August 2021

Justin Montalbano

Kate Vajda

Wireless CyberTruck 2021

Introduction - Background

Justin Montalbano

- Reverse Engineering / Web / Networking / Mobile
- Automotive / Healthcare / Startups
- DefCON Car Hacking Village Lead

Kate Vajda

- Vulnerabilities / Detections / Reverse Engineering
- Industrial Control Systems / Utilities

Outline

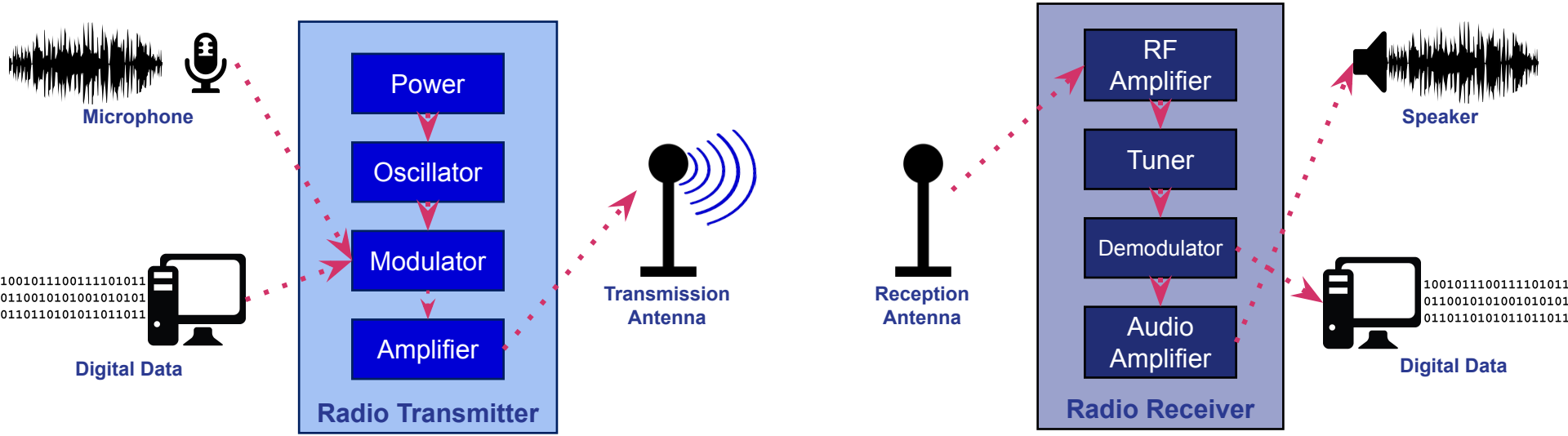
- Fundamentals of Wireless
- Wi-Fi (2.4 GHz / 5 GHz)
- Bluetooth (2.4 GHz)
- Cellular
- GPS
- Software Defined Radio (SDR)

Fundamentals of Wireless

- What is a Radio?
- What is a Radio Frequency (RF)?

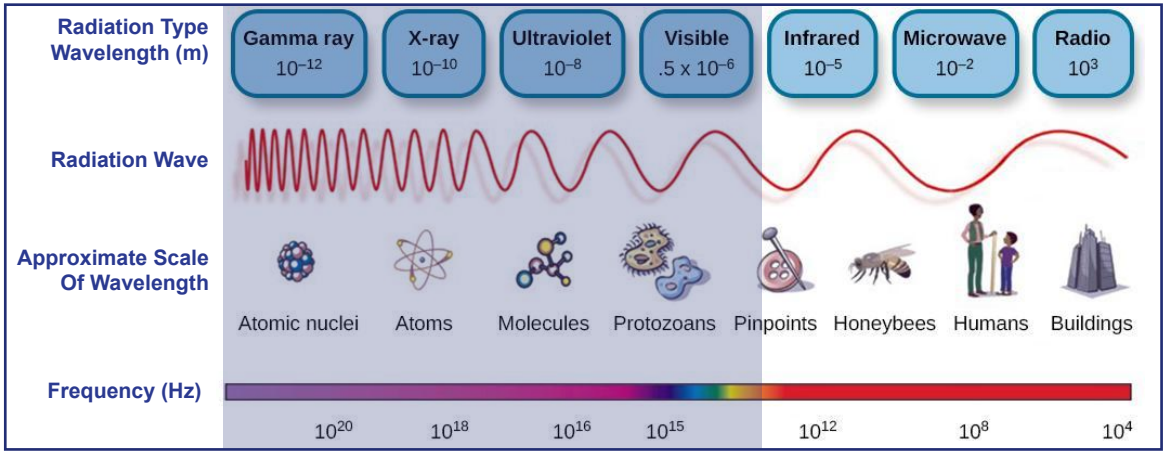
What is a Radio? – Components

- Antenna
- Transmitter
- Receiver
- Transceiver (combination of Receiver and Transmitter)

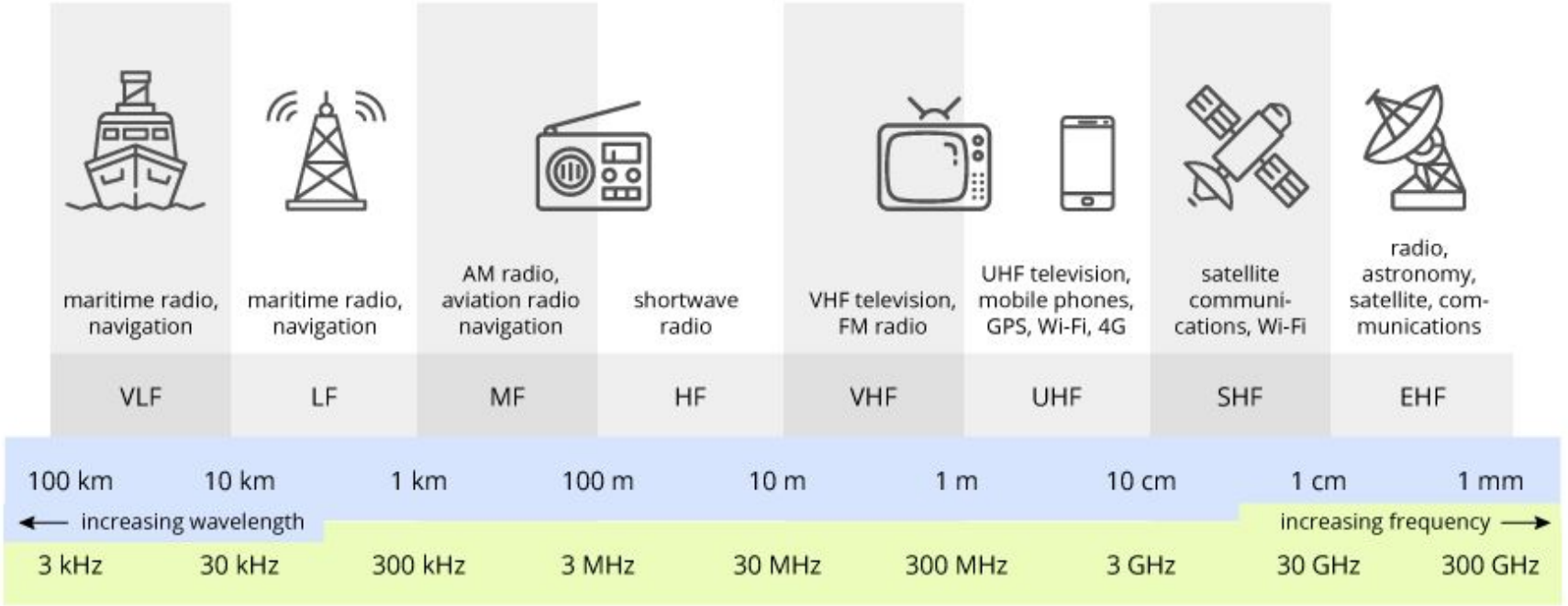


What is RF? – Electromagnetic Radiation

Radio Waves - A form of electromagnetic radiation with an identified frequency which range from 3 kHz to 300 GHz.



What is RF? – Applications



UNITED STATES FREQUENCY ALLOCATIONS

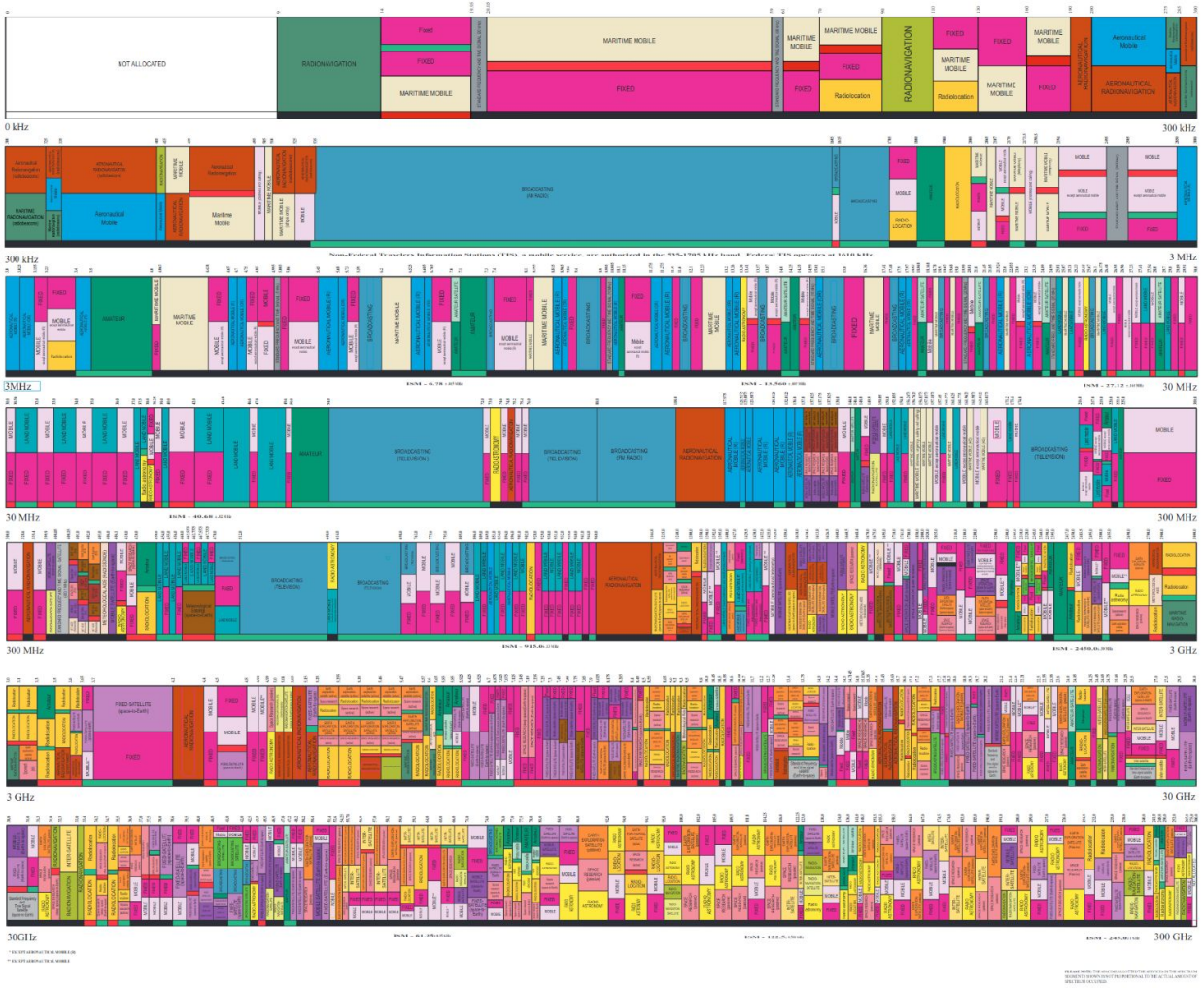
THE RADIO SPECTRUM

ALLOCATION USAGE DESIGNATION

SERVICE	EXAMPLE	DESCRIPTION
Primary	FIXED	Capital Letters
Secondary	Mobile	1st Capital with lower case letters

This chart is a graphic single-point-in-time portrayal of the Table of Frequency Allocations used by the FCC and NTIA. As such, it may not completely reflect all aspects, i.e. footnotes and recent changes made to the Table of Frequency Allocations. Therefore, for complete information, users should consult the Table to determine the current status of U.S. allocations.

U.S. DEPARTMENT OF COMMERCE
National Telecommunications and Information Administration
Office of Spectrum Management
JANUARY 2016



* EXCEPT AERONAUTICAL MOBILE (R)

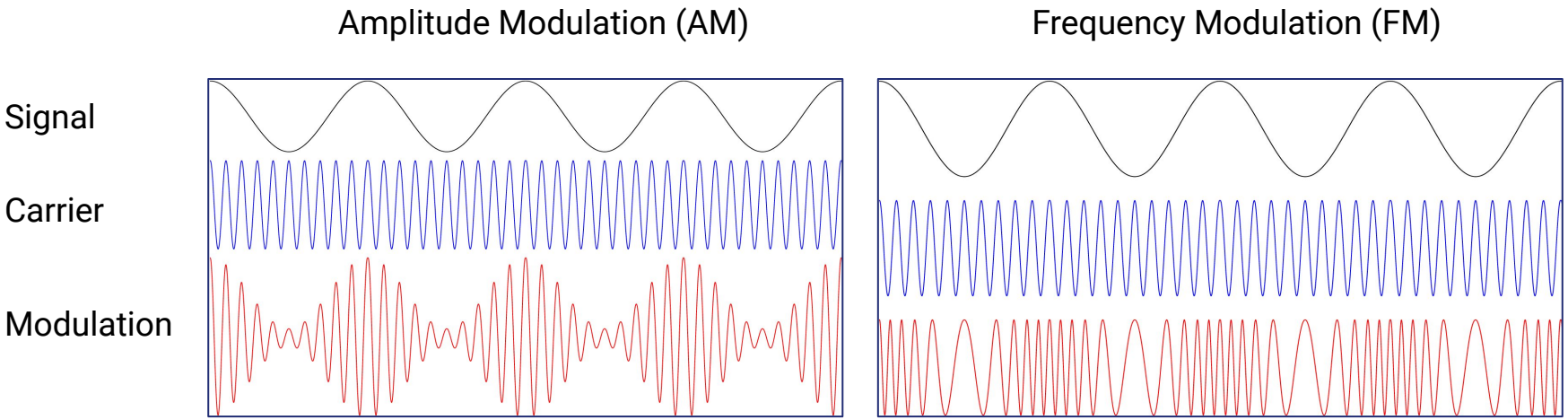
** EXCEPT AERONAUTICAL MOBILE

PLEASE NOTE: THE SPACING ALLOTTED FOR SERVICES IN THE SPECTRUM SEGMENTS SHOWN DOES NOT REFLECT PROPORTIONALLY THE ACTUAL AMOUNT OF SPECTRUM INCLUDED.

What is RF? – Radio Frequency Signal

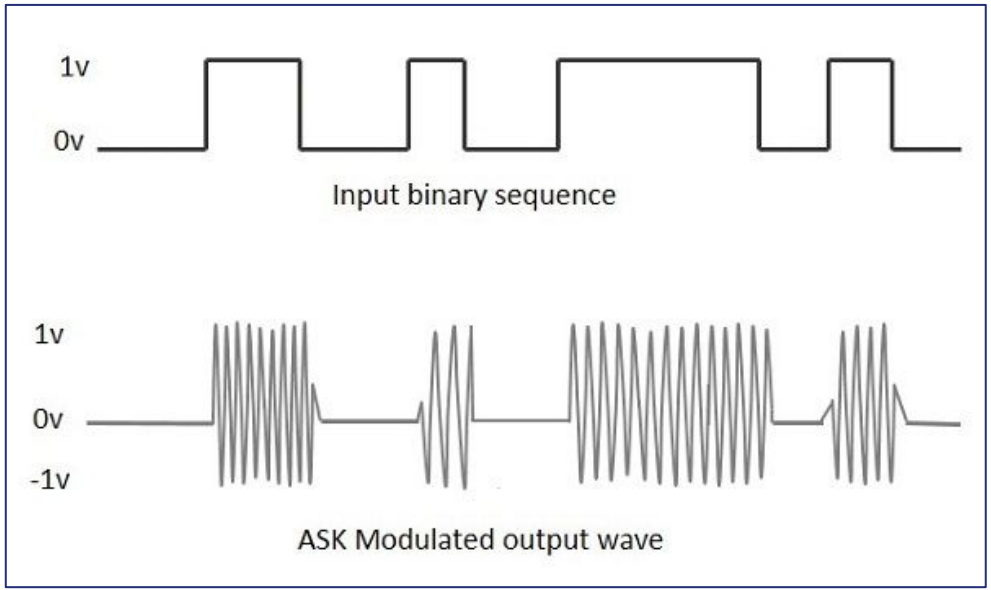
A wireless electromagnetic signal used as a form of telecommunication.

Modulation - The process of varying one or more properties of a periodic waveform, called the carrier signal, with a separate signal called the modulation signal



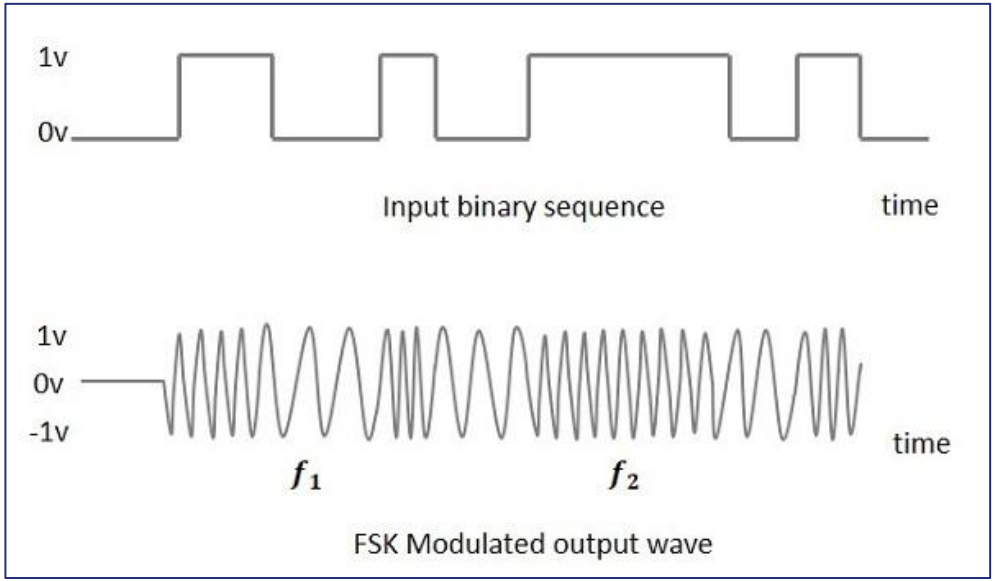
What is RF? – Amplitude Shift-Keying (ASK)

A form of amplitude modulation that represents digital data as variations in the amplitude of a carrier wave.



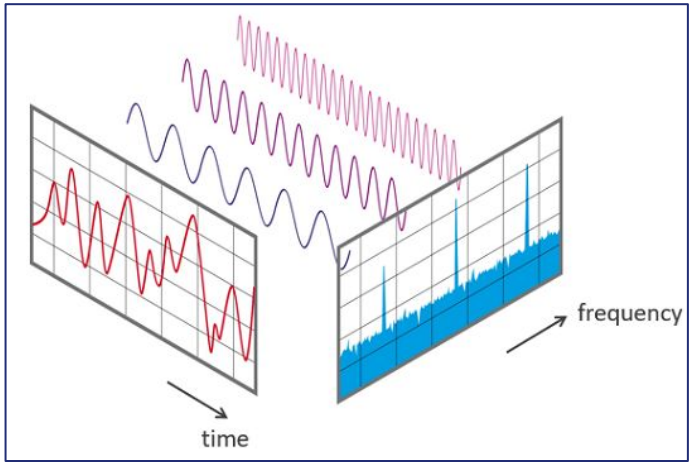
What is RF? – Frequency Shift-Keying (FSK)

A frequency modulation scheme in which digital information is transmitted through discrete frequency changes of a carrier signal.



What is RF? – Fast Fourier Transform (FFT)

An algorithm that samples a signal over a period of time (or space) and divides it into its frequency components



Fundamentals of Wireless

- FCCID Lookup

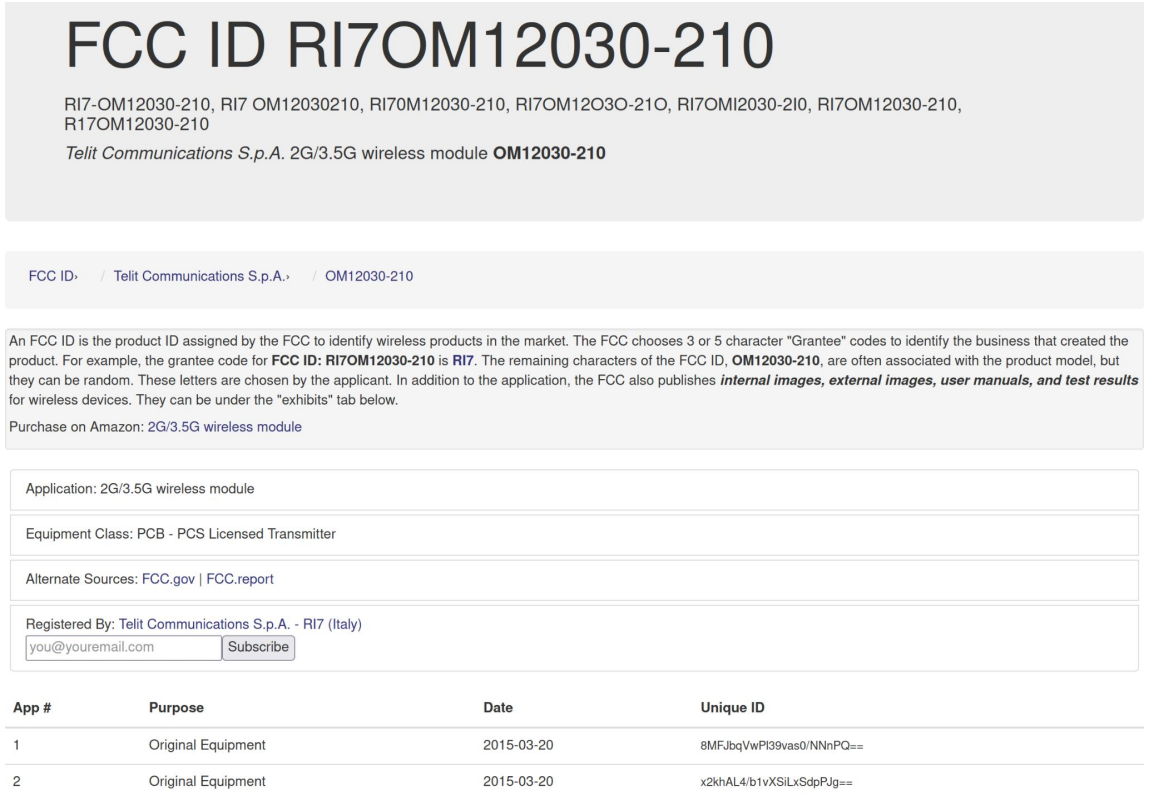
FCCID Lookup – Physical Inspection

What to look for? – FCC ID / IMEI #

FCCID Lookup – Research

Type FCC ID # into Google:

FCC ID RI7OM12030-210



FCC ID RI7OM12030-210

RI7-OM12030-210, RI7 OM12030210, RI70M12030-210, RI7OM12O3O-21O, RI7OMI2030-2I0, RI7OM12030-210, R17OM12030-210

Telit Communications S.p.A. 2G/3.5G wireless module **OM12030-210**

FCC ID› / Telit Communications S.p.A.› / OM12030-210

An FCC ID is the product ID assigned by the FCC to identify wireless products in the market. The FCC chooses 3 or 5 character "Grantee" codes to identify the business that created the product. For example, the grantee code for **FCC ID: RI7OM12030-210** is **RI7**. The remaining characters of the FCC ID, **OM12030-210**, are often associated with the product model, but they can be random. These letters are chosen by the applicant. In addition to the application, the FCC also publishes ***internal images, external images, user manuals, and test results*** for wireless devices. They can be under the "exhibits" tab below.

Purchase on Amazon: 2G/3.5G wireless module

Application: 2G/3.5G wireless module

Equipment Class: PCB - PCS Licensed Transmitter

Alternate Sources: FCC.gov | FCC.report

Registered By: Telit Communications S.p.A. - RI7 (Italy)

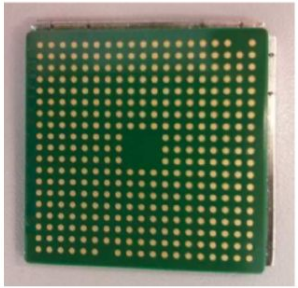
you@youremail.com Subscribe

App #	Purpose	Date	Unique ID
1	Original Equipment	2015-03-20	8MFJbqVwPl39vas0/NNnPQ==
2	Original Equipment	2015-03-20	x2khAL4/b1vXSiLxSdpPJg==

FCCID Look-up – Research

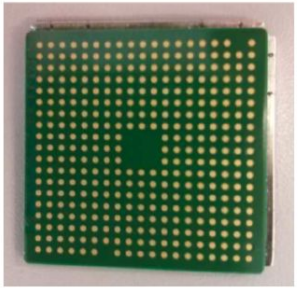
OM12030/210

External pictures



OM12030/210

Internal pictures



Operating Frequencies

Frequency Range	Power Output	Tolerance
824.2-848.8 MHz	344.3 mW	1ppm
826.4-846.6 MHz	199.9 mW	1ppm
826.4-846.6 MHz	124.2 mW	1ppm
1.7124-1.7526 GHz	89.5 mW	1ppm
1.7124-1.7526 GHz	98.2 mW	1ppm
1.8502-1.9098 GHz	535.8 mW	1ppm
1.8502-1.9098 GHz	209.9 mW	1ppm
1.8524-1.9076 GHz	184.9 mW	1ppm
1.8524-1.9076 GHz	114.8 mW	1ppm

Wi-Fi
(2.4 GHz / 5 GHz)

- Overview
- DEMO: Wi-Fi Pineapple Overview

Wi-Fi Overview

Rel. Year	1999	2007	2009	2013	2020	2023(?)
Freq. Band	2.4 GHz	2.4 GHz	2.4 + 5 GHz	5 GHz	2.4 + 5 + 6 GHz (6E)	2.4 + 5 + 6 GHz
Bandwidth	20 MHz	20 MHz	40 MHz	80 MHz, 160 MHz	80 MHz, 160 MHz	240 MHz, 320 MHz

Common Attacks:

- Deauth attacks
- Evil Twin Attack
- WPA2 MITM Attack
- WEP IV Attack

Wi-Fi Pineapple Overview

$100 device for hacking Wi-Fi

https://www.wifipineapple.com/

WiFi Pineapple NANO

The ultimate WiFi pentest companion, in your pocket.

- 6th generation WiFi Pineapple software featuring PineAP, web interface and modules
- Dual discrete 2.4 GHz b/g/n Atheros radios
- Up to 400 mW per radio with included antennas
- Integrated Power over USB Ethernet Plug
- Memory expansion via Micro SD (up to 200 GB)
- Optional mobile EDC Tactical case and battery
- USB 2.0 Host accessory expansion port

WiFi Pineapple TETRA

The amplified, dual-band (2.4/5 GHz) powerhouse.

- 6th generation WiFi Pineapple software featuring PineAP, web interface and modules
- Dual discrete 2.4/5 GHz a/b/g/n Atheros 2:2 MIMO radios
- 4 onboard Skybridge amplifiers
- Up to 800 mW per radio with included antennas
- Integrated Power over USB Ethernet Port
- Integrated Power over USB Serial Port
- Onboard NAND Flash (2 GB)
- USB 2.0 Host and RJ45 Ethernet Ports

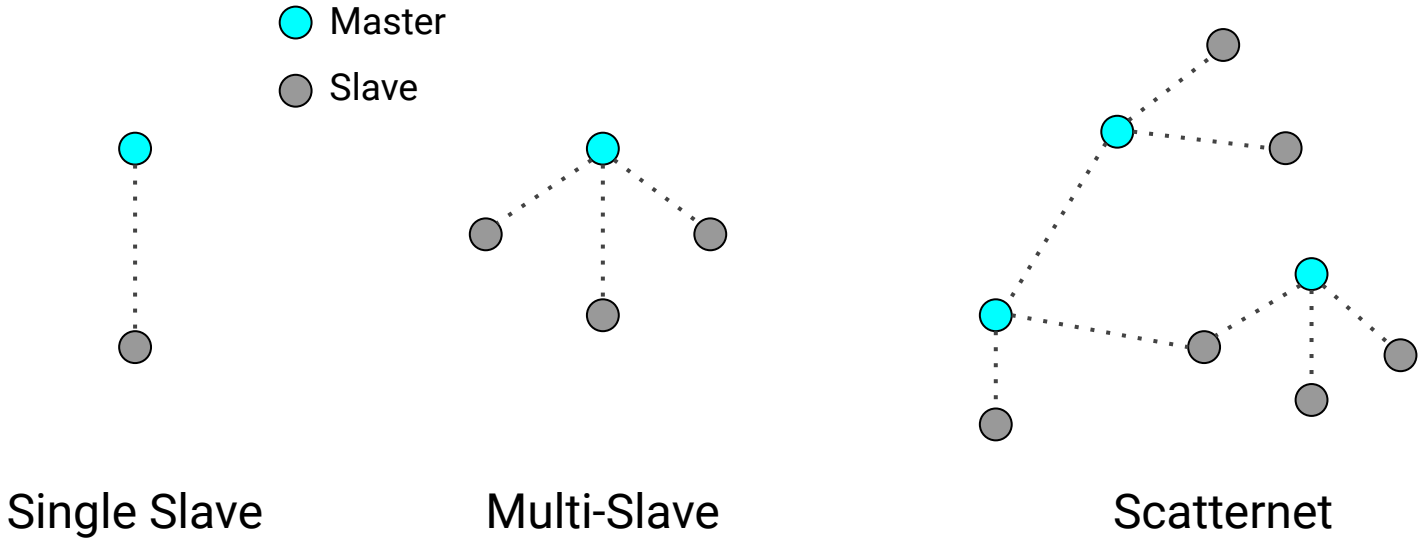
DEMO – Wi-Fi Pineapple Overview

Bluetooth (2.4 GHz - ISM)

- Overview
- DEMO:
  Gathering info on Bluetooth devices

Bluetooth – Personal Area Networks

- Master (central) scan for other devices, and initiate connection.
- Slave (peripheral) advertise and wait for connections.

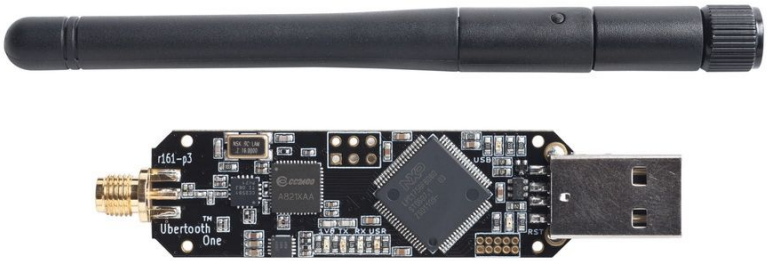


Bluetooth – Automotive

Information that can be saved on your car when you connect via Bluetooth:

- GPS history
- Device name.
- Address book.
- In-car internet search history.
- Music-streaming login, such as Spotify or Pandora
- Call log and text messages if you use hands-free calling
- WiFi identifiers

Bluetooth – Tools



Ubertooth One ($140)
https://greatscottgadgets.com/ubertoothone/

UD100 ($40)
http://www.senanetworks.com/ud100-g03.html

Bluetooth Demo

- Gathering info on Bluetooth devices

Step 1 - Install and run bettercap

```
git clone github.com/bettercap/bettercap.git

apt-get install golang libpcap-dev libusb-1.0-0-dev libnetfilter-queue-dev

make build

make install

./bettercap
```

Step 2 - Manual - help

```
sudo ./bettercap
bettercap v2.31.1 (built for linux amd64 with go1.13.8) [type 'help' for a list of commands]

192.168.168.0/24 > 192.168.168.84  » [23:42:56] [sys.log] [inf] gateway monitor started ...
192.168.168.0/24 > 192.168.168.84  » help

          help MODULE : List available commands or show module specific help if no module name is provided.
               active : Show information about active modules.
                 quit : Close the session and exit.
        sleep SECONDS : Sleep for the given amount of seconds.
             get NAME : Get the value of variable NAME, use * alone for all, or NAME* as a wildcard.
       set NAME VALUE : Set the VALUE of variable NAME.
  read VARIABLE PROMPT : Show a PROMPT to ask the user for input that will be saved inside VARIABLE.
                clear : Clear the screen.
       include CAPLET : Load and run this caplet in the current session.
            ! COMMAND : Execute a shell command and print its output.
       alias MAC NAME : Assign an alias to a given endpoint given its MAC address.

Modules

     any.proxy > not running
      api.rest > not running
     arp.spoof > not running
     ble.recon > not running
            c2 > not running
       caplets > not running
   dhcp6.spoof > not running
     dns.spoof > not running
  events.stream > running
           gps > not running
```

Step 3 - Listening service - ble.recon on

```
192.168.168.0/24 > 192.168.168.84  » ble.recon on
192.168.168.0/24 > 192.168.168.84  » [23:43:10] [ble.device.new] new BLE device detected as 5D:4F:DE:B4:CA:D4 (Apple, Inc.) -54 dBm.
192.168.168.0/24 > 192.168.168.84  » [23:43:10] [ble.device.new] new BLE device detected as 54:90:A1:EA:B7:1A (Apple, Inc.) -57 dBm.
192.168.168.0/24 > 192.168.168.84  » [23:43:11] [ble.device.new] new BLE device detected as 76:8B:7E:EE:6F:39 (Apple, Inc.) -65 dBm.
192.168.168.0/24 > 192.168.168.84  » [23:43:12] [ble.device.new] new BLE device detected as F7:EB:ED:0D:C1:4C -45 dBm.
192.168.168.0/24 > 192.168.168.84  » ble.show
```

RSSI ▲	MAC	Vendor	Flags	Connect	Seen
-42 dBm	f7:eb:ed:0d:c1:4c		LE + BR/EDR (controller)	✔	**23:43:27**
-53 dBm	54:90:a1:ea:b7:1a	Apple, Inc.	BR/EDR Not Supported	✔	**23:43:29**
-53 dBm	5d:4f:de:b4:ca:d4	Apple, Inc.	BR/EDR Not Supported	✔	**23:43:28**
-71 dBm	76:8b:7e:ee:6f:39	Apple, Inc.	LE + BR/EDR (controller), LE + BR/EDR (host)	✔	**23:43:29**

Step 4 - Enumerate bluetooth - ble.enum

```
192.168.168.0/24 > 192.168.168.84 » ble.enum 62:fc:a9:22:ba:41
[23:58:33] [sys.log] [inf] ble.recon connecting to 62:fc:a9:22:ba:41 ...
192.168.168.0/24 > 192.168.168.84 »
```

Handles	Service > Characteristics	Properties	Data
0001 -> 0005	Generic Access (1800)		
0003	Device Name (2a00)	READ	worktop
0005	Appearance (2a01)	READ	Generic Computer
0006 -> 0009	Generic Attribute (1801)		
0008	Service Changed (2a05)	READ, INDICATE	00000000
0010 -> 0014	Device Information (180a)		
0012	Manufacturer Name String (2a29)	READ	Apple Inc
0014	Model Number String (2a24)	READ	MacBookPro15,1
0020 -> 0023	Apple Continuity Service (d0611e78bbb44591a5f8487910ae4366)		
0022	8667556c9a374c9184ed54ee27d90049	**WRITE**, NOTIFY	
0024 -> 0027	9fa480e0496745429390d343dc5d04ae		
0026	af0badb15b9943cd917aa77bc549e3cc	**WRITE**, NOTIFY	

ble.write <mac> <uuid> <value>

```
192.168.168.0/24 > 192.168.168.84 » ble.enum 40:b6:1f:33:1f:86
[00:15:12] [sys.log] [inf] ble.recon connecting to 40:b6:1f:33:1f:86 ...
192.168.168.0/24 > 192.168.168.84 »
```

Handles	Service > Characteristics	Properties	Data
0001 -> 0005	Generic Access (1800)		
0003	Device Name (2a00)	READ	worktop
0005	Appearance (2a01)	READ	Generic Computer
0006 -> 0009	Generic Attribute (1801)		
0008	Service Changed (2a05)	READ, INDICATE	00000000
0010 -> 0014	Device Information (180a)		
0012	Manufacturer Name String (2a29)	READ	Apple Inc
0014	Model Number String (2a24)	READ	MacBookPro15,1
0020 -> 0023	Apple Continuity Service (d0611e78bbb44591a5f8487910ae4366)		
0022	8667556c9a374c9184ed54ee27d90049	**WRITE,** NOTIFY	
0024 -> 0027	9fa480e0496745429390d343dc5d04ae		
0026	af0badb15b9943cd917aa77bc549e3cc	**WRITE**, NOTIFY	

```
192.168.168.0/24 > 192.168.168.84 » ble.write 40:b6:1f:33:1f:86 8667556c9a374c9184ed54ee27d90049 ffffffffffffffff
[00:15:36] [sys.log] [inf] ble.recon connecting to 40:b6:1f:33:1f:86 ...
192.168.168.0/24 > 192.168.168.84 » [00:15:37] [sys.log] [err] ble.recon error while writing: insufficient authentication
192.168.168.0/24 > 192.168.168.84 »
```

Cellular

- Frequencies
- Tools

Cellular - Frequencies

2G Frequencies			
Frequency	800 MHz	850 MHz	1900 MHz
Band	SMR	CLR	PCS

3G Frequencies				
Frequency	850 MHz	1700 MHz	1900 MHz	2100 MHz
Band	CLR	AWS	PCS	AWS

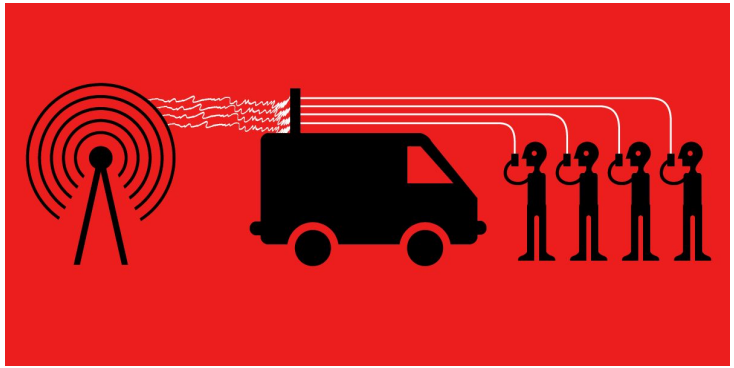
4G Frequencies												
Frequency	L700 MHz	L700 MHz	U700 MHz	800 MHz	850 MHz	1700/2100 MHz	1900 MHz	2300 MHz	2500 MHz	3500 MHz	5200 MHz	5700 MHz
Band	12,17	29	13	26	5	4,66	2,25	30	41	48	252	255

5G has a very large range of frequencies. Could not fit on this page. LINK

Cellular - Tools

- Cell site simulator
- SDR - BladeRF with YatesBTS
- SDR - Ettus Research USRP
- IMSI Catcher (StingRay)

Rohde & Schwarz Cell Site Simulator



Stingray (IMSI Catcher)

GPS

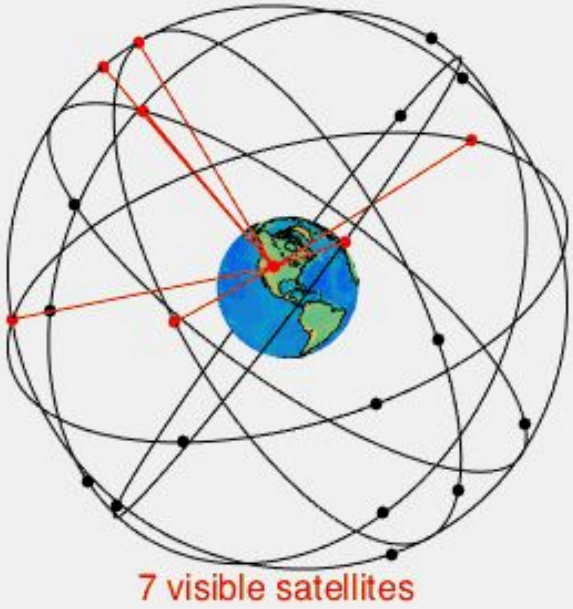
L1 Band (1575.42 MHz)
L2 Band (1227.6 MHz)
L5 Band (11476.45 MHz)

- Overview
- Spoofing
- DEMO: GPS Spoofing

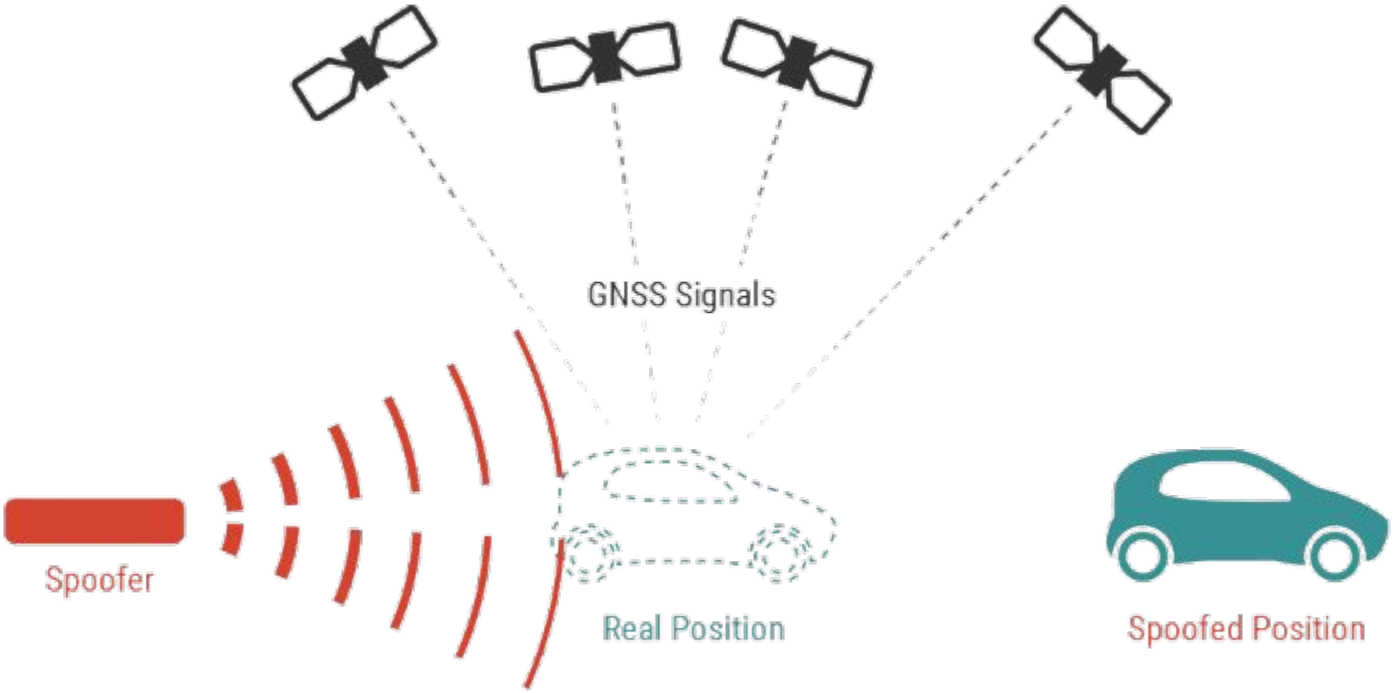
GPS - Overview

- Launch-1978 / Full Coverage-1994 / United States Government
- Multiple Bands / Multiple Frequencies
- Other Satellite Navigation Systems:
  - GLONASS (Launch-1982 / Full Coverage-1995 / Russian)
  - Galileo (Launch-2011 / Full Coverage-2021 / European Union)
  - BeiDou (Launch-2000 / Full Coverage-2020 / China)

Band	Frequency	Description
L1	1575.42 MHz	Used for civilian technologies (cellular, cars, trucks, etc.)
L2	1227.60 MHz	Used for civilian technologies (cellular, cars, trucks, etc.)
L3	1381.05 MHz	Used for nuclear detonation (NUDET) detection.
L4	1379.913 MHz	Being studied for additional ionospheric correction
L5	11476.45 MHz	Proposed for use as a civilian safety-of-life (SoL) signal.



GPS - Spoofing



GPS Demo

- Spoofing GPS Satellites

Step 1 - Install bladeRF

$ sudo apt-get install libusb-1.0-0-dev libusb-1.0-0 build-essential cmake libncurses5-dev libtecla1 libtecla1-dev pkg-config git wget

$ git clone https://github.com/Nuand/bladeRF.git ./bladeRF

$ cd ./bladeRF

$ cd host/

$ mkdir build

$ cd build

$ cmake ../

$ make && sudo make install && sudo ldconfig

Step 2 - Install gps-sdr-sim

$ git clone https://github.com/osqzss/gps-sdr-sim.git

$ gcc gpssim.c -lm -O3 -o gps-sdr-sim

$ cd gps-sdr-sim/

Step 3 - Download latest GNSS archive

Create login:

https://urs.earthdata.nasa.gov/users/new

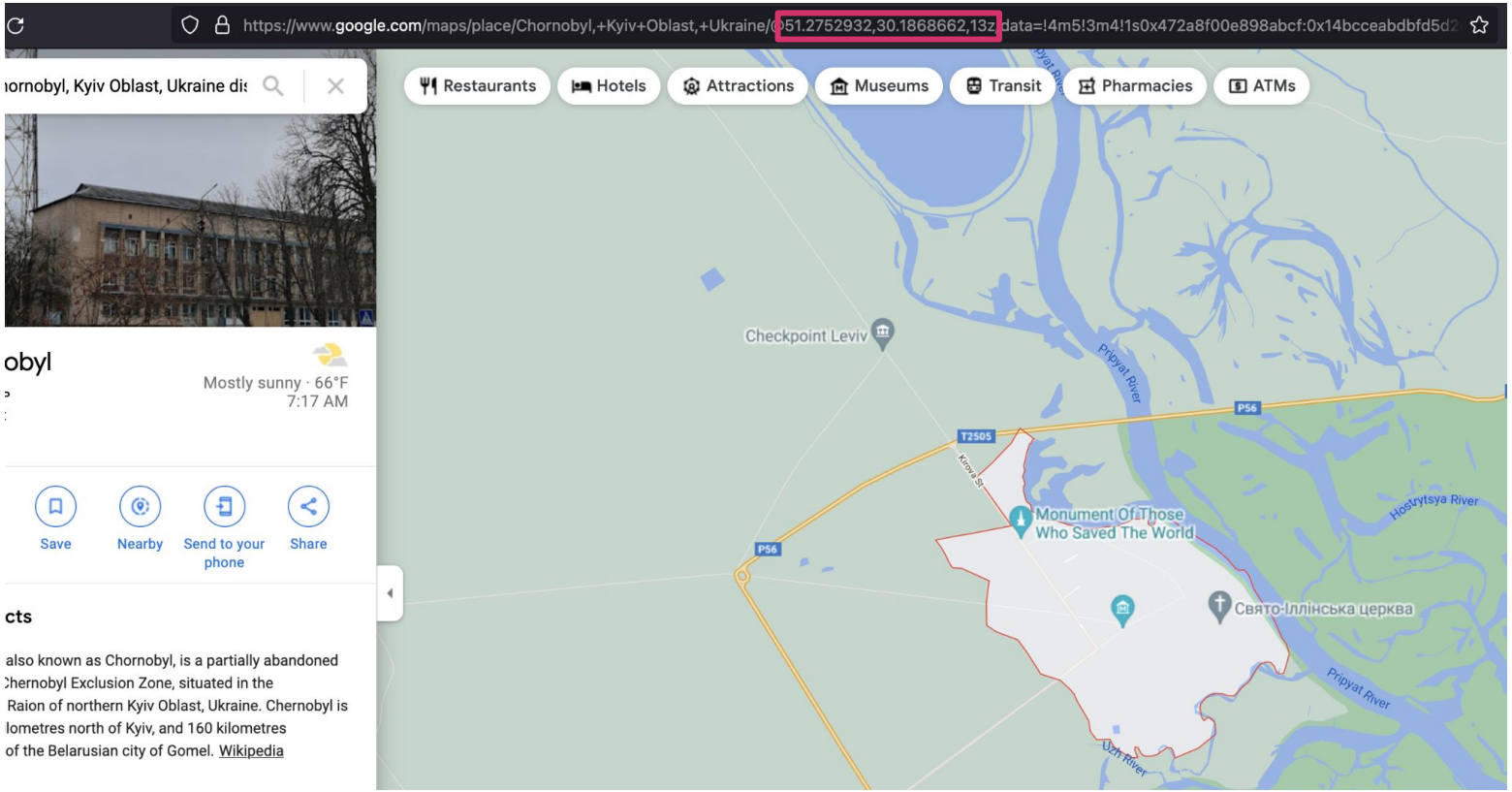
Download latest files:

https://cddis.nasa.gov/archive/gnss/data/daily/2021/brdc/

And copy to directory

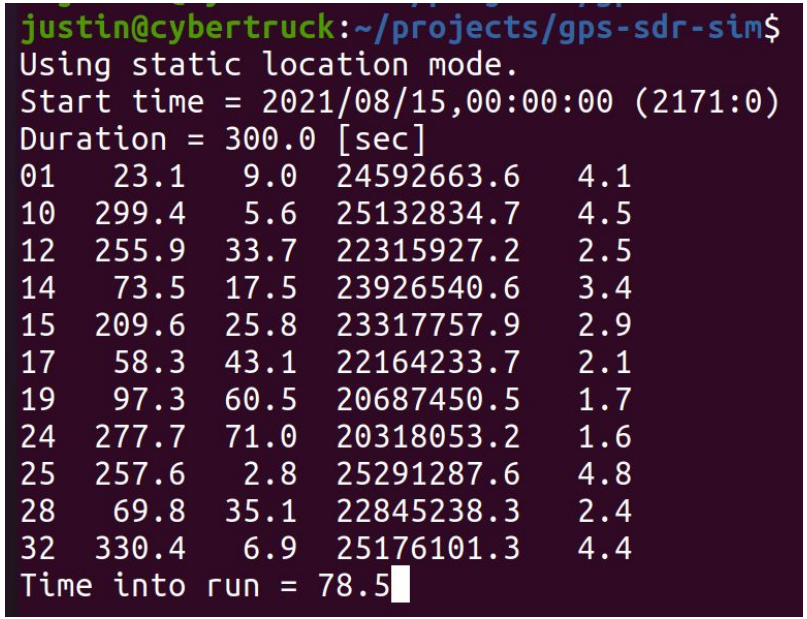
$ cp ~/Downloads/<latest> ~/gps-sdr-sim/

Step 4 - Select a spot



Step 4 - Generate constellations

./gps-sdr-sim -e brdc2270.21n -l 51.2752981,30.2131308,15z

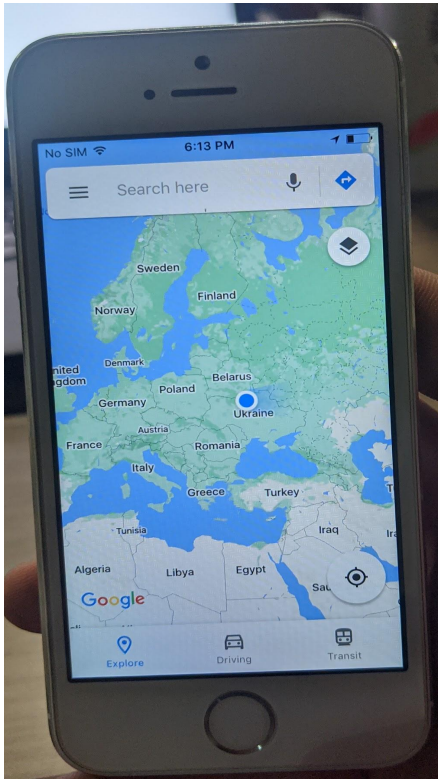


Step 4 - Spoof

$ bladeRF-cli -s bladerf.script

```
justin@cybertruck:~/projects/gps-sdr-sim$ bladeRF-cli -s bladerf.script

For best results, it is not recommended to set both RX and TX to the
same frequency. Instead, consider offsetting them by at least 1 MHz
and mixing digitally.

For the above reason, 'set frequency <value>` is deprecated and
scheduled for removal in future bladeRF-cli versions.

Please use 'set frequency rx' and 'set frequency tx' to configure
channels individually.

  RX1 Frequency: 1575420000 Hz (Range: [237500000, 3800000000])
  TX1 Frequency: 1575420000 Hz (Range: [237500000, 3800000000])


  Setting RX1 sample rate - req:   2600000 0/1Hz, actual:   2600000 0/1Hz
  Setting TX1 sample rate - req:   2600000 0/1Hz, actual:   2600000 0/1Hz


  RX1 Bandwidth:   2500000 Hz (Range: [1500000, 28000000])
  TX1 Bandwidth:   2500000 Hz (Range: [1500000, 28000000])


Setting TX1 txvga1 gain to -25 dB
           txvga1:  -25 dB (Range: [-35, -4])


  LPF tuning module: 23

  TX LPF I filter: 33
  TX LPF Q filter: 33

  RX LPF I filter: 30
  RX LPF Q filter: 30

  RX VGA2 DC reference module: 23
  RX VGA2 stage 1, I channel: 41
  RX VGA2 stage 1, Q channel: 41
  RX VGA2 stage 2, I channel: 27
  RX VGA2 stage 2, Q channel: 27


TX DC I: Value =  -272, Error =  0.414
TX DC Q: Value =   352, Error =  0.400
```

GPS - Spoofing Demo



Software Defined Radio (SDR)

- Overview
- Equipment
- GQRX Overview
- DEMO/EXERCISE:
  Find a Radio Station
- DEMO/EXERCISE:
  Find a Key Fob Signal
- GNU Radio Overview
- DEMO/EXERCISE:
  Record and Replay Key Fob Signal
- DEMO/EXERCISE:
  Decode Key Fob Signal

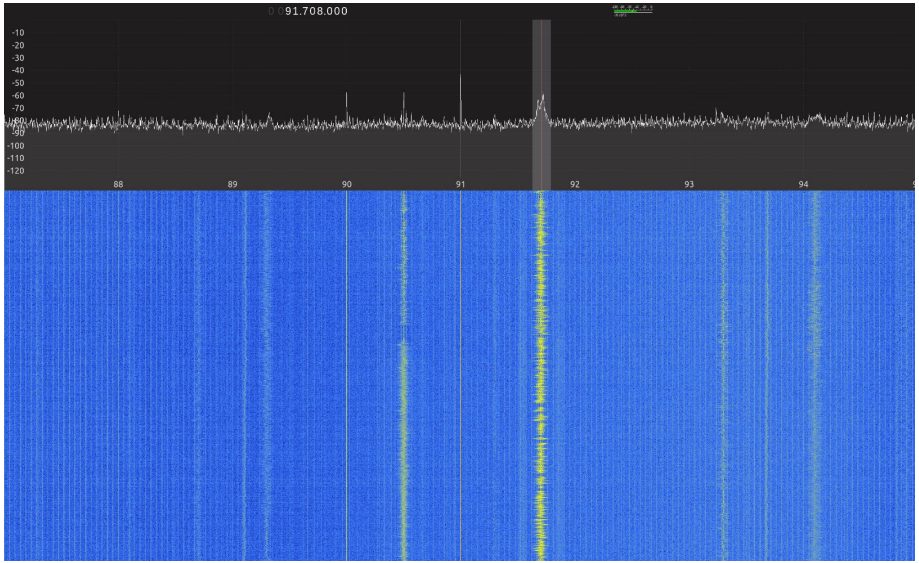
SDR - Overview

A radio system where components that have been traditionally implemented in hardware are instead implemented in software.

Extremely costly 10+ years ago

Defined by IEEE P1900.1

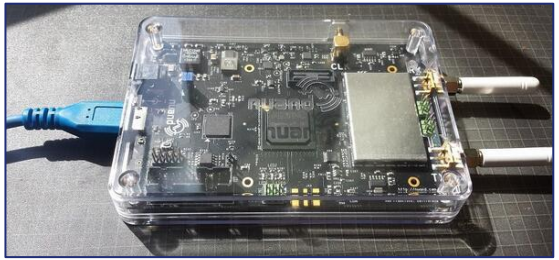
- “Radio in which some or all of the physical layer functions are software defined”



SDR - Equipment

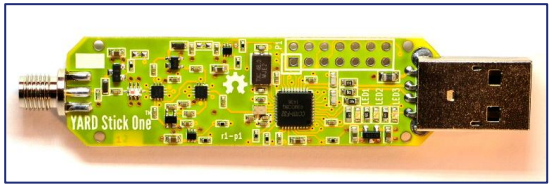
HackRF One ($300)
Half-duplex transceiver
1MHz to 6GHz / 20 MHz bandwidth
https://greatscottgadgets.com/hackrf/

Ettus USRP ($5,000+)
High-performance, scalable SDR
10MHz to 6GHz / 40-160 MHz bandwidth
ettus.com/product/category/USRP-X-Series



BladeRF ($420+)
Full-Duplex transceiver
300MHz to 3.8GHz / 50MHz+ bandwidth
https://www.nuand.com

RTL-SDR ($25)
DVB-T TV tuner based on RTL2832U
500KHz-1.75GHz / 5MHz bandwidth
https://www.rtl-sdr.com/



YARD Stick One ($120)
Half-duplex transceiver
300-348MHz / 391-464MHz / 782-928MHz
https://greatscottgadgets.com/yardstickone/

Proxmark ($300)
Read RFID / Spoof reader or tag
125KHz / 134KHz / 127.66KHz / 13.56MHz
hackerwarehouse.com/product/proxmark3-rdv2-kit/

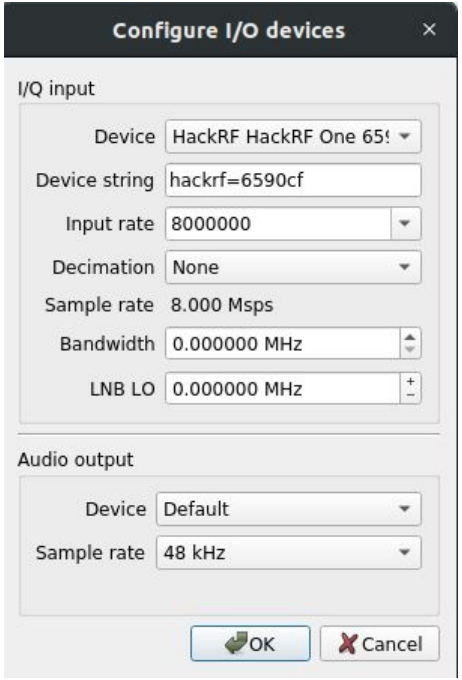
Software Defined Radio (SDR)

- GQRX Overview

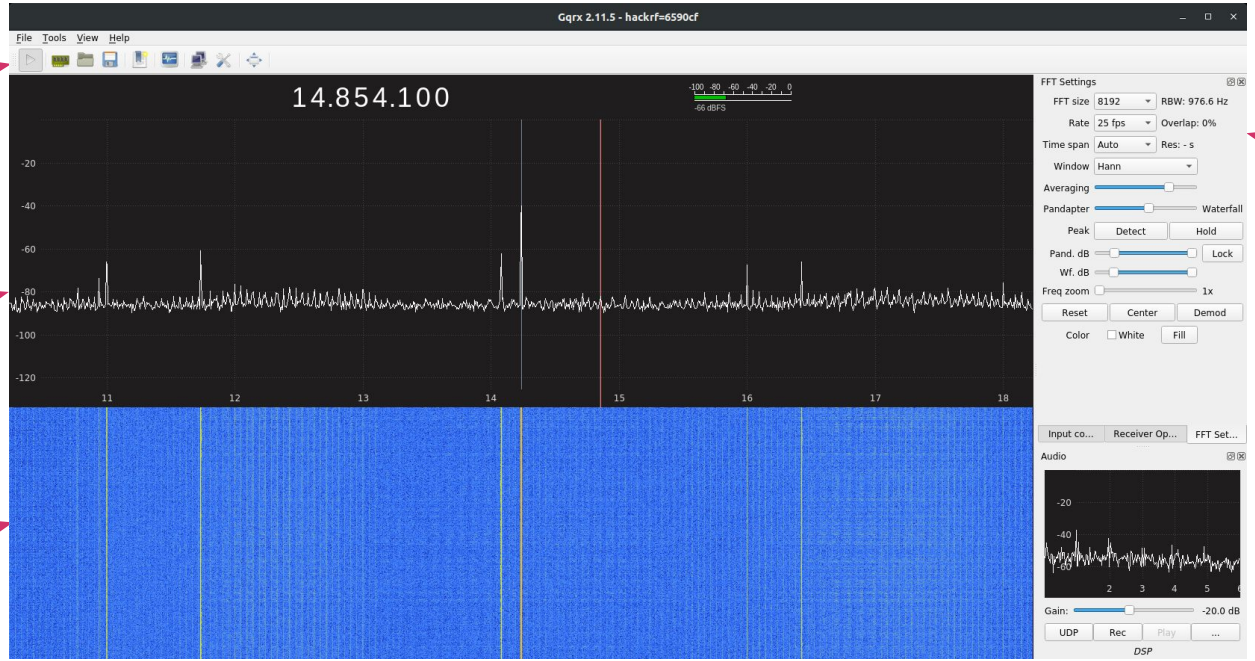
GQRX Overview - Configuration

Two important settings:

1. Device: hackrf=[model#]
2. Input Rate: 8000000 (8 Mega samples / second)



GQRX Overview – Main Screen



Start Button

Frequency View

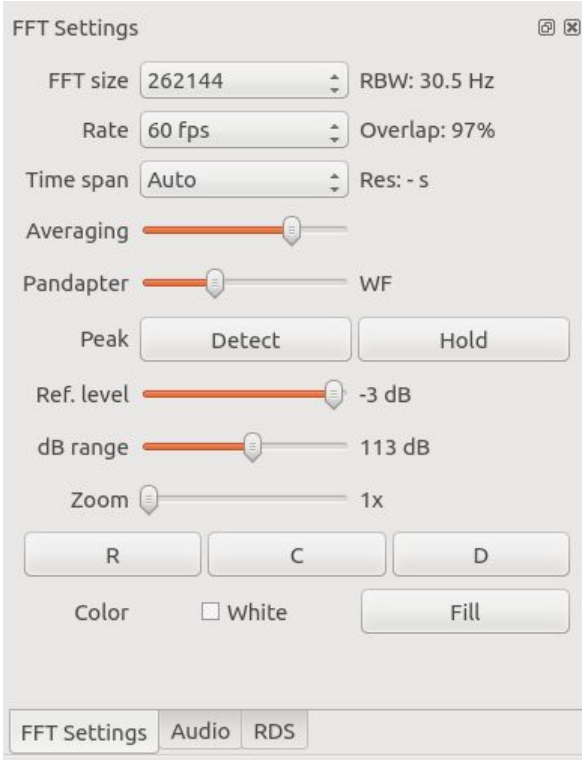
Waterfall View (Spectrogram)

Control Panel

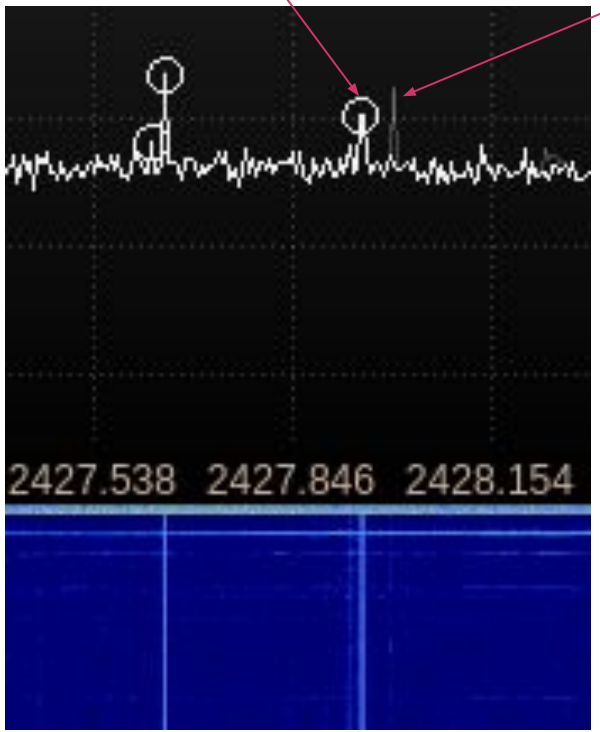
GQRX Overview – FFT Settings

Four common settings:

1. FFT Size – Sets resolution of waterfall and frequency view.
   Higher = Better
   Higher = More CPU
2. Peak Detect – Highlights and measures peak signals
3. Peak Hold – Keep outline of highest waves
4. Zoom – Zooms in on specified frequency



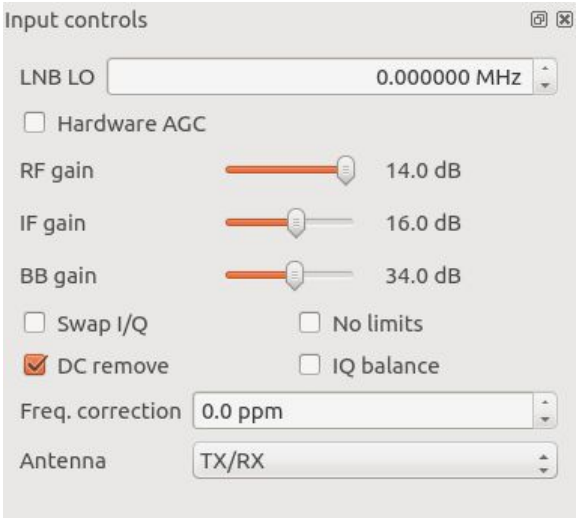
GQRX Overview – Peak Detect and Peak Hold



GQRX Overview – Input Settings (the HackRF)

Three common settings:

1. RF Gain – On or Off (14 dB is somewhat misleading)
   On = Better signals, but more noise
2. IF Gain and BB Gain
   Generally leave them around 16 dB or 24 dB
   Higher = louder signals, but much more noise
3. DC Remove - Remove annoying spike in the middle screen



GQRX Overview – Example Signal

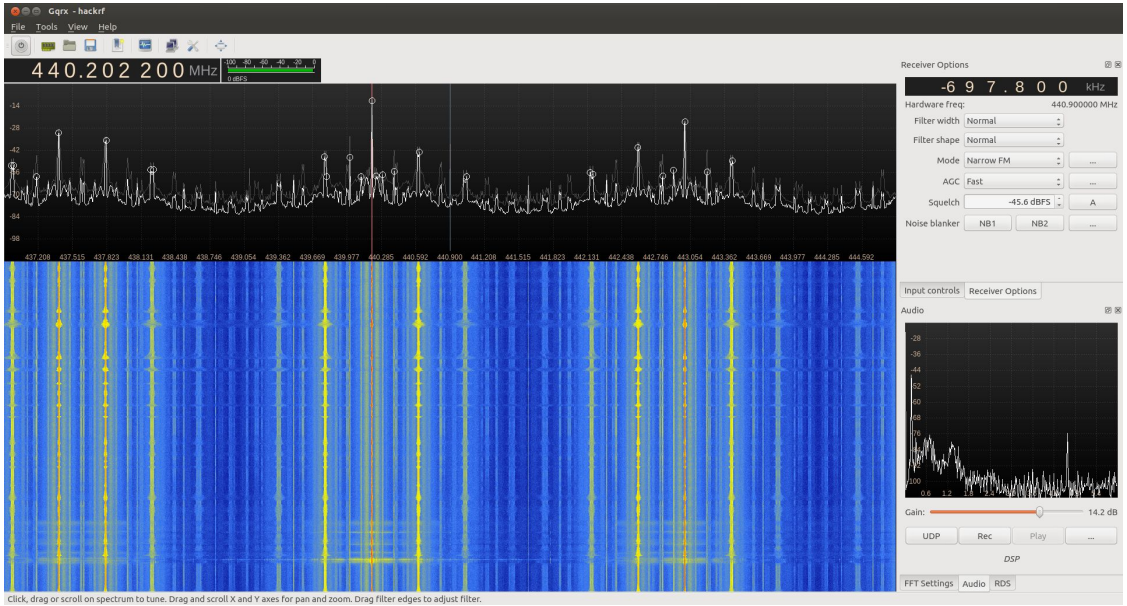
Signal = solid spike

This example is a handheld transceiver

Notice: the signal is so loud it has “harmonics”, signals repeated nearby

Note: if a signal is louder than 5 dB it can damage the HackRF

(not -5 dB)



Keep away from powerful RF sources
Towers, powerful radios, directional antennas, etc…
Turn RF gain down to compensate for loud signals

GQRX Overview – Demodulate Signal

Click on a signal to highlight it and play it over sound

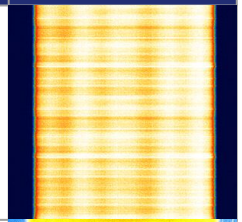
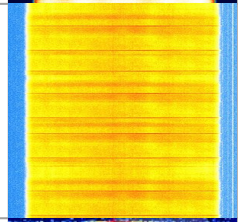
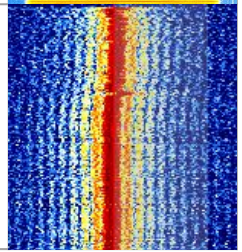
Receiver options control the demodulation

Important Settings:

1. Filter Width – Set the size of the signal
   Look this up, or guess
2. Mode - Raw IQ, AM, Narrow FM, Wide FM (WFM)...etc.
   Play with these settings to find the right sounding option
   Raw IQ is usually best to use when exporting to other programs
3. Squelch – Don't play static noise, only signals
   Select an area with no signal and click the "A" to automatically set

GQRX Overview – Other Signals

www.sigidwiki.com - a great source for active signals

Signal Type	Description	Frequency	Mode	Modulation	Bandwidth	Waterfall Image
2G CDMA (IS-95)	CDMA-One also known as IS-95, was the first ever cellular standard technology based off of CDMA. It is now defunct due to GSM and later classes of cellular techs replacing it.	850 MHz	AM	QPSK	1.228 MHz	
3G WCDMA	WCDMA, known primarily as 3G mobile, is a family of 3G data protocols used to send voice, text and signaling data to smart phones and other wireless devices.	824 MHz — 2,100 MHz	RAW, AM	QAM, QPSK, CDMA	4.2 MHz	
49MHz RC Car Controller	The sound of an RC controller signal from an old amphibious toy car	49.2 MHz	USB			

GQRX Overview – Other Resources

sigidwiki.com - Resource for signal identification

radioreference.com - Database of radio stations, repeaters, and communication frequencies

websdr.org - Tune into SDRs around the world, or broadcast yours to the world

w1hkj.com/FldigiHelp-3.21/Modes - Ham Radio Digital Signals

arrl.org/getting-licensed - Get licensed to broadcast around the world

rtl-sdr.com - Keep up to date with SDR news and experiments

cgran.org - Huge collection of advanced GNURadio blocks

SDR Demo

- Find a Radio Station / Key Fob Signal

Step 1 - Find a Radio Station / Key Fob

Plugin hackRF or other SDR

Ensure hackRF is connected

```
justin@cybertruck:~$ hackrf_info
hackrf_info version: unknown
libhackrf version: unknown (0.5)
Found HackRF
Index: 0
Serial number: 0000000000000000909864c8345517cf
Board ID Number: 2 (HackRF One)
Firmware Version: 2015.07.2 (API:1.00)
Part ID Number: 0xa000cb3c 0x00554757
```

Step 2 - Find a Radio Station / Key Fob

Start up GQRX

$ gqrx

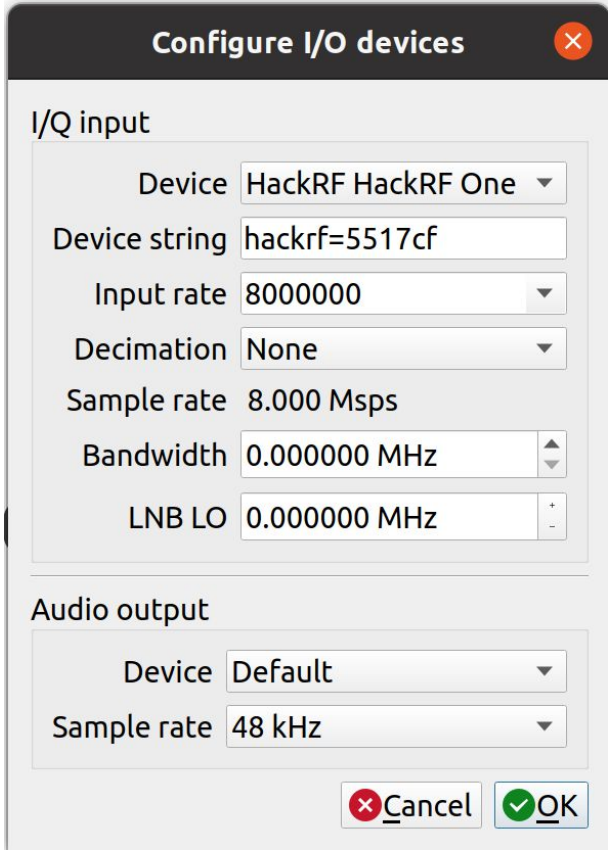
```
justin@cybertruck:~$ gqrx
Controlport disabled
No user supplied config file. Using "default.conf"
gr-osmosdr 0.2.0.0 (0.2.0) gnuradio 3.8.1.0
built-in source types: file osmosdr fcd rtl rtl_tcp uhd miri hackrf bladerf rfsp
ace airspy airspyhf soapy redpitaya freesrp
gr::log :WARN: file_source0 - file size is not a multiple of item size
FM demod gain: 3.05577
Resampling audio 96000 -> 48000
IQ DCR alpha: 1.04166e-05
Using audio backend: auto
BookmarksFile is /home/justin/.config/gqrx/bookmarks.csv
[INFO] [UHD] linux; GNU C++ version 9.2.1 20200304; Boost_107100; UHD_3.15.0.0-2
```

Step 3 - Find a Radio Station / Key Fob

Setup GQRX for hackRF

Set Device to hackrf

Set Input Rate to 8000000

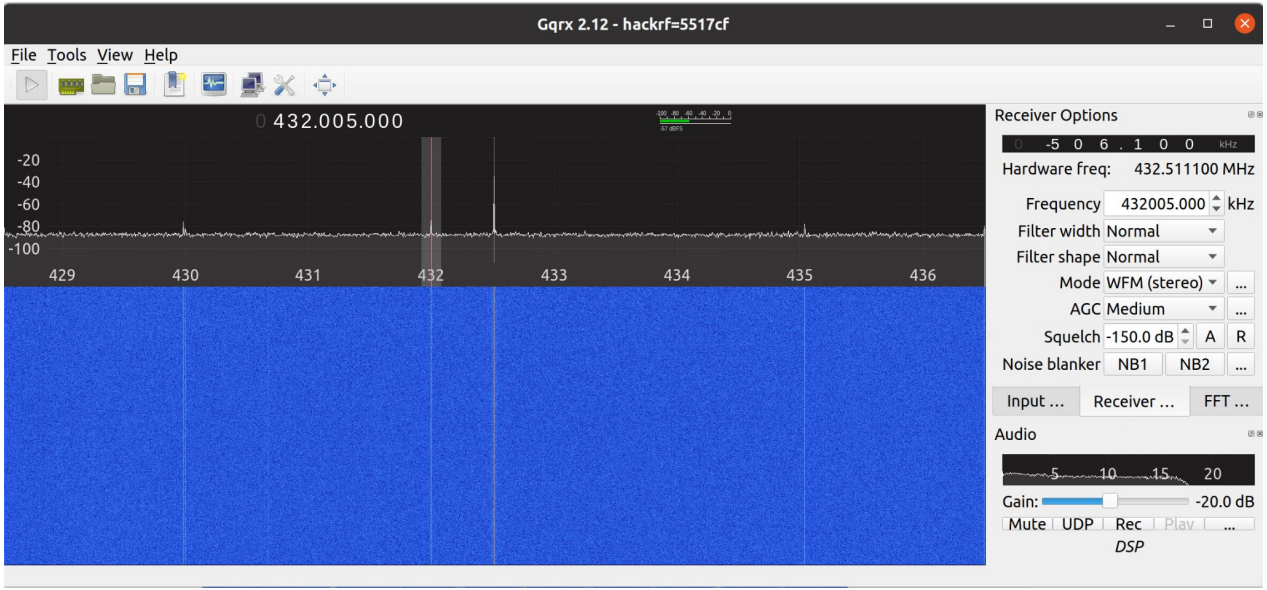


Step 4 - Find a Radio Station / Key Fob

Change Mode to WFM (stereo)

Tune to your favorite radio station

Click the ‘Play’ icon in the upper left hand corner

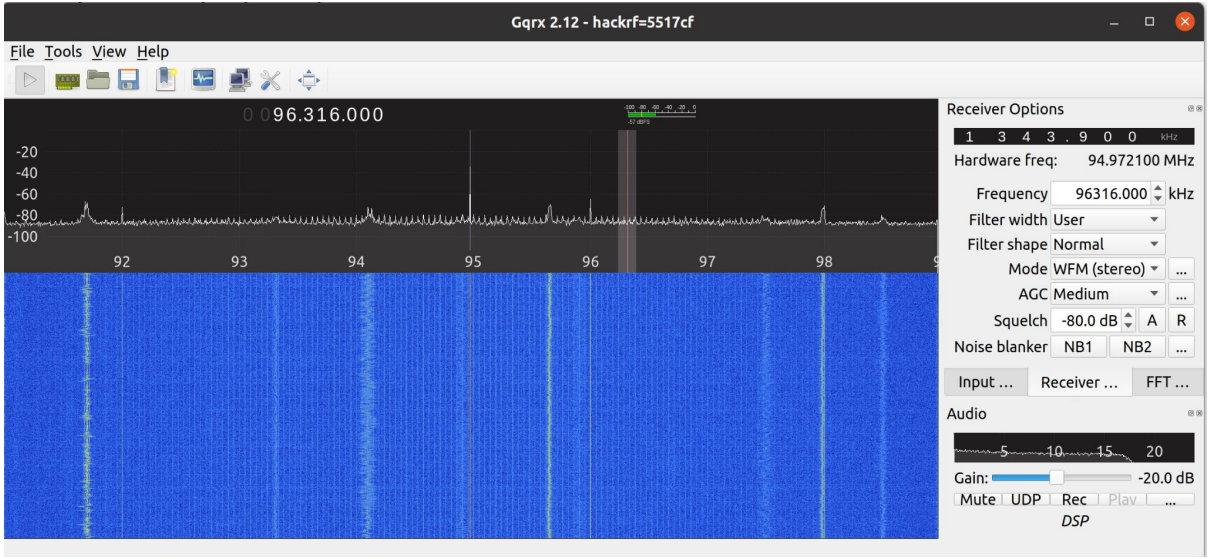


Step 5 - Find a Radio Station / Key Fob

Turn up the gain (volume)

Select an area next to frequency, click the ‘A’ button next to Squelch

Go back to frequency, Squelch should help remove some noise



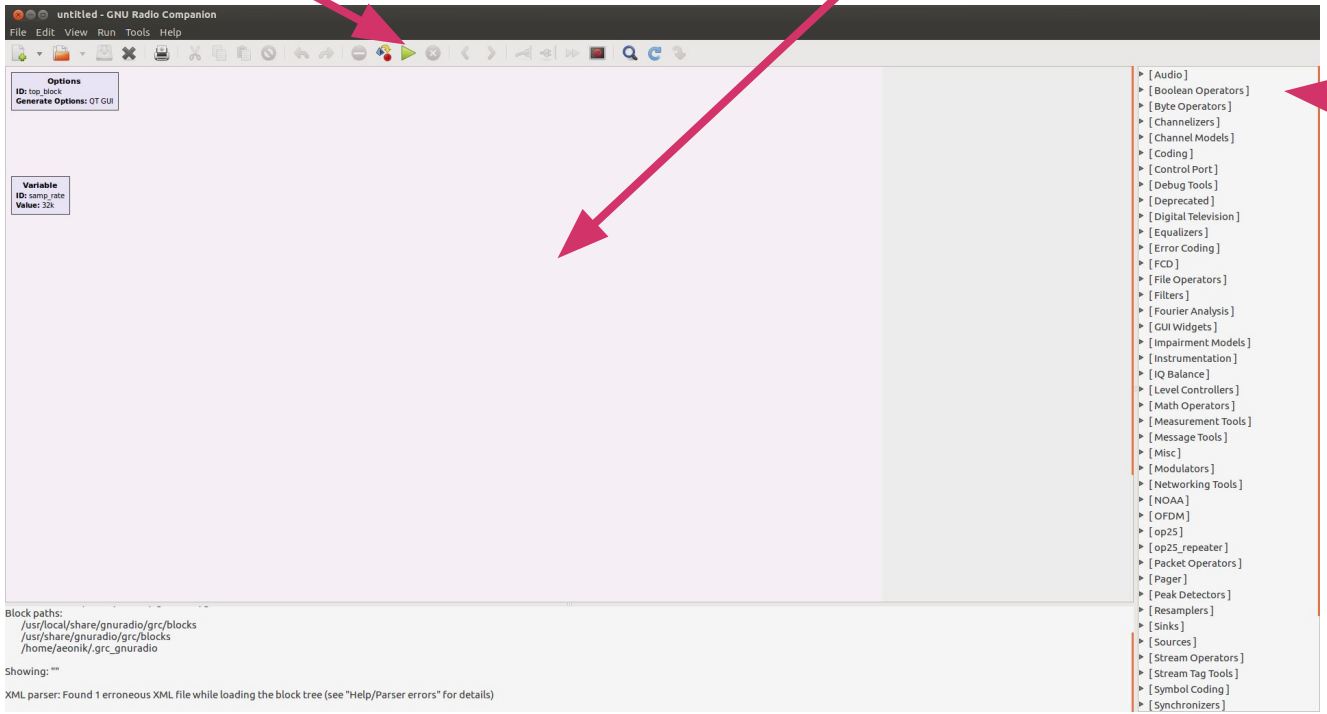
Software Defined Radio (SDR)

- GNU Radio Overview

GNU Radio – Starting Page

Play Flow Chart

Flow Chart Canvas



Function Blocks

Console Output

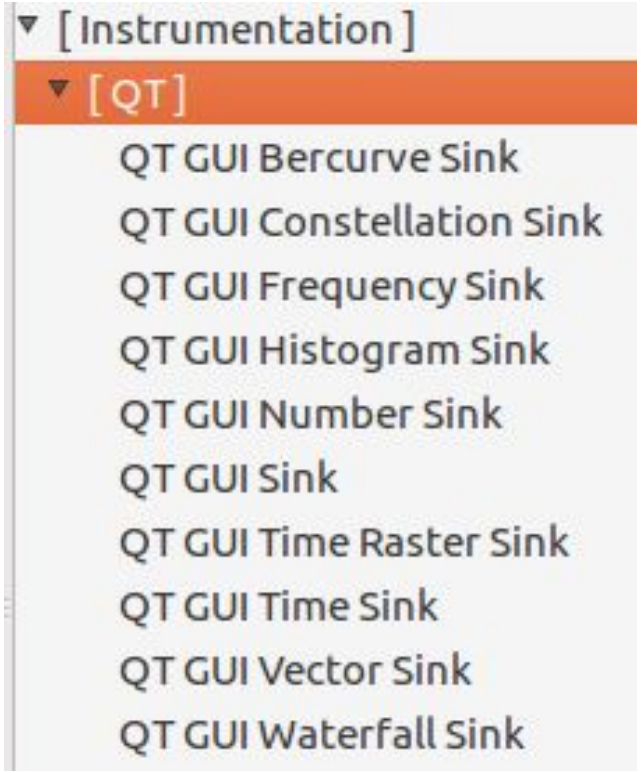
GNU Radio – Function Blocks

Must Have Blocks:

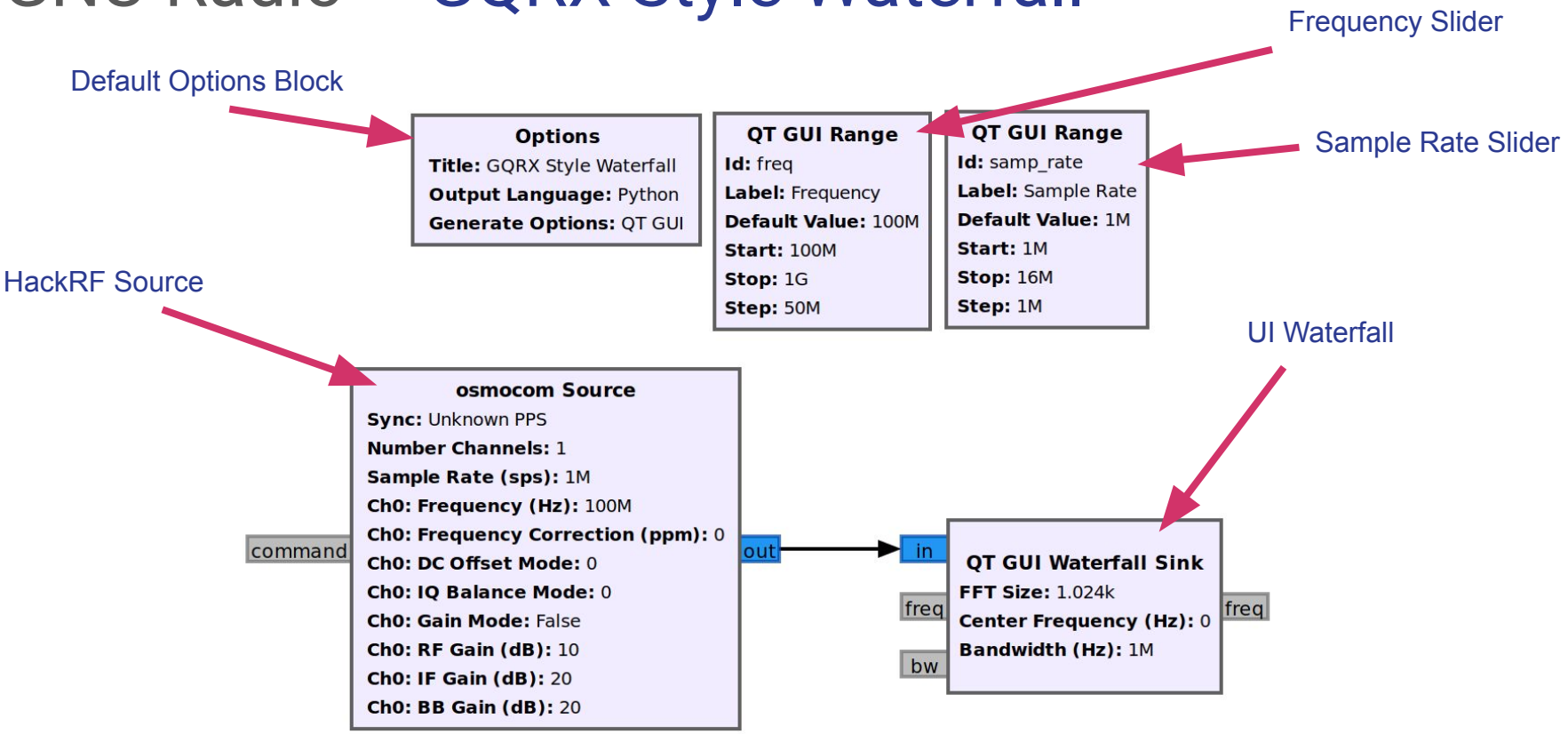
- Source
- Sink

Most Common Blocks:

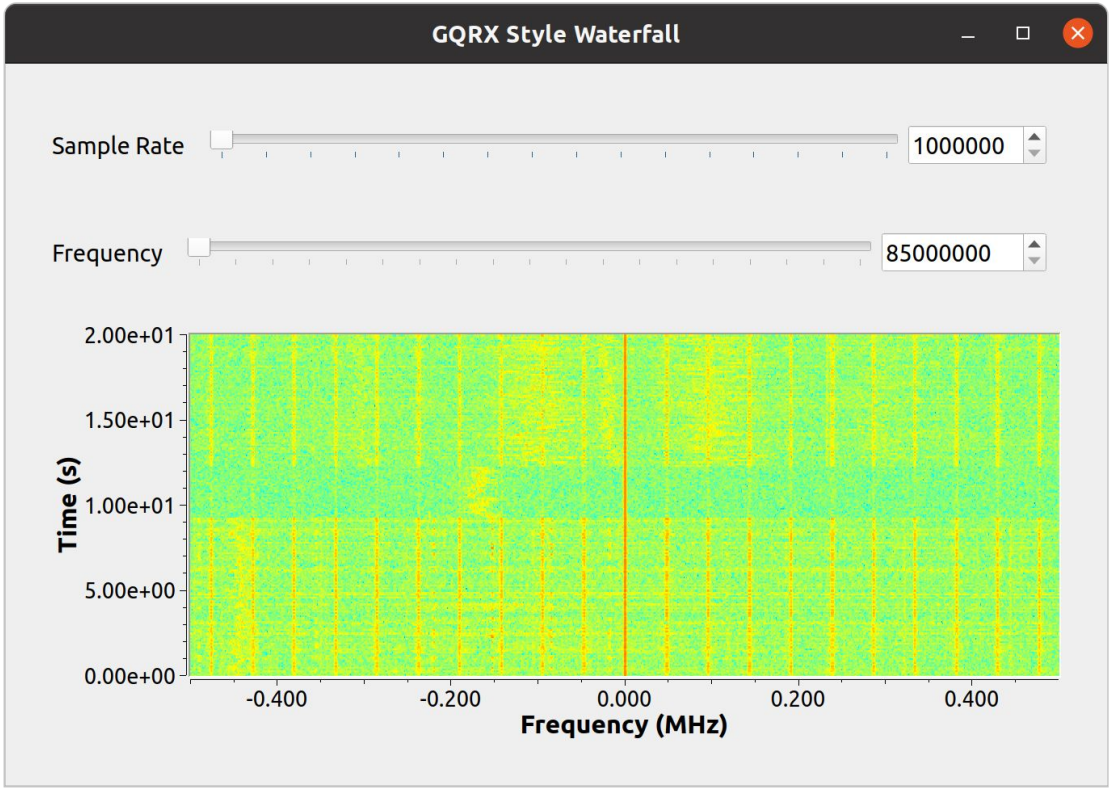
- Filters
- Instrumentation (aka measurements)
- Modems (Modulators and Demodulators)
- Variables and Controls



GNU Radio – GQRX Style Waterfall



GNU Radio – GQRX Style Waterfall



SDR Demo

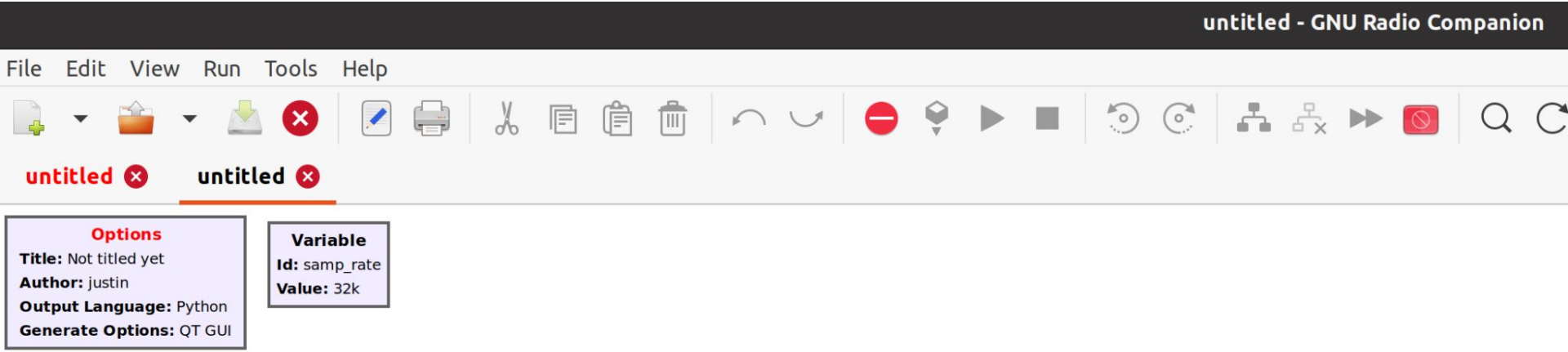
- Record and Replay Key Fob Signal

GNU Radio – Start up

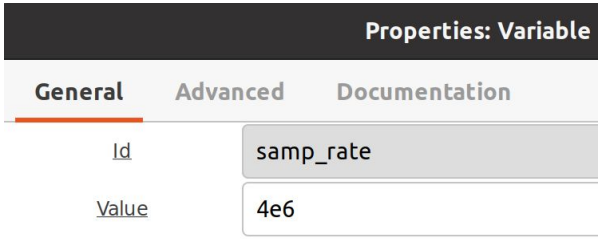
$ gnuradio-companion

```
justin@cybertruck:~/Downloads$ gnuradio-companion
<<< Welcome to GNU Radio Companion 3.8.1.0 >>>

Block paths:
        /usr/share/gnuradio/grc/blocks
```

Create a new project



Edit values



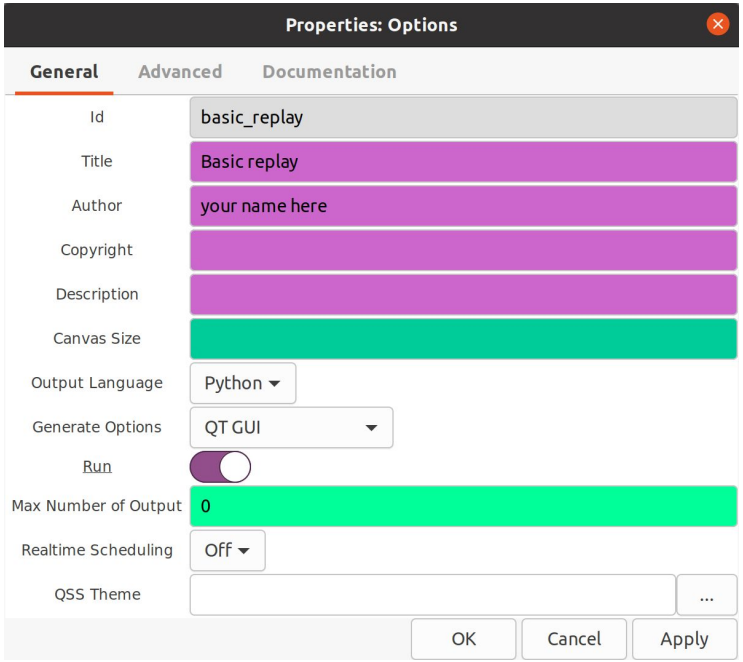
Options

Id: basic_replay

title: Basic Replay

Variable

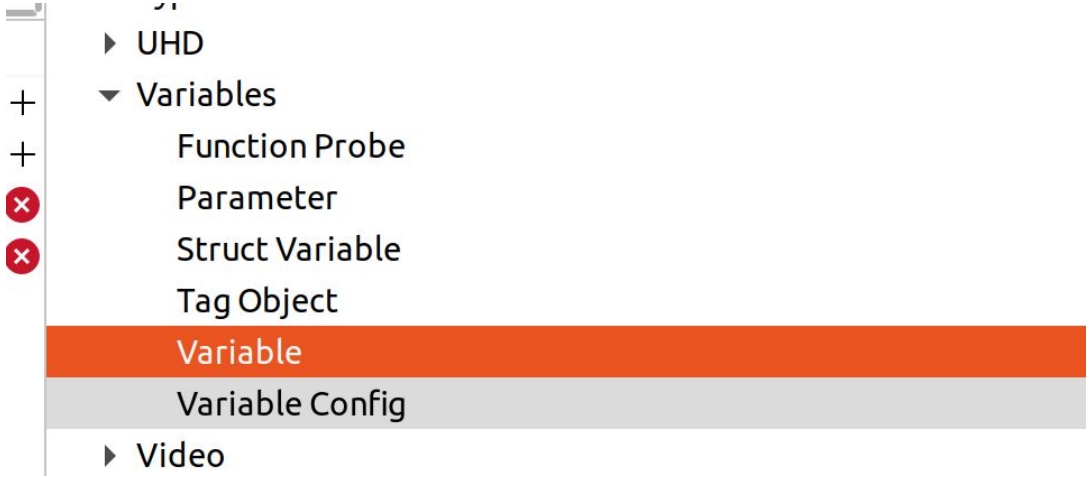
sample_rate: 4e6

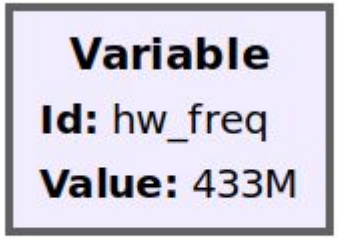


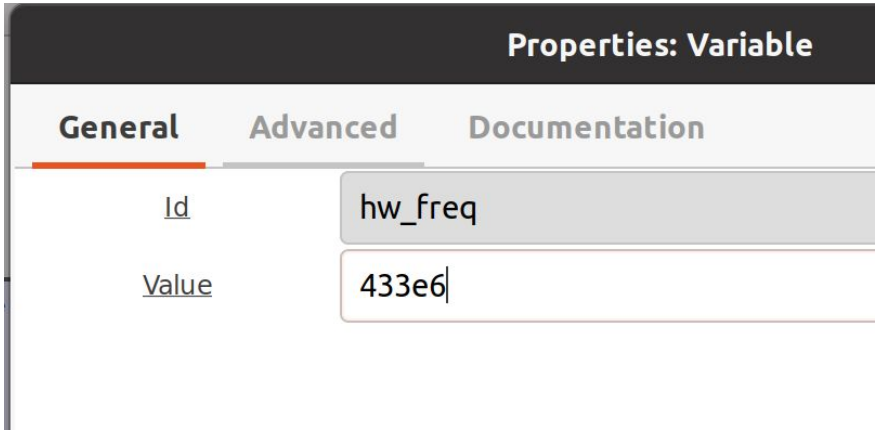
Create new

Id: hw_freq

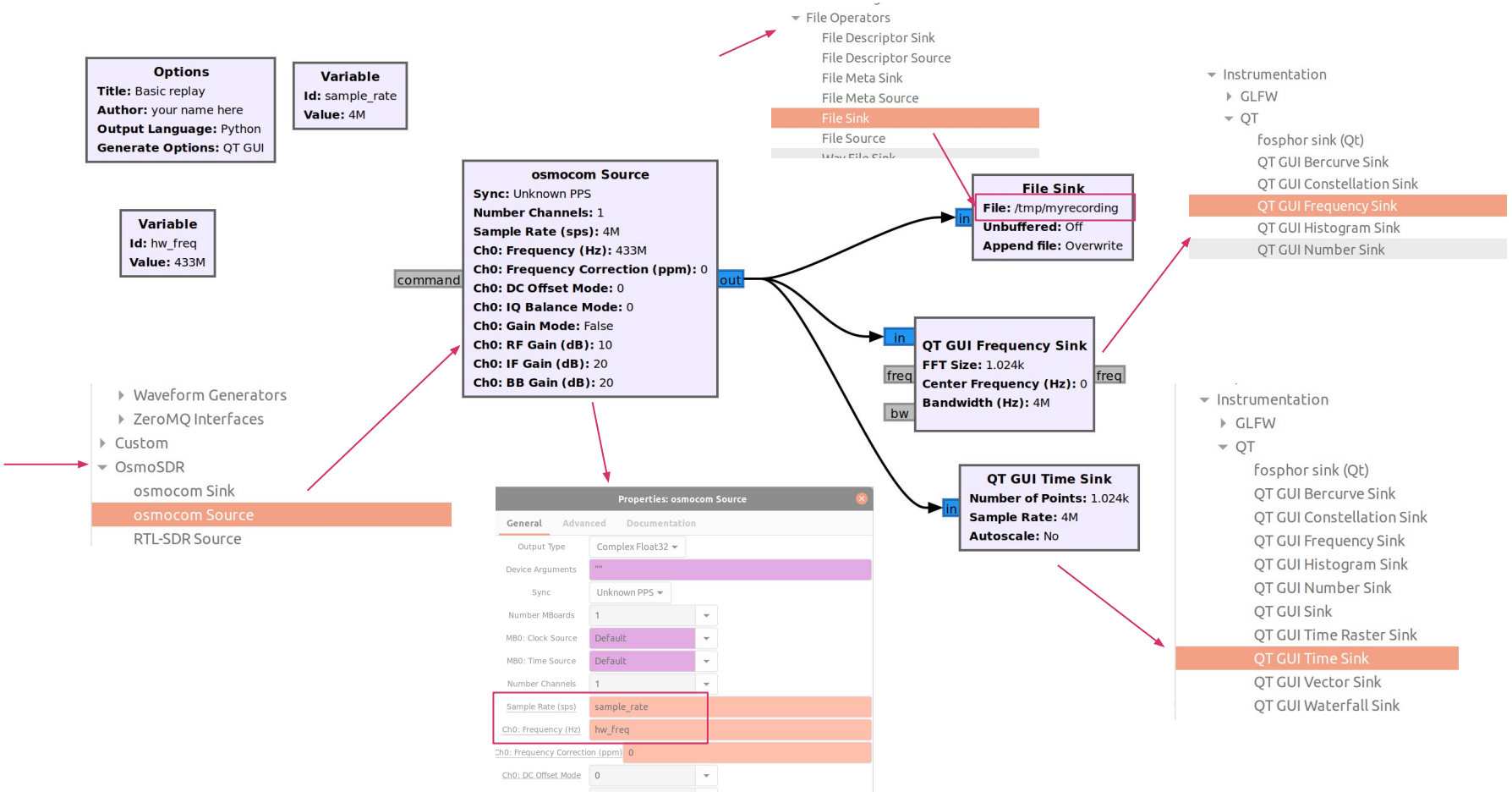
Value: 433e6



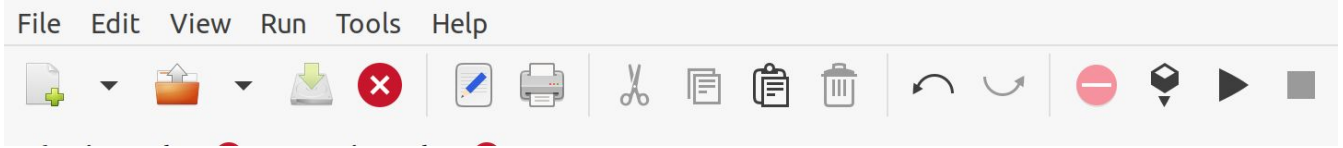


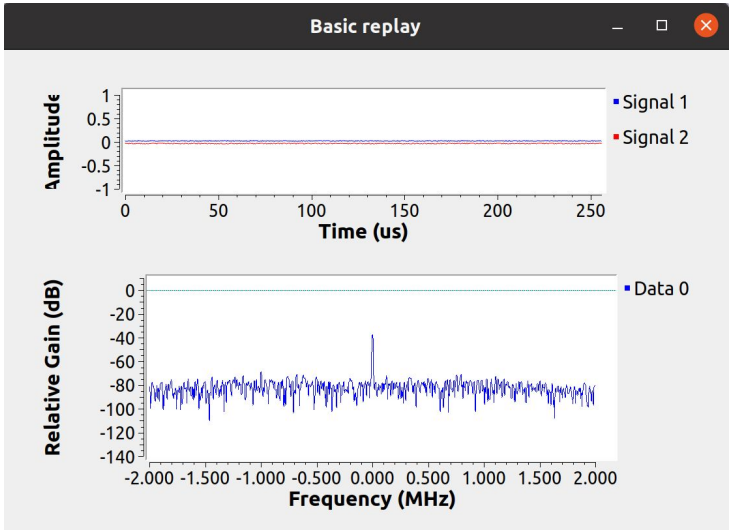


Basic capture to file

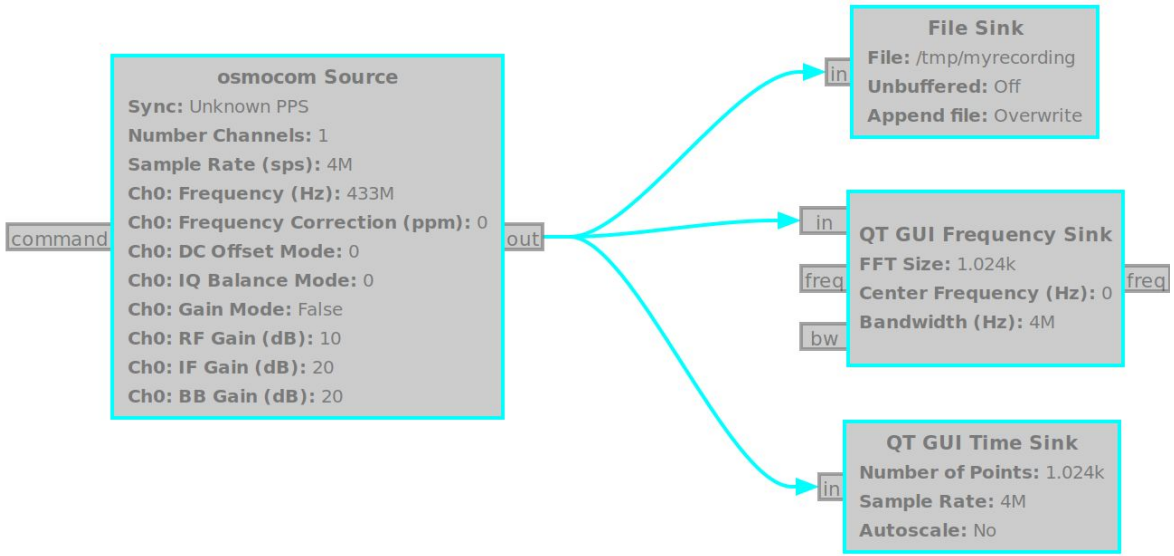


Execute

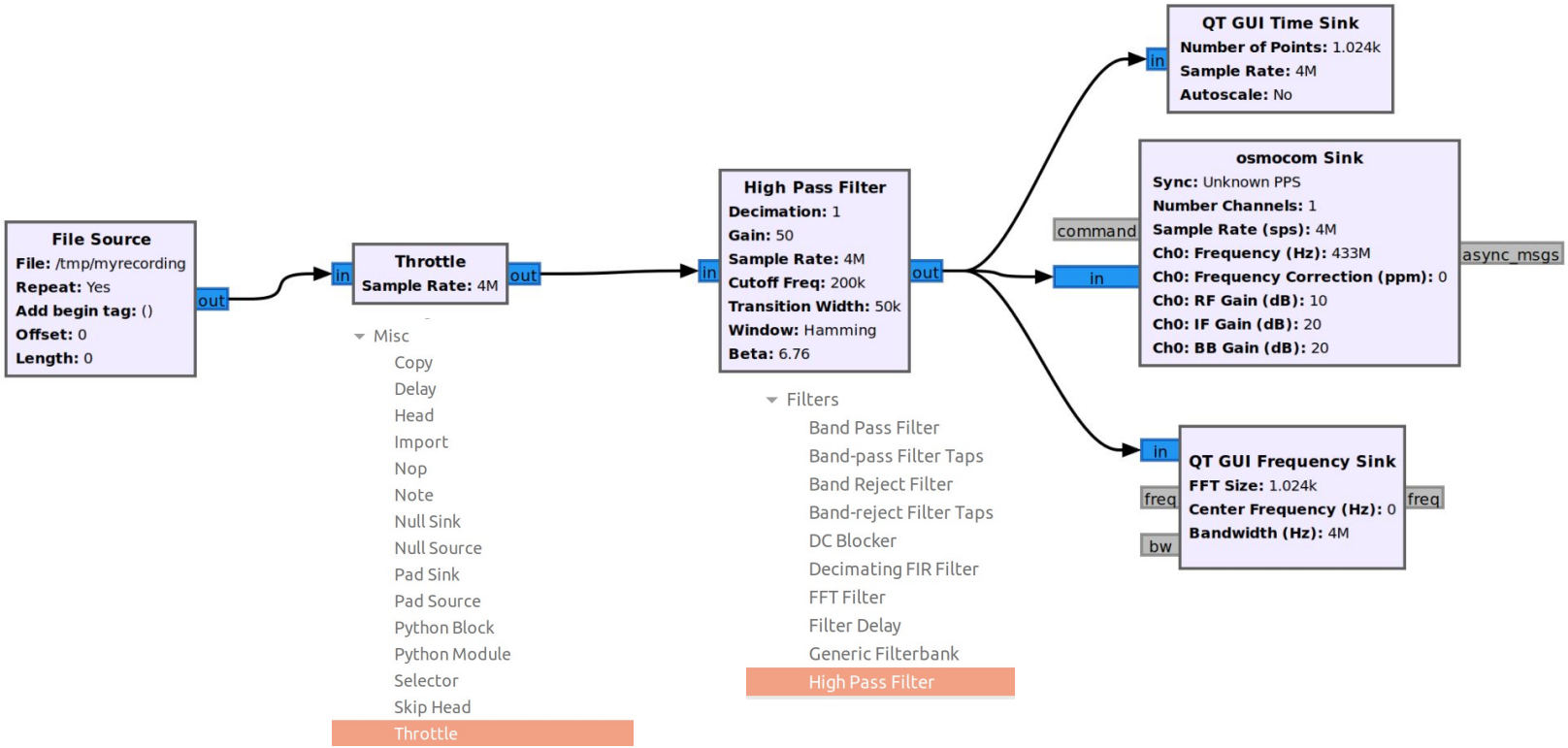

File
Edit
View
Run
Tools
Help


Basic replay
Amplitude
1
0.5
0
-0.5
-1
0
50
100
150
200
250
Time (us)
Signal 1
Signal 2
Relative Gain (dB)
0
-20
-40
-60
-80
-100
-120
-140
-2.000 -1.500 -1.000 -0.500 0.000 0.500 1.000 1.500 2.000
Frequency (MHz)
Data 0

Right-click -> disable



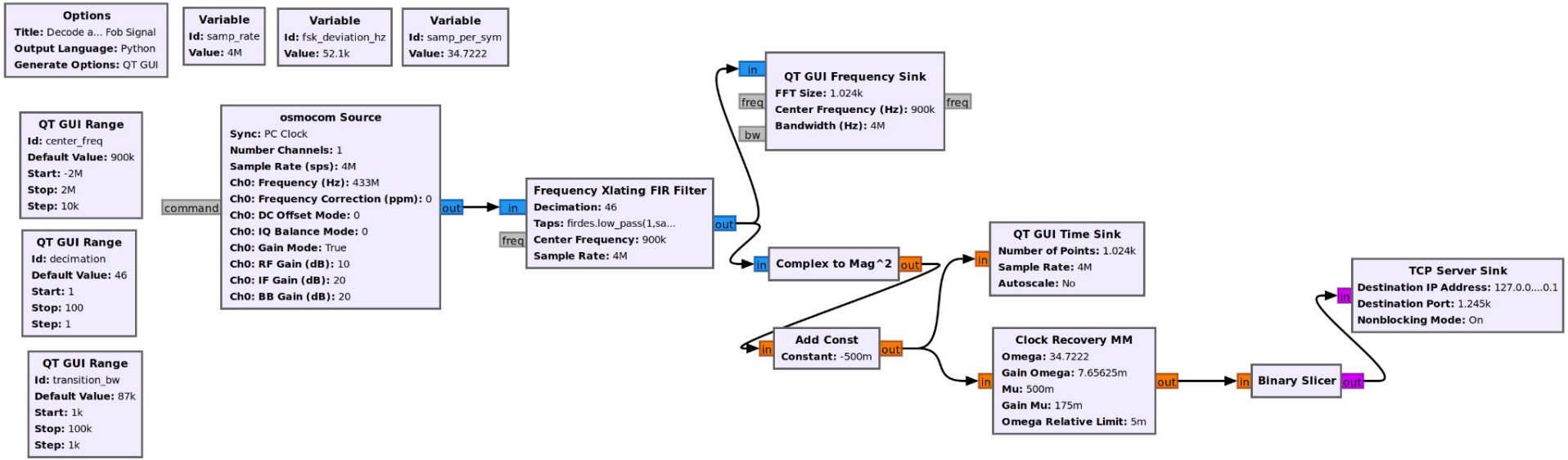
Replay


File Source
File: /tmp/myrecording
Repeat: Yes
Add begin tag: ()
Offset: 0
Length: 0
out
Throttle
Sample Rate: 4M
in
out
Misc
Copy
Delay
Head
Import
Nop
Note
Null Sink
Null Source
Pad Sink
Pad Source
Python Block
Python Module
Selector
Skip Head
Throttle
High Pass Filter
Decimation: 1
Gain: 50
Sample Rate: 4M
Cutoff Freq: 200k
Transition Width: 50k
Window: Hamming
Beta: 6.76
Filters
Band Pass Filter
Band-pass Filter Taps
Band Reject Filter
Band-reject Filter Taps
DC Blocker
Decimating FIR Filter
FFT Filter
Filter Delay
Generic Filterbank
High Pass Filter
QT GUI Time Sink
Number of Points: 1.024k
Sample Rate: 4M
Autoscale: No
osmocom Sink
Sync: Unknown PPS
Number Channels: 1
Sample Rate (sps): 4M
Ch0: Frequency (Hz): 433M
Ch0: Frequency Correction (ppm): 0
Ch0: RF Gain (dB): 10
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20
command
in
async_msgs
QT GUI Frequency Sink
FFT Size: 1.024k
Center Frequency (Hz): 0
Bandwidth (Hz): 4M
freq
bw
freq

SDR Demo

- Decode Key Fob Signal

GNU Radio – Decoding Key Fob Signal



GNU Radio – Decoding Key Fob Signal

```
nc 127.0.0.1 1245 | xxd | grep 101
```

```
00002480: 0100 0001 0100 0101 0001 0001 0100 0001  ................
00002490: 0001 0001 0000 0101 0001 0100 0100 0100  ................
000024a0: 0001 0001 0000 0100 0001 0001 0001 0101  ................
00002500: 0100 0100 0100 0101 0001 0100 0001 0100  ................
00002520: 0000 0100 0101 0001 0100 0100 0001 0000  ................
00002530: 0100 0101 0001 0001 0100 0100 0101 0000  ................
00002540: 0100 0100 0100 0100 0001 0000 0101 0000  ................
00002550: 0101 0001 0001 0001 0001 0000 0101 0001  ................
00002560: 0100 0100 0001 0000 0101 0001 0001 0100  ................
00002570: 0001 0100 0101 0001 0000 0100 0001 0100  ................
00002580: 0100 0100 0100 0101 0001 0100 0100 0100  ................
00002590: 0101 0000 0100 0101 0001 0100 0100 0100  ................
000025a0: 0101 0000 0101 0001 0001 0100 0100 0100  ................
000025b0: 0101 0100 0001 0100 0001 0000 0101 0001  ................
000025c0: 0100 0001 0001 0100 0101 0000 0100 0101  ................
000025e0: 0100 0001 0000 0101 0000 0100 0100 0001  ................
000025f0: 0100 0101 0000 0101 0100 0001 0001 0001  ................
00002610: 0000 0101 0100 0100 0100 0001 0000 0101  ................
00002620: 0001 0001 0001 0100 0100 0101 0000 0101  ................
00002640: 0100 0001 0100 0101 0001 0001 0100 0001  ................
00002650: 0001 0001 0000 0101 0001 0100 0100 0100  ................
00002660: 0001 0001 0000 0100 0001 0001 0001 0101  ................
00002670: 0101 0000 0000 0000 0000 0000 0000 0000  ................
```

Thank you! - Happy Hacking