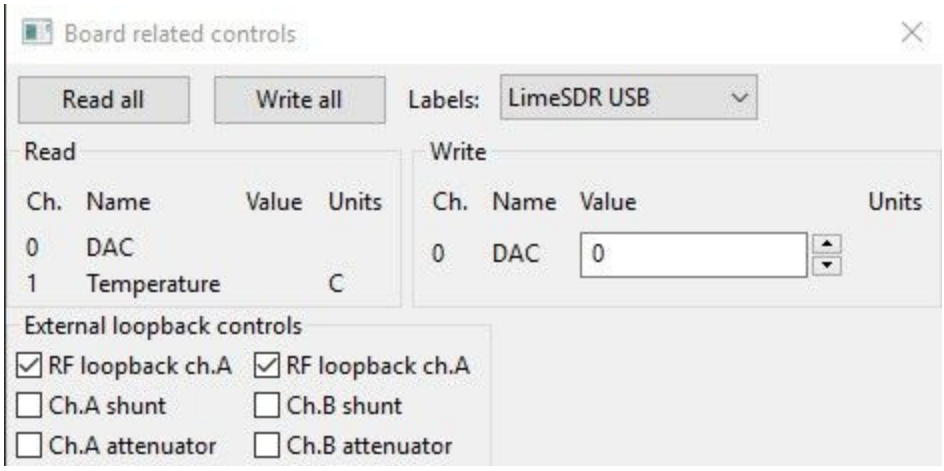
1. Go to the “SXT” tab.
2. Press the ‘Calculate’ button.
3. Press the ‘Tune’ button.

1.4 Configure CLKGEN

1. Go to the “CLKGEN” tab.
2. Press the ‘Calculate’ button.
3. Press the ‘Tune’ button.

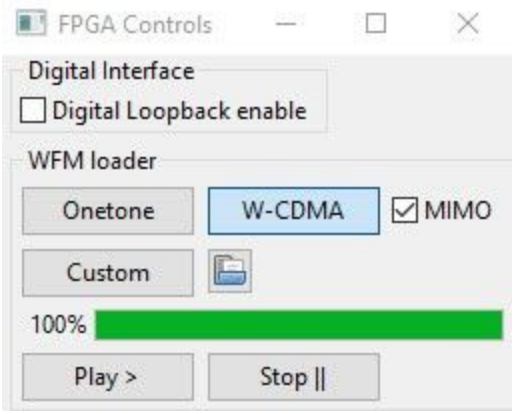
1.5 Enable RF loopback

1. From the menu bar select: ‘Modules->Board controls’
2. Tick RF loopback Ch.A and RF loopback Ch.B



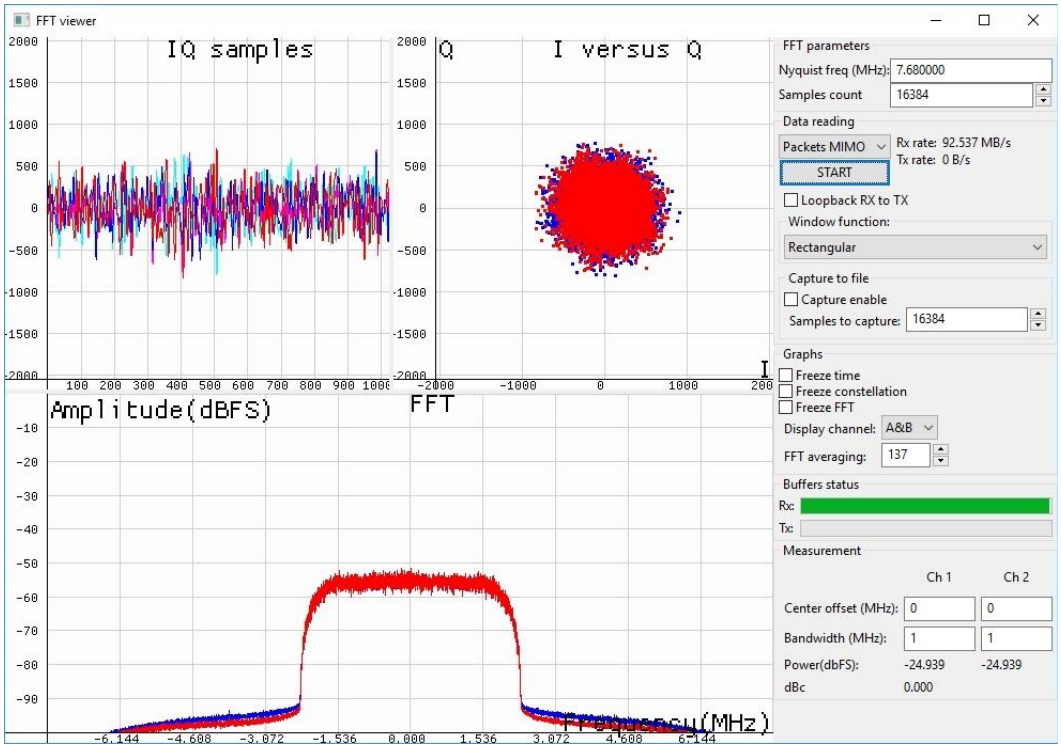
1.6 Load WCDMA waveform

1. From the menu bar select: ‘Modules->FPGA controls’.
2. Tick ‘MIMO ‘.
3. Click ‘W-CMDA ‘.



1.7 Viewing the signal using FFT viewer

1. From the menu bar select: ‘Modules->FFTviewer’.
2. Select ’Data reading’ -> ’LMS MIMO’.
3. Select ’Graphs'->’Display channel’->’A&B’.
4. Click “Start” button to start receiving samples.



2. RECEIVING A SIGNAL

2.1 Connect to board

1. Launch the Lime Suite GUI as before.
2. From menu bar select: ‘Options->ConnectionSettings’
3. Select the device and click the “Connect” button

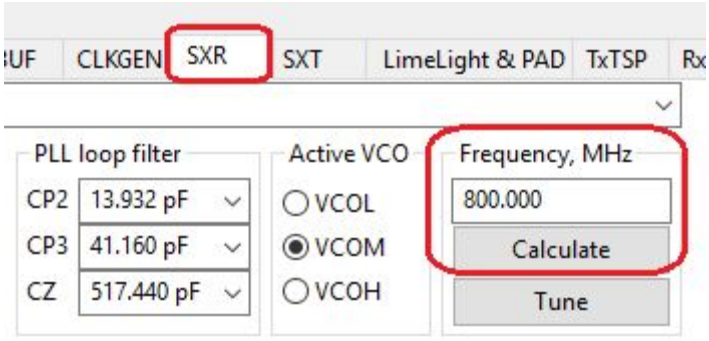
2.2 Load the configuration file

1. In LimeSuite GUI click the ‘Open’ button.
2. Navigate and select example INI file ‘example.ini’
3. Lime Suite GUI should be updated with values loaded from INI file.

2.3 Change the carrier frequency

In ‘example.ini’ the receiver frequency is set to 800 MHz. To change the receiver frequency:

1. Go to the “SXR” tab in Lime Suite GUI
2. Enter desired RX frequency in the field labelled “Frequency, MHz”.
3. Press the ‘Calculate’ button.



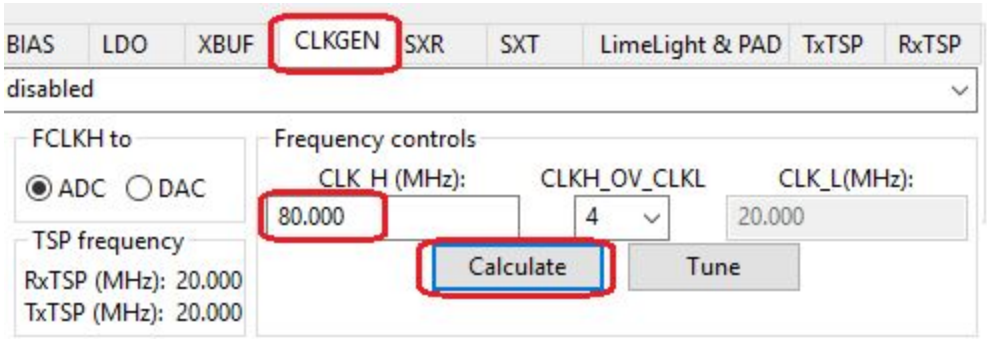
NOTE: The minimum allowed frequency is 30 MHz.

2.4 Change the sampling rate

The sampling rate set in the ‘example.ini’ configuration file is 10 MHz.

The simplest way to change the sampling rate without changing any dividers:

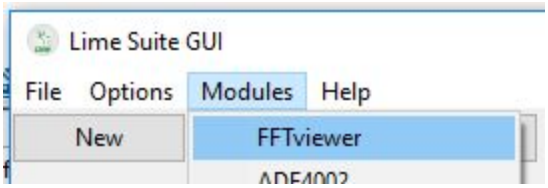
1. Go to the “CLKGEN” tab.
2. Adjust the “CLK_H (MHz)” value so that it is 8 times the desired sample rate. E.g. 80 MHz CLK_H will result in 10 MHz sample rate (10 MHz RF bandwidth).
3. Click the “Calculate” button.



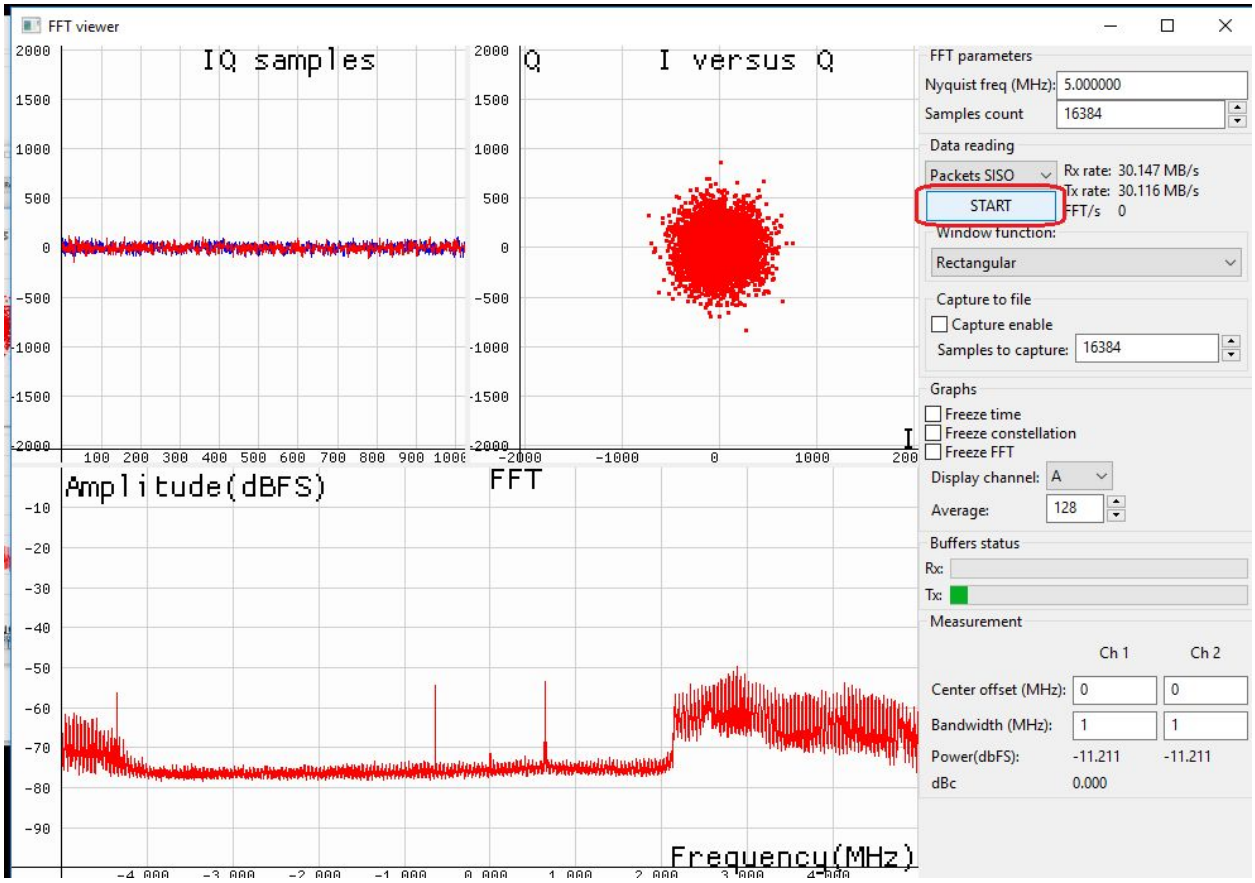
2.5 Viewing the signal using FFT viewer

Once the receiver frequency and sampling rate is configured, the RF signal can be observed using FFT viewer. In the ‘example.ini’ file the receiver is configured to use ‘RX1_L’ input and so the antenna should be connected to this port.

1. From the menu bar select: ‘Modules->FFTviewer’

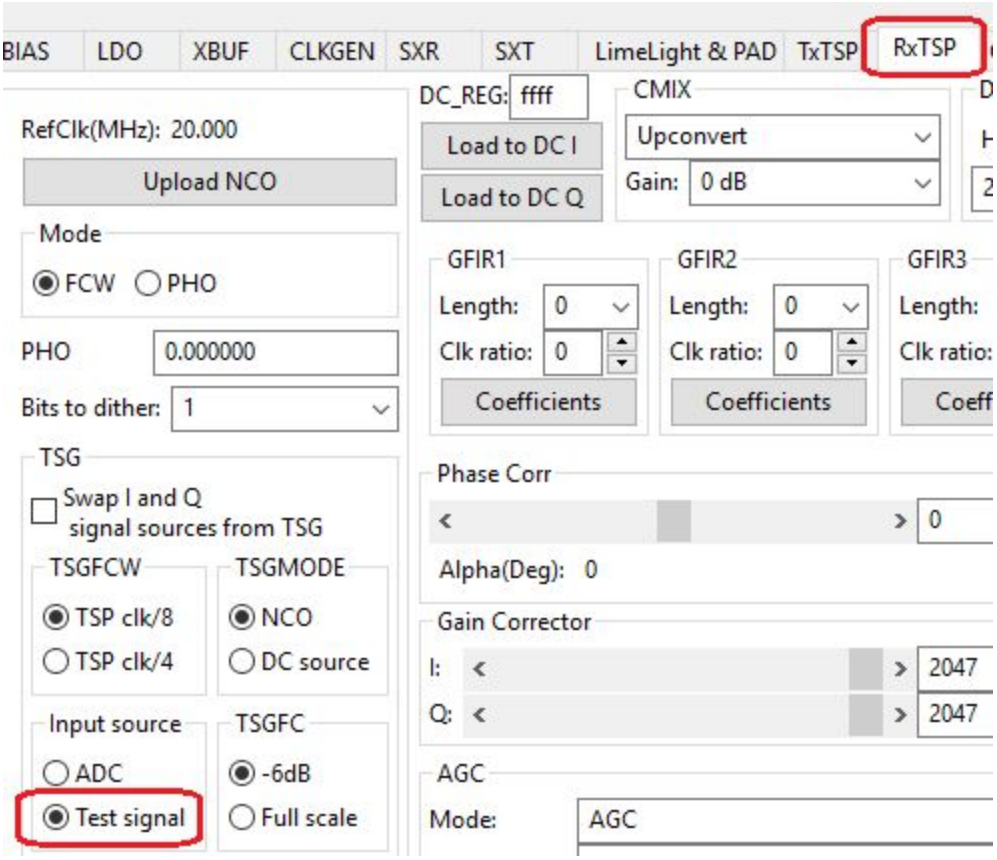


2. Click “Start” button to start receiving samples


FFT viewer
IQ samples
Q
I versus Q
I
FFT
Amplitude(dBFS)
Frequency(MHz)
FFT parameters
Nyquist freq (MHz): 5.000000
Samples count 16384
Data reading
Packets SISO
Rx rate: 30.147 MB/s
Tx rate: 30.116 MB/s
START
FFT/s 0
Window function:
Rectangular
Capture to file
Capture enable
Samples to capture: 16384
Graphs
Freeze time
Freeze constellation
Freeze FFT
Display channel: A
Average: 128
Buffers status
Rx:
Tx:
Measurement
Ch 1
Ch 2
Center offset (MHz): 0 0
Bandwidth (MHz): 1 1
Power(dbFS): -11.211 -11.211
dBc 0.000

2.6 Receiving test signal

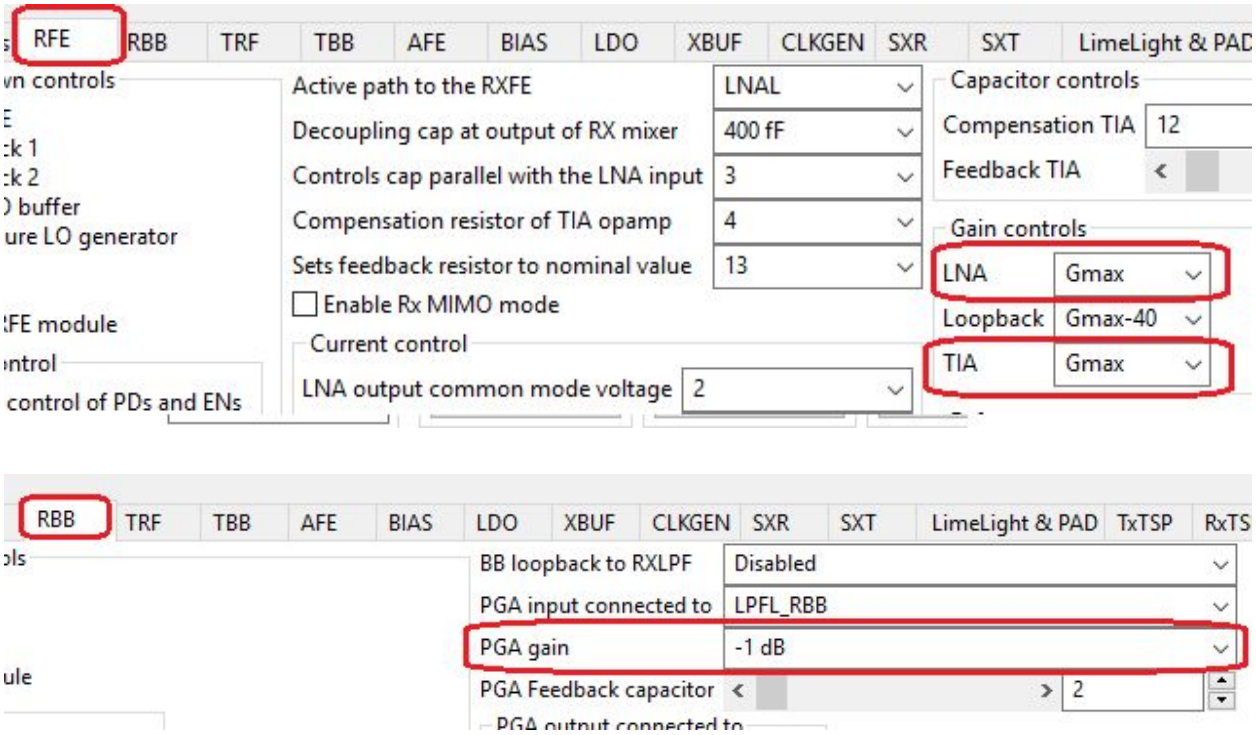
To enable test signal RX go to “RxTSP” tab and set input source to “Test signal”.



You can try playing around with test signal by changing TSGFCW, TSGMODE, TSGFC, CMIX values.

2.7 Changing RX gain

Rx gains can be adjusted in the “RFE” tab by changing ‘LNA’ and the ‘TIA’ values (Figure 9), and in the “RBB” tab by changing ‘PGA gain’ value.


RFE
RBB
TRF
TBB
AFE
BIAS
LDO
XBUF
CLKGEN
SXR
SXT
LimeLight & PAD
Active path to the RXFE
LNAL
Decoupling cap at output of RX mixer
400 fF
Controls cap parallel with the LNA input
3
Compensation resistor of TIA opamp
4
Sets feedback resistor to nominal value
13
Enable Rx MIMO mode
Current control
LNA output common mode voltage
2
Capacitor controls
Compensation TIA
12
Feedback TIA
Gain controls
LNA
Gmax
Loopback
Gmax-40
TIA
Gmax
RBB
TRF
TBB
AFE
BIAS
LDO
XBUF
CLKGEN
SXR
SXT
LimeLight & PAD
TxTSP
BB loopback to RXLPF
Disabled
PGA input connected to
LPFL_RBB
PGA gain
-1 dB
PGA Feedback capacitor
2

3. Transmitting a signal

3.1 Connect to the board

1. Launch the Lime Suite GUI as before.
2. From the menu bar select: ‘Options→ConnectionSettings’
3. Select the device to connect to and click the “Connect” button.

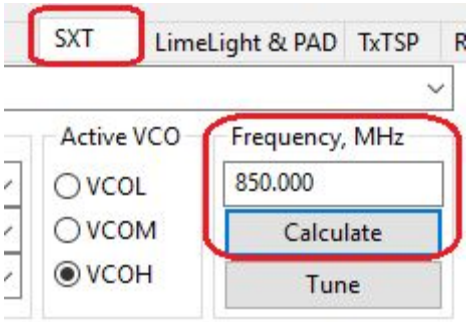
3.2 Load the configuration file

1. In Lime Suite GUI click the ‘Open’ button.
2. Navigate and select the example INI file ‘example.ini’
3. Lime Suite GUI should be updated with values loaded from the INI file.

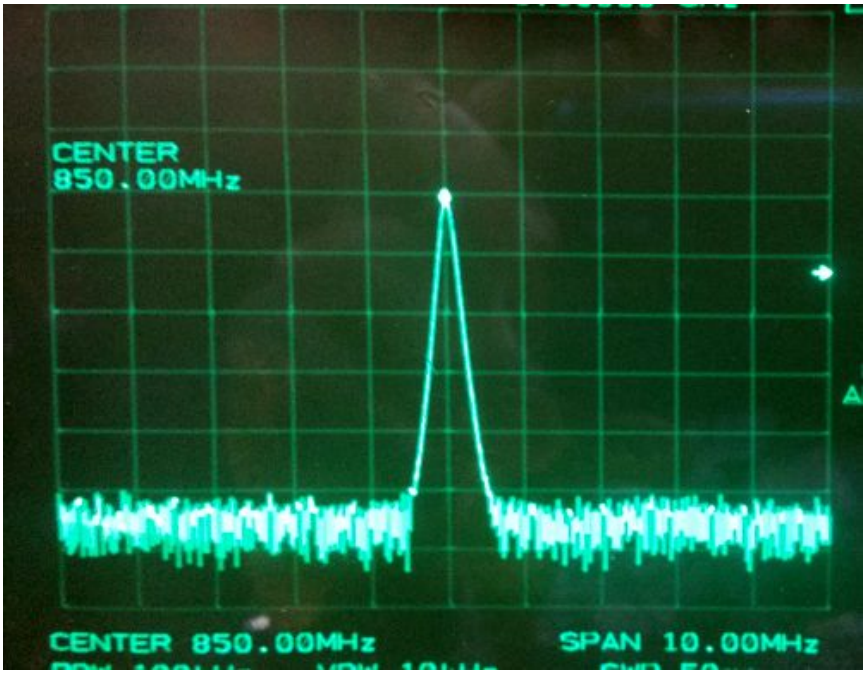
3.3 Change the carrier frequency

In ‘example.ini’ the transmitter frequency is set to 850 MHz. To change the transmitter frequency:

1. Go to the “SXT” tab in Lime Suite GUI
2. Enter the desired TX frequency in the field labelled “Frequency, MHz”. Note that the minimum allowed frequency is 30 MHz.
3. Press the ‘Calculate’ button.

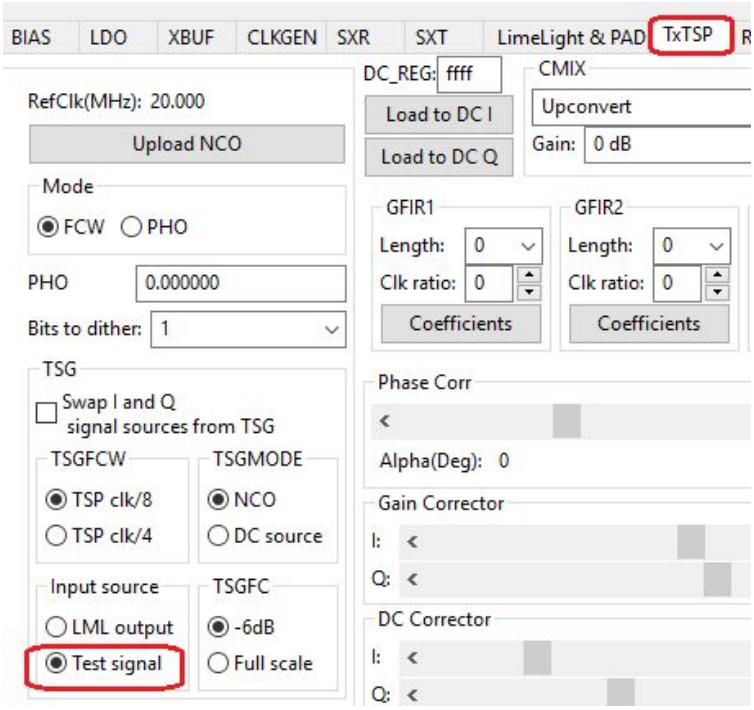


4. A spectrum analyser can be used to view the TX carrier signal

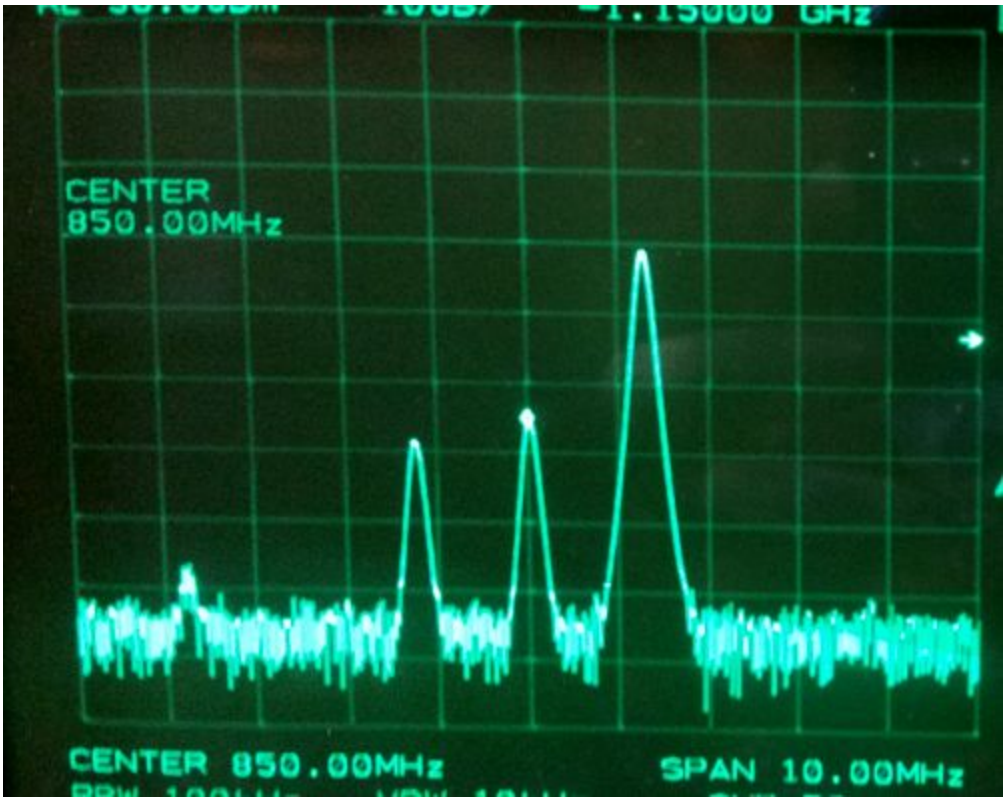


3.4 Transmitting test signal

To enable TX test signal go to the “TxTSP” tab and set the input source to “Test signal”.

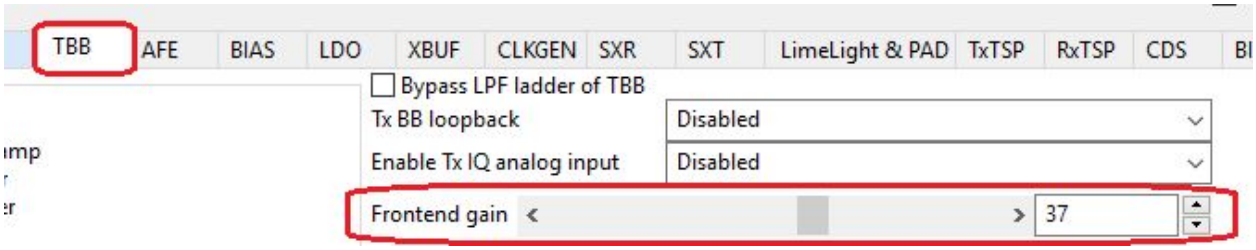


A spectrum analyzer can be used to view the TX test signal.



3.5 Changing TX gain

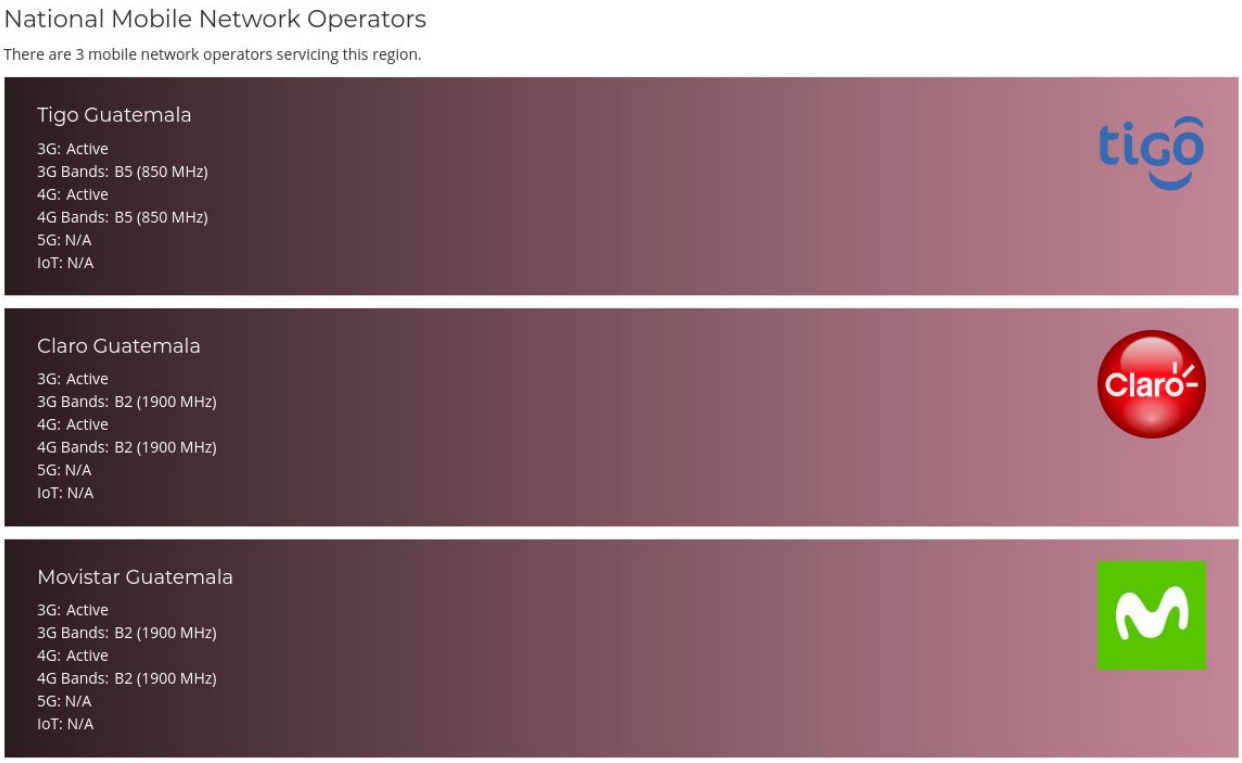
Tx gain can be adjusted in the “TBB” tab by changing the ‘Frontend gain’ value.



4. RECEIVING SIGNALS FROM AIR

Now that you have tested your LimeSDR, it will be interesting to see some nearby signals.

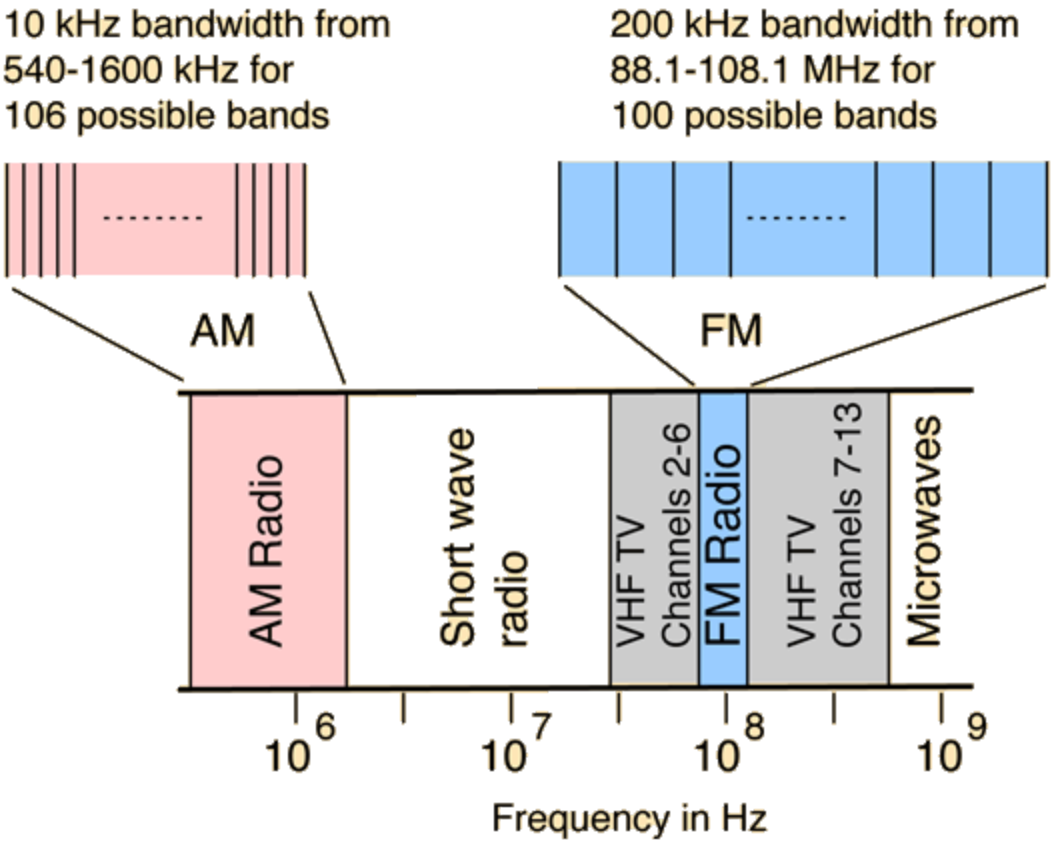
4.1 See what your local carriers are transmitting and add a graph below.



4.2 Look at your WiFi router and see if you can capture some signals at the 2.4GHz band and add a graph below.

Channel	F_0 (MHz)	Frequency Range (MHz)	North America[7]	Japan [7]	Most of world [7][8][9][10] [11][12][13][14]
1	2412	2401–2423	Yes	Yes	Yes
2	2417	2406–2428	Yes	Yes	Yes
3	2422	2411–2433	Yes	Yes	Yes
4	2427	2416–2438	Yes	Yes	Yes
5	2432	2421–2443	Yes	Yes	Yes
6	2437	2426–2448	Yes	Yes	Yes
7	2442	2431–2453	Yes	Yes	Yes
8	2447	2436–2458	Yes	Yes	Yes
9	2452	2441–2463	Yes	Yes	Yes
10	2457	2446–2468	Yes	Yes	Yes
11	2462	2451–2473	Yes	Yes	Yes
12	2467	2456–2478	No[B] except CAN	Yes	Yes
13	2472	2461–2483	No[B]	Yes	Yes
14	2484	2473–2495	No	**11b only**[C]	No

4.3 See if you can capture some AM and FM signals and add a graph below.



REFERENCES

[1] https://wiki.myriadrf.org/LimeSDR-USB_Quick_Test

[2] https://halberdbastion.com/intelligence/countries-nations/guatemala

[3] https://en.wikipedia.org/wiki/List_of_WLAN_channels