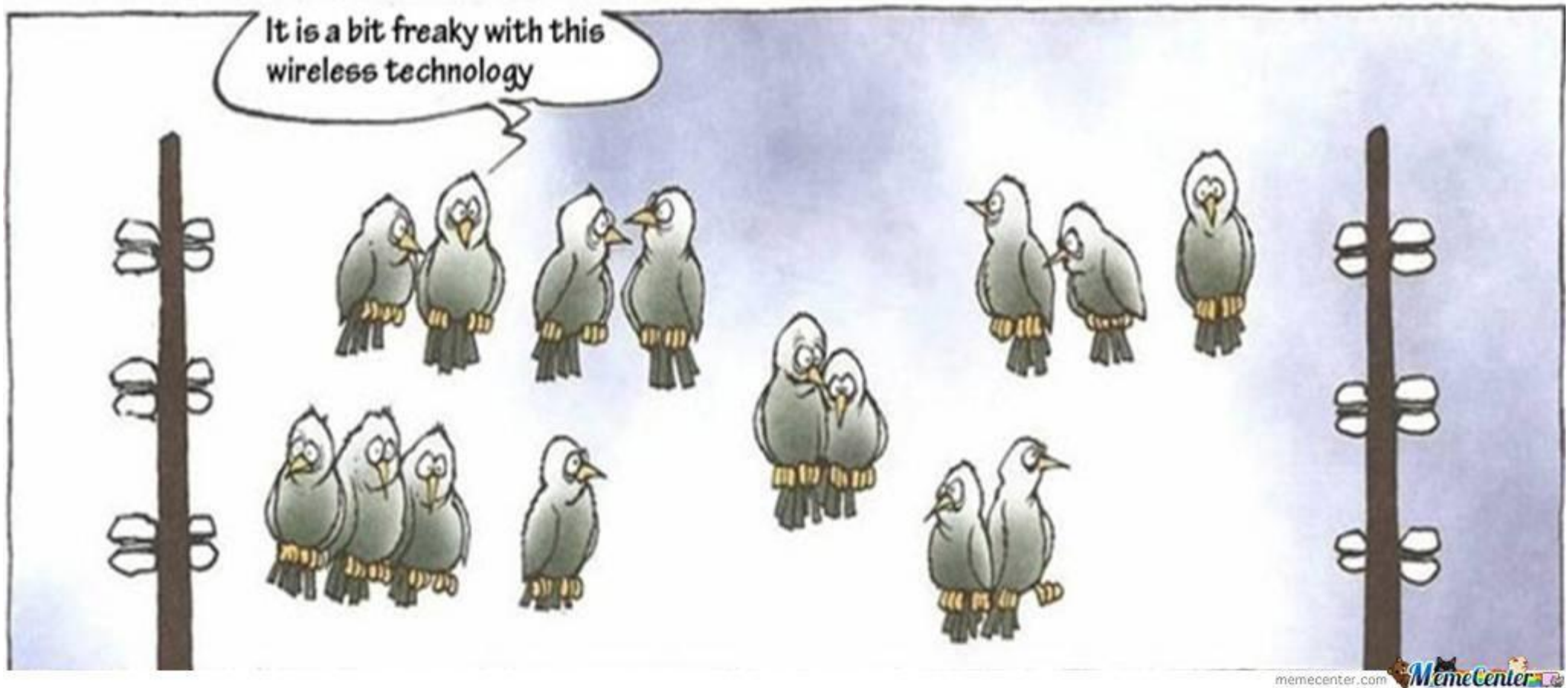# Radio Hacking with Software Defined Radio

" If you can't see doesn't mean that it is not there… "   - VU3TFQ
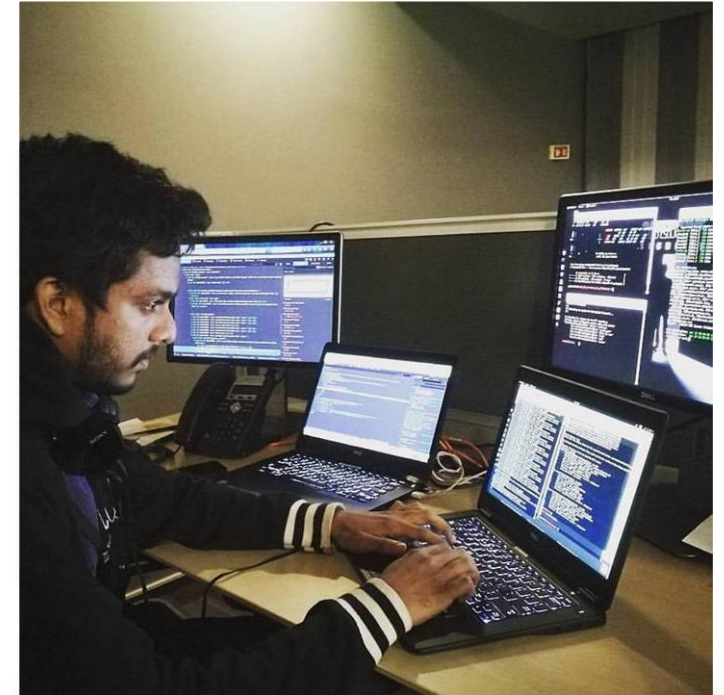
CQ…CQ… **Who is on this frequency ?**
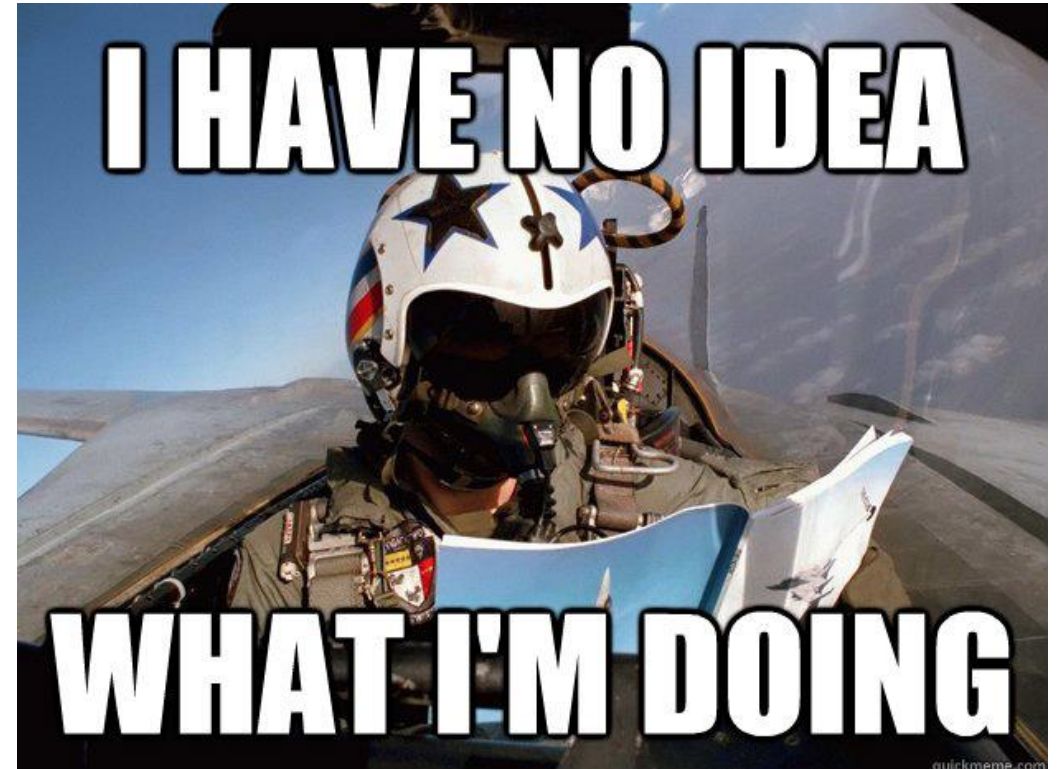
Thoufeeque N S

- Security Analyst in EY
- Hacking web applications for living
- Loves to play with Radio toys
- Licensed Ham Radio operator

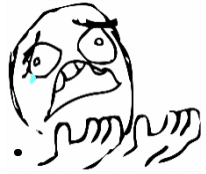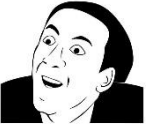# What is going on here ?

- What is SDR ?

- Type of Attacks

- Radios to use

- Software to use

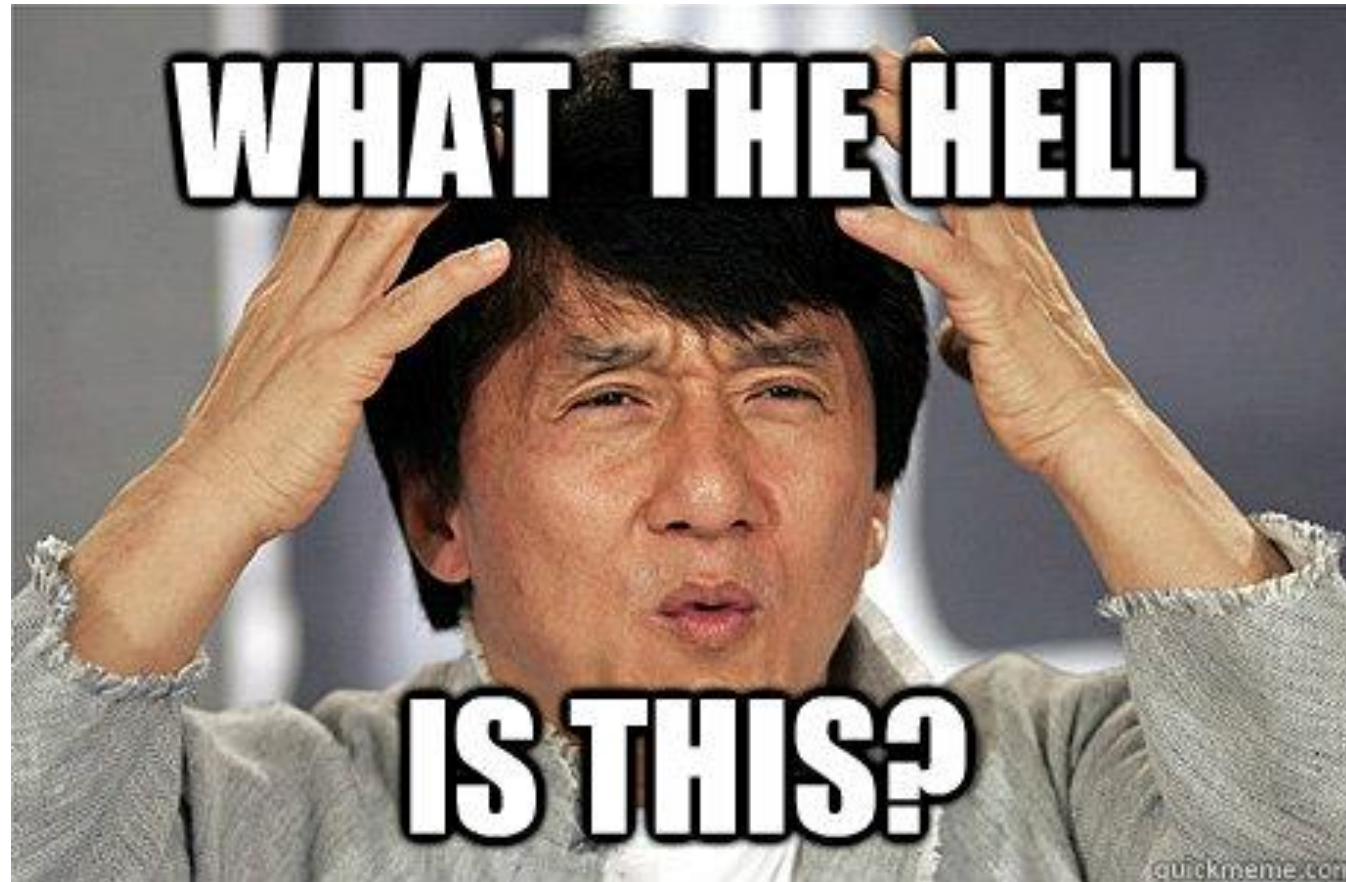- Antennas

- Demo*

*only if my radio doesn't go against me

Why Radio

- Radio doesn't get much focus from a security perspective.

- Radio is one of the core component on almost all IoT devices.

- You don't need physical access to exploit the device.

So… software defined radio, right ?

Software defined radio!

Software-defined radio (SDR) is a radio communication system where components that have been traditionally implemented in hardware (e.g. mixers, filters, amplifiers, modulators/demodulators, detectors, etc.) are instead implemented by means of software on a personal computer or embedded system.[1] While the concept of SDR is not new, the rapidly evolving capabilities of digital electronics render practical many processes which used to be only theoretically possible.

-Wikipedia😉

Possible attacks

- Reply attacks

- GPS Spoofing

- Sniffing GSM Signals

- Key fob attacks

Can I buy these SDR ?

Let's have a look at some SDR hardware that
we can use for hacking.
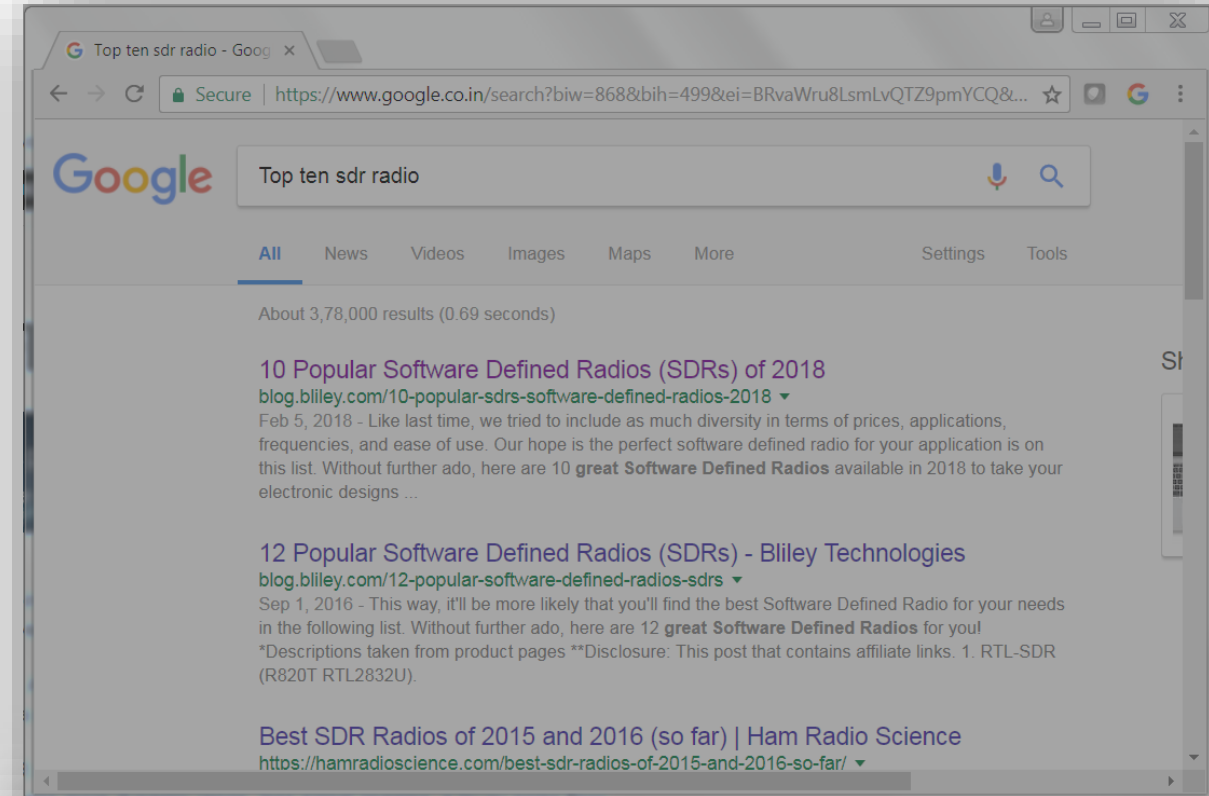
USRP

- Product by Ettus Research

- 1 TX, 1 RX, Half or Full Duplex

- Coverage from 70 MHz–6 GHz RF

- 2 SMA-Bulkhead Cables

https://www.ettus.com/

Cost you around $2000

HackRF

- 1 MHz to 6 GHz operating frequency

- half-duplex transceiver

- open source hardware

- compatible with GNU Radio, SDR#, and more

https://greatscottgadgets.com/hackrf/

Somewhere between $300-$400

bladeRF

300MHz - 3.8GHz RF frequency range

Independent RX/TX 12-bit 40MSPS quadrature sampling

Capable of achieving full-duplex 28MHz channels

https://www.nuand.com/blog/product/bladerf-x40/

50000 INR !

Don't you see a problem here ? Over!

Wait… We have a solution.

It's called RTL-SDR!

How much it cost ?

$20

No. I don't belive!.

Trust me..   It's only $20.

Yeah, This is true. We have an SDR that will set you back $20.

RTL- SDR

- Meant to be a TV tuner card
- Converted it as an SDR
- RX b/w 24 - 1766 MHz
- Listen to AIS, ADS-B, GPS, GSM communications….

Below 1000 INR

Some cool things you can do with RTL-SDR

- Receive aircraft signals (ADS-B).
- AIS (Ship transition) can be received.
- You can sniff most protocols in the air.
- Sniff radio communication between IoT devices.
- Once you received the data you can decode the it (It requires a lot knowledge and effort)

**Now we have the hardware... What's next ?**

The software... What else!

GNU Radio

GNU Radio is a free software development toolkit that provides signal processing blocks to implement software-defined radios and signal-processing systems. It can be used with external RF hardware to create software-defined radios, or without hardware in a simulation-like environment. It is widely used in hobbyist, academic, and commercial environments to support both wireless communications research and real-world radio systems.
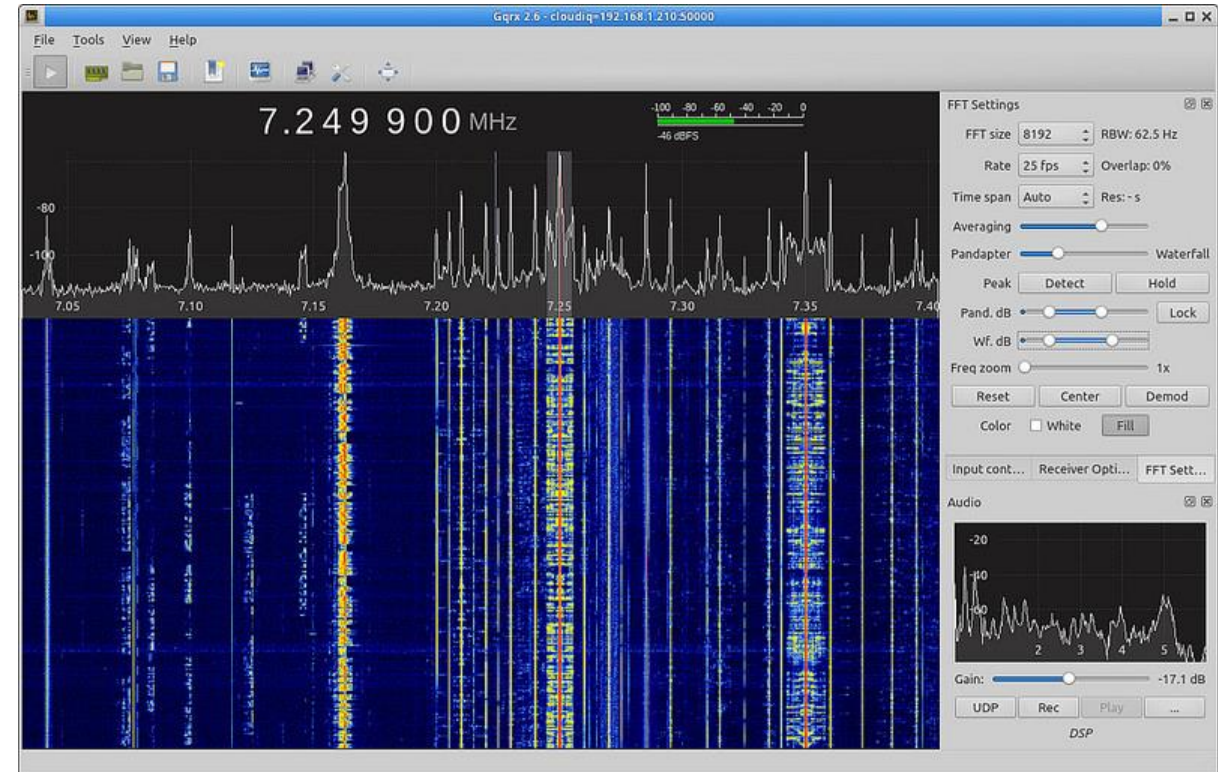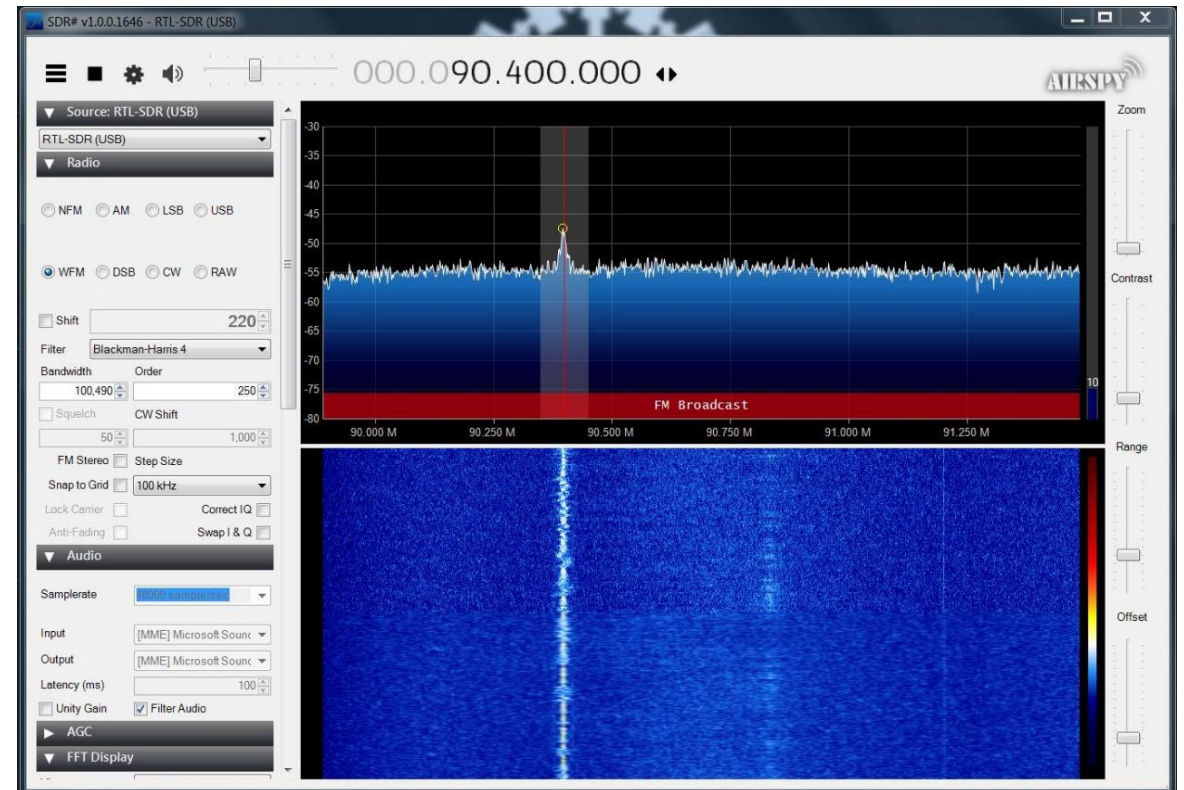
- wikipedia 😉
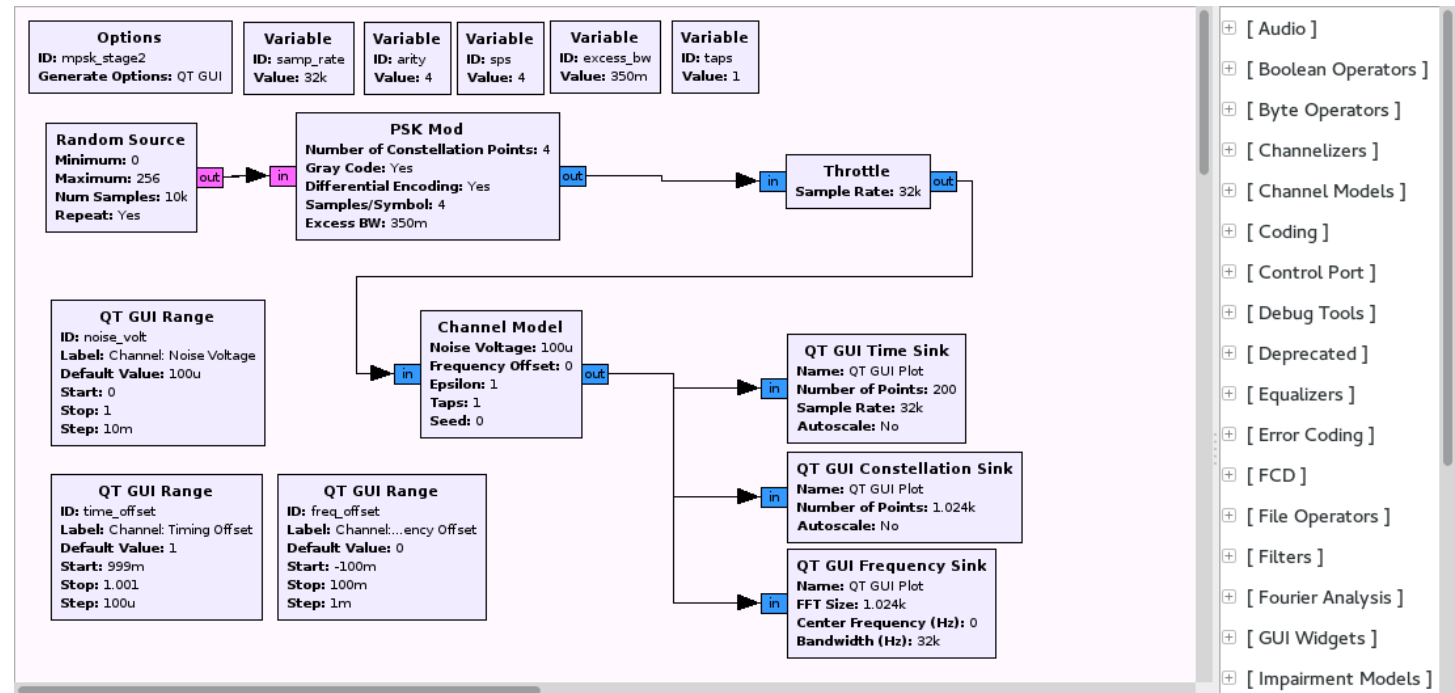
gqrx

- For linux and osx
- Best to view the waterfall diagram
- GUI

SDR Sharp

- Spectrum Display
- Waterfall Display
- Recording
- Windows tool
- GUI

Rtl_sdr

- Linux package

- Command line tool

- Can be used to record signals with rtl sdr

```
root@kali:~# rtl_fm -h
rtl_fm, a simple narrow band FM demodulator for RTL2832 based DVB-T receivers

Use:     rtl_fm -f freq [-options] [filename]
         -f frequency_to_tune_to [Hz]
            use multiple -f for scanning (requires squelch)
            ranges supported, -f 118M:137M:25k
         [-M modulation (default: fm)]
            fm, wbfm, raw, am, usb, lsb
            wbfm == -M fm -s 170k -o 4 -A fast -r 32k -l 0 -E deemp
            raw mode outputs 2x16 bit IQ pairs
         [-s sample_rate (default: 24k)]
         [-d device_index (default: 0)]
         [-T enable bias-T on GPIO PIN 0 (works for rtl-sdr.com v3 dongles)]
         [-g tuner_gain (default: automatic)]
         [-l squelch_level (default: 0/off)]
         [-p ppm_error (default: 0)]
         [-E enable_option (default: none)]
            use multiple -E to enable multiple options
            edge:   enable lower edge tuning
            dc:     enable dc blocking filter
            deemp:  enable de-emphasis filter
            direct: enable direct sampling
            offset: enable offset tuning
         filename ('-' means stdout)
            omitting the filename also uses stdout

Experimental options:
         [-r resample_rate (default: none / same as -s)]
         [-t squelch_delay (default: 10)]
            +values will mute/scan, -values will exit
         [-F fir_size (default: off)]
            enables low-leakage downsample filter
            size can be 0 or 9.  0 has bad roll off
         [-A std/fast/lut choose atan math (default: std)]

Produces signed 16 bit ints, use Sox or aplay to hear them.
         rtl_fm ... | play -t raw -r 24k -es -b 16 -c 1 -V1 -
                    | aplay -r 24k -f S16_LE -t raw -c 1
         -M wbfm  | play -r 32k ...
         -s 22050 | multimon -t raw /dev/stdin
```

We got hardware and software… What is next ?
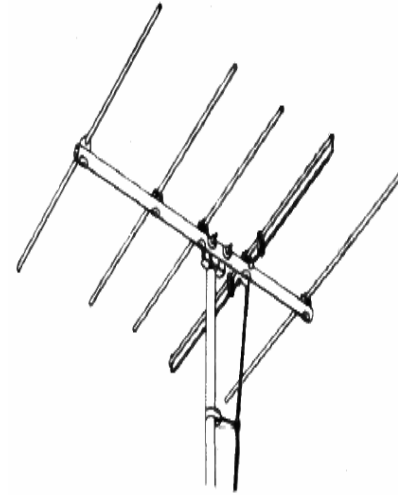
Antenna!

Antennas…….. Antennas everywhere !

Ground plan antenna

Rubber ducky antenna

Turnstile antenna

Yagi-Uda antenna

Hardware, Software, Antenna…

So what is next ?

This is just a beginning.

- There are much more tools available for radio research.

- We are planning to conduct a series of Practical radio hacking session in the future events of DC0471.

- So stay tuned for more cool stuffs.

MY GOD, IT'S JUST THE BEGINNING!

Q&A time…

Just kidding… :D

Over... Over...

Thank you!

/thoufeequens
/thoufeequens
/thoufeequens