osmocom: Overview of our SDR projects

rtl-sdr, gr-osmosdr, osmo-tetra, osmo-gmr, gr-fosphor and more !

Sylvain Munaut

FOSDEM 2014, February 2nd, 2014

About the speaker

- Linux and free software enthusiast since 1999
- M.Sc. in C.S. + some E.E.
- General orientation towards low level
  - Embedded, Kernel, Drivers and such.
  - Hardware (Digital stuff, FPGA, RF, ...)
- Interest in RF telecom for about 5 years
  - GSM, GMR-1, TETRA, POCSAG, ...
  - Within the Osmocom project
  - Mostly in my spare time

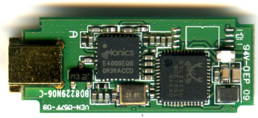
About osmocom

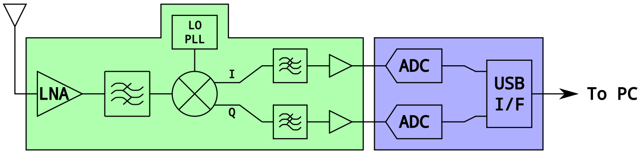
- osmocom
  - **O**pen **S**ource **Mo**bile **Com**munication
  - That's what happens when engineers name things.
- Collection of Free Software / Open Source Software projects in the area of mobile communications.
- Spawned off the OpenBSC project at the start of 2010.
- Originally geared toward GSM but evolved over time to include many other projects / protocols.
  - A growing share of them involving SDR, which is what will be talked about today.
- Disclosure: I'm not actively involved in all of theses, I'm just introducing them.

Outline

rtl-sdr

- Turns consumer DVB-T USB dongle into SDR receivers
- Readily available at very low cost (15 EUR)
- Basic SDR architecture
  - Contains LNA / Filter / IQ Mixer / low pass / ADC / USB
  - Retrieves raw IQ radio samples
  - Tunes from 50 MHz to 2 GHz (depends on tuner model)
- Performance is not great
  - But sufficient for a lot of protocols
  - POCSAG, ADSB, P25, GSM, GMR-1, TETRA, APT, ...





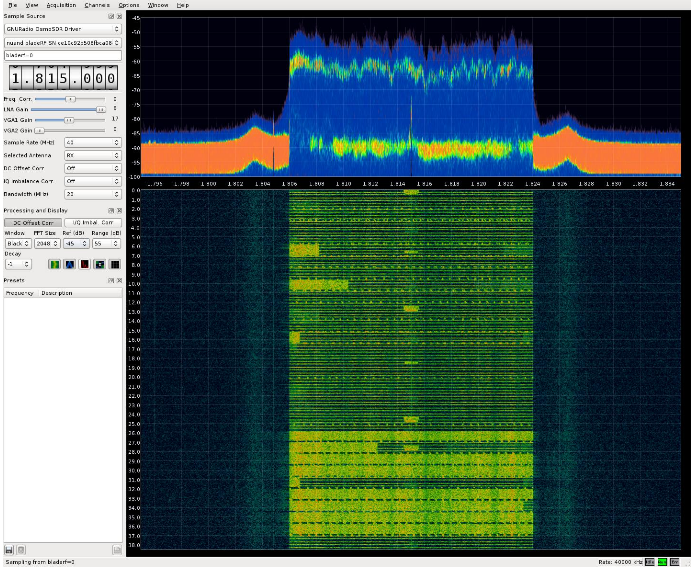
libmirisdr

- Equivalent of rtl-sdr for the Mirics based TV dongle
- Not as refined as rtl-sdr
  - Closer to a "Proof of Concept"
- Better raw specs (sample rate, bit depth)
- Hardware is more expensive and less available
- RF wiring of most dongles prevents wide band usage
  - Tuner has multiple antenna input depending on frequency band
  - Often, only one is connected, others are grounded

gr-osmosdr

- Abstraction library / gnuradio block for radio hardware
  - API to query frequency range, sample rates, gains, ...
  - Allows applications to work with any supported hardware
- Supported backends :
  - IQ file, FCD(PP), UHD (USRPs / UmTRX), OsmoSDR, rtl-sdr, rtl-tcp, mirisdr, HackRF, BladeRF, RFSPACE receivers, AirSpy
  - Thanks to hardware vendors who donated hardware: Fairwaves, HackRF, Lime, Nuand, Nutaq, RFSPACE, AirSpy
- Used by many applications already
  - gqrx, gr-modes, OpenLTE, OP25, ...
  - You should use it too for your next project !
- Includes demo applications:
  - `osmocom_fft`: Spectrum analyzer
  - `osmocom_siggen`: Signal generator
  - And more to come ...

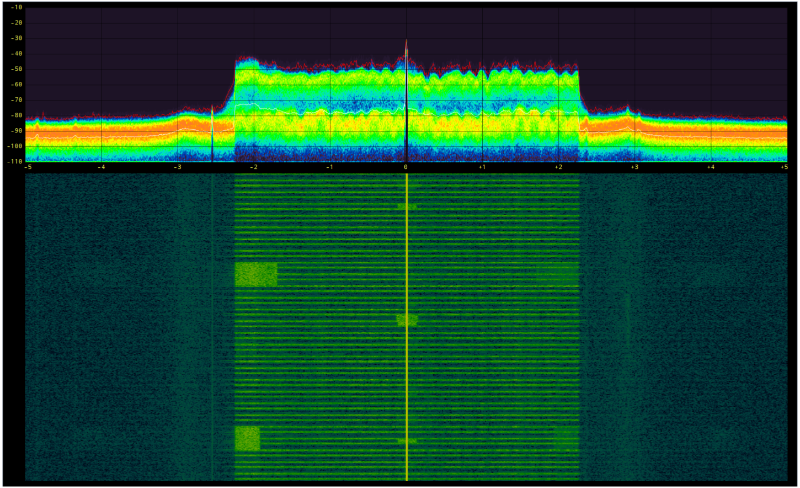
SDRangelove (1)



SDRangelove (2)

- Standalone SDR application
- Gorgeous UI using GL and Qt5
  - Can be resource hungry though
- Hardware support
  - rtl-sdr and OsmoSDR via native drivers
  - Many others through gr-osmosdr plugin in "standalone" mode
- Plugin interface
  - Demodulation support upcoming (WFM/NFM/AM/...)
- Saved profiles / Bookmarked channels
- Export of channelized data to external application by TCP

gr-fosphor (1)



gr-fosphor (2)

- GNU Radio block version of SDRangelove main display
  - Complete rewrite for GPU acceleration
  - Uses mix of OpenCL & OpenGL
  - Meaning you need drivers that support this for it to run at all
- Very fast
  - 30 Msps on laptop (G96M), up to more than 200 Msps for high-end GPUs (HD7870 / GTX760)
  - Perfect for all those new wideband SDRs
- Will eventually be integrated to SDRangelove as well
- Integrated to `osmocom_fft` via the `-F` option

SDRangelove / gr-fosphor

Common display elements

4 main parts :

- FFT
  - Process every samples at the input in at least one FFT window
  - Ideally in several, using overlapping windows (future)
- Live spectrum
  - IIR average of all computed spectra
  - So many of them that noise is smoothed while keeping responsiveness
- Waterfall
  - Again, all spectra used
  - 1:1 currently, in the future $1{:}2^n$ using aggregation (min/max/avg)
- Histogram
  - Statistical view of spectra
  - Perfect to see bursts, glitches or any transients

osmo-gmr

- GMR-1: **G**EO **M**obile **R**adio
  - Sat-phone protocol heavily inspired by GSM
  - Used by Thuraya
- `osmo-gmr`: SDR + PHY layer implementation
  - PHY is RX/TX, SDR is RX only currently
  - GNURadio to channelize
  - IQ samples piped to custom application
- Wireshark dissector
- Cipher reverse-engineered (& broken)
- Voice codec: DVSI AMBE variant
  - Reverse-engineered and pure-C implementation published
- Future
  - GmPRS (packet data)
  - TX support
  - Better GNU Radio integration

osmo-tetra

- TETRA: **Te**rrestrial **T**runked **Ra**dio
  - Digital trunking designed for government, emergency services, public safety, ...
  - Widely used in Europe, including here in BE (look around 390M)
- `osmo-tetra`: Implementation of TETRA radio-modem
  - PHY / MAC layers
  - RX fully supported. TX partial.
  - GNU Radio for channelization and demodulation
  - Custom application for the upper layers
- Can be used for air-interface intercept
- Wireshark dissector available for protocol analysis
- Voice support (not in `master` yet)

OP25

- APCO Project 25
  - Digital trunking designed for police and emergency services
  - Deployed most notably in US, Canada and Australia
- `OP25`: Implementation of APCO P25 using SDR
  - Based on GNU Radio
- All-in-one
  - Demodulator
  - Protocol stack (PHY/...)
  - Voice codec (IMBE only)
  - Wireshark packet dissection
  - Crypto
- Recently updated to GR3.7 and latest gr-osmosdr

Thanks

Thanks to all my osmocom fellow developpers and in general everyone working on free software SDR.

Thank you for your attention !

Questions ?

Any questions ?

Resources

- osmocom
  - `http://osmocom.org`
  - `http://git.osmocom.org`
- rtl-sdr: `http://sdr.osmocom.org/trac/wiki/rtl-sdr`
- libmirisdr: `http://git.osmocom.org/libmirisdr/`
- gr-osmosdr: `http://sdr.osmocom.org/trac/wiki/GrOsmoSDR`
- gr-fosphor: `http://sdr.osmocom.org/trac/wiki/`
- sdrangelove: `http://sdrangelove.org`
- osmo-gmr: `http://gmr.osmocom.org`
- osmo-tetra: `http://tetra.osmocom.org`
- op25: `http://op25.osmocom.org`