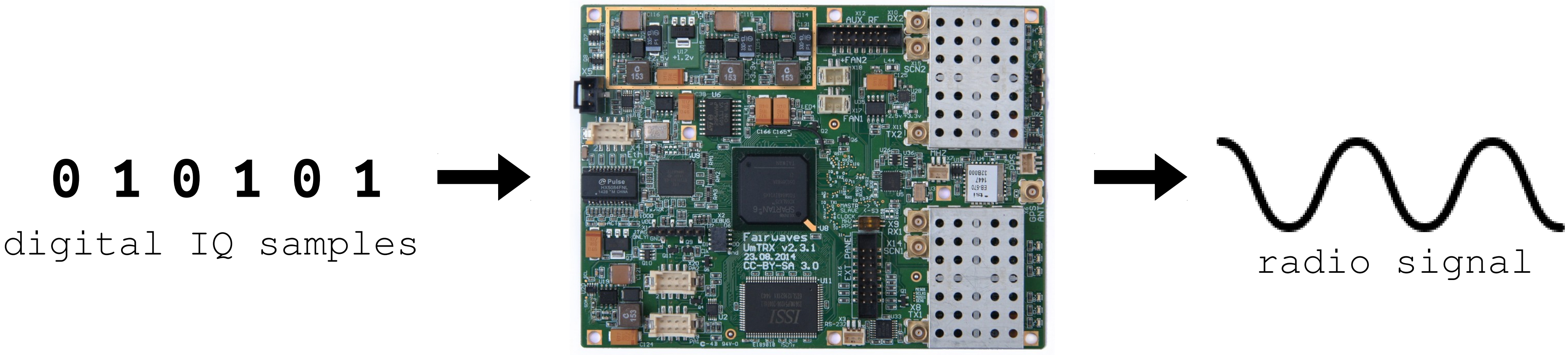
Software Defined Radio hardware for Osmocom BTS

Alexander Chemeris
CTO, Fairwaves, Inc.

Software Defined Radio (SDR):

“a sound-card for radio waves”

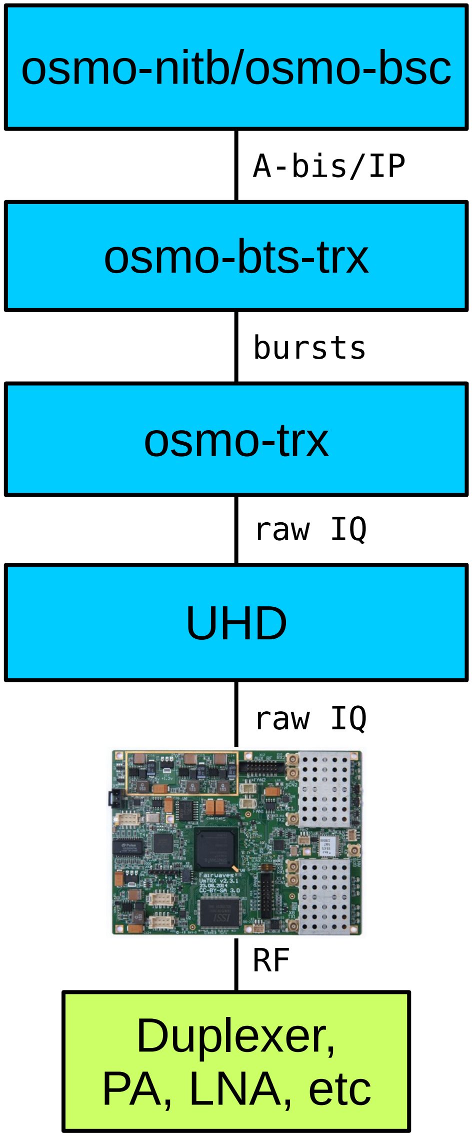


0 1 0 1 0 1

digital IQ samples

radio signal

Architecture



osmo-nitb/osmo-bsc

A-bis/IP

osmo-bts-trx

bursts

osmo-trx

raw IQ

UHD

raw IQ

RF

Duplexer, PA, LNA, etc

L3+: signaling and above

L1+L2: physical layer, data-link layer

L0 aka Lower L1: modulation/demodulation

Software Defined Radio driver

Software Defined Radio

Radio front-end

Hitchhiker’s guide to SDR hardware

- **UmTRX**
  - Designed specifically for GSM
  - Has open-source hardware version
- **USRP B2x0**
  - Popular flexible SDR with USB interface

most popular

- USRP1
  - The first SDR used for GSM back in 2008 by OpenBTS project
  - No longer actively supported
- USRP N2x0/X3x0
  - High-end SDR with Ethernet interface
- USRP Exxx
  - Embedded SDR with Xilinx Zynq CPU (ARM+FPGA)
- LimeSDR
  - New inexpensive SDR with USB interface
- XTRX
  - Upcoming inexpensive SDR with miniPCIe/PCIe interface
- No BladeRF support
  - Should be capable, but no one has bothered to add support = volunteers welcome!
- No HackRF support
  - Because it’s half-duplex

Parameters important for GSM (1/2)

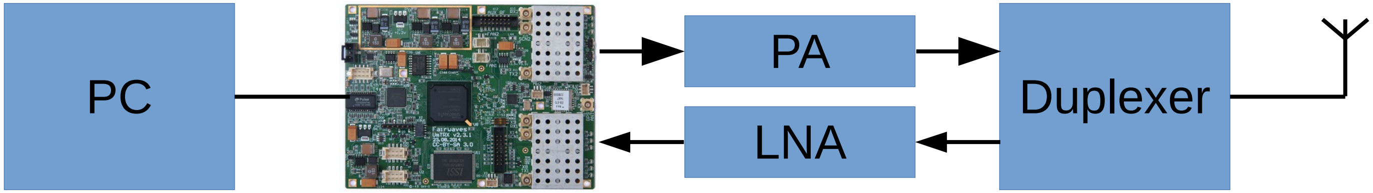
- Clock accuracy
  - 0.05 ppm / 0.1 ppm requirement for macro/picocells
  - 0.2 ppm - 0.4 ppm is ok for lab testing
  - 1-2 ppm clocks in most SDRs (USRPs, LimeSDR, BladeRF, etc)
    - Requires external clock source *or* GPSDO *or* regular calibration
  - UmTRX/XTRX are the only popular SDR I know of with 0.1 ppm clock and built in GPSDO
- Clock rate
  - GSM symbol clock rate is 13/48 MHz
  - Integer multiples are recommended – fractional resampling is CPU intensive
  - USRP1 (64MHz) and USRP N (50MHz) clocks are not integer multiples
  - UmTRX clock is integer multiple (26 MHz)
  - USRP B2x0, X3x0, LimeSDR, XTRX have flexible clock and are configured for an integer multiple

Parameters important for GSM (2/2)

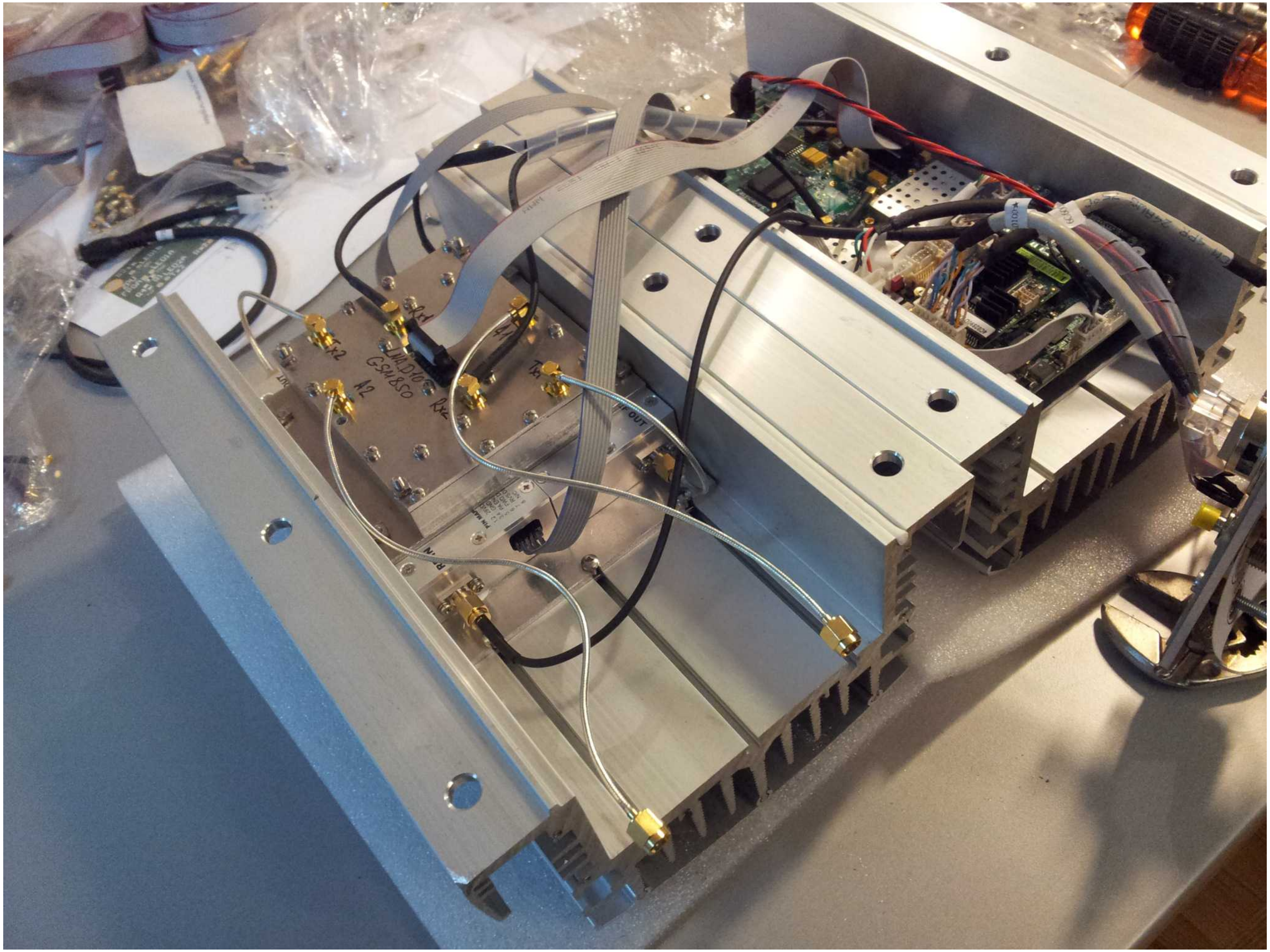
- Interface
  - USB (USRP B2x0, LimeSDR)
    - convenient as a portable development setup, but has variable latency and is not very stable in long term deployments.
  - Ethernet (UmTRX, USRP N2x0, USRP X3x0)
    - stable in long running applications, fixed latency
  - PCIe (XTRX)
    - stable in long running applications, almost no latency
- Output RF power
  - <= 10dBm (USRP B2x0, E3x0, LimeSDR)
    - few meters of coverage
  - >= 20dBm (UmTRX, USRP N2x0, X3x0)
    - 100m+ of coverage

Radio Frontend

- Power Amplifier (PA)
  - Amplifies transmitted signal
- Low-Noise Amplifier (LNA)
  - Amplifies received signal
- Duplexer
  - Splits transmitted signal from received signal
  - Filters out out-of-band spurs

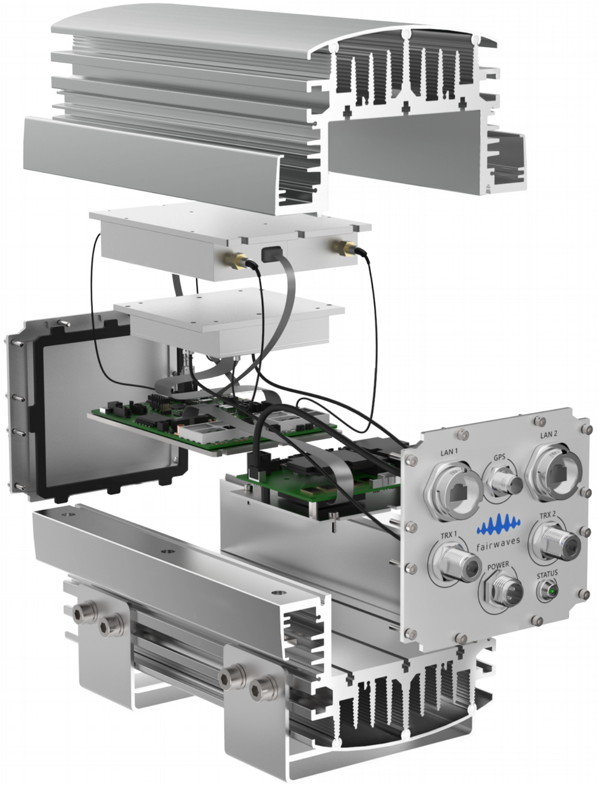


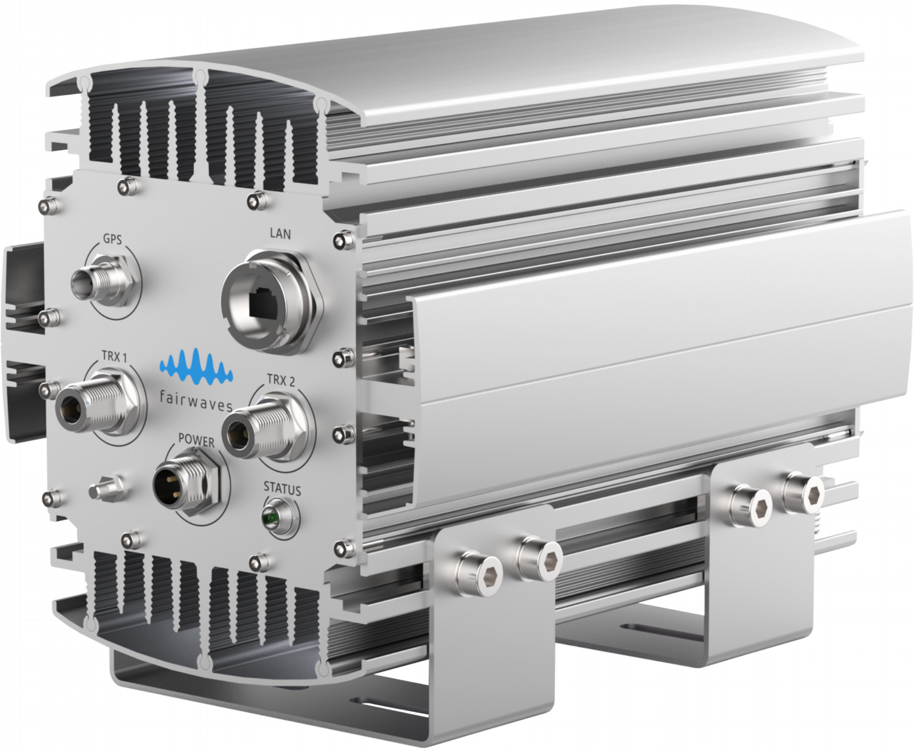
UmSITE base station inside



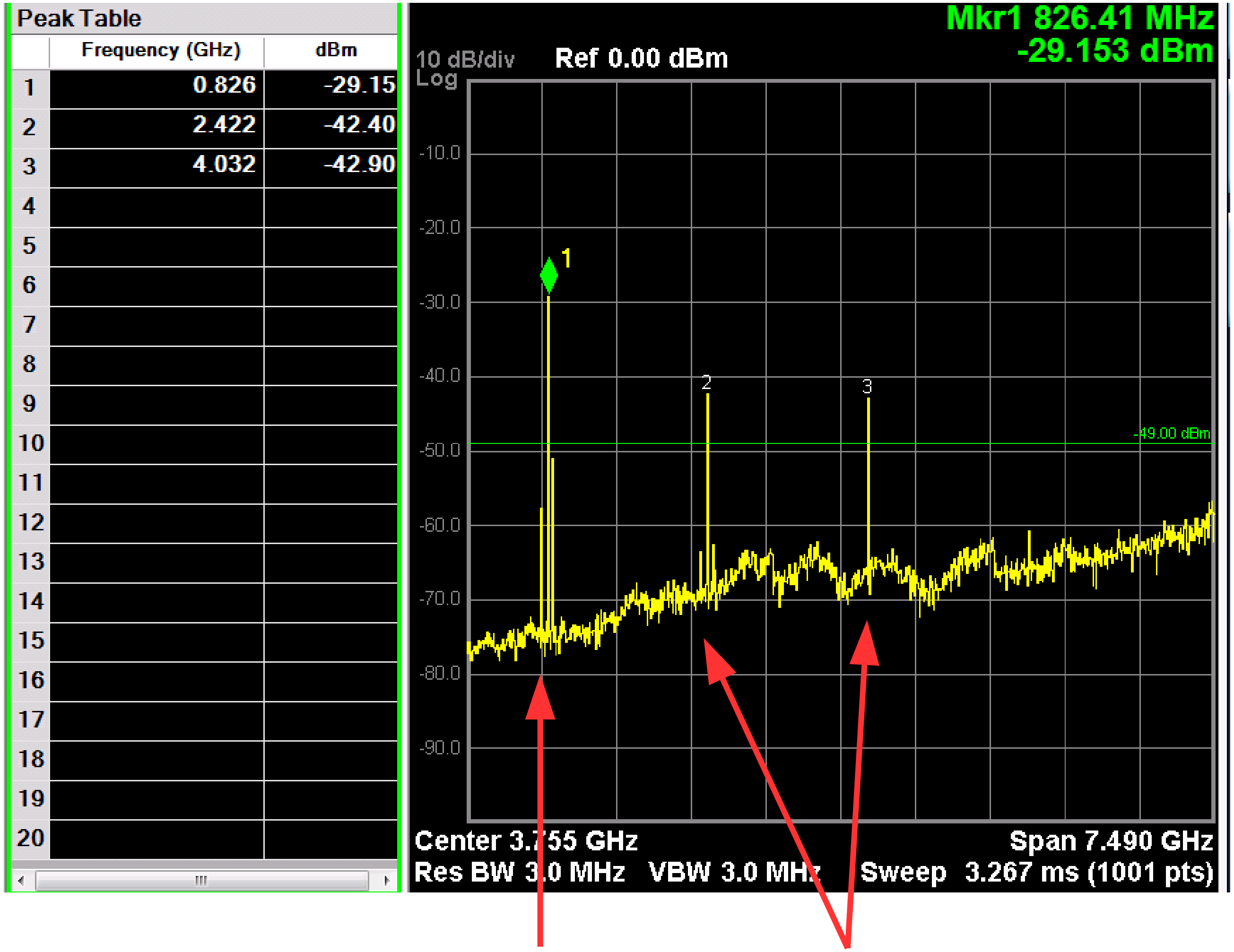
Fairwaves UmSITE

A production BTS based on SDR and osmo-bts





Don’t run PA without filters



Peak Table

	Frequency (GHz)	dBm
1	0.826	-29.15
2	2.422	-42.40
3	4.032	-42.90
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		

Mkr1 826.41 MHz
-29.153 dBm

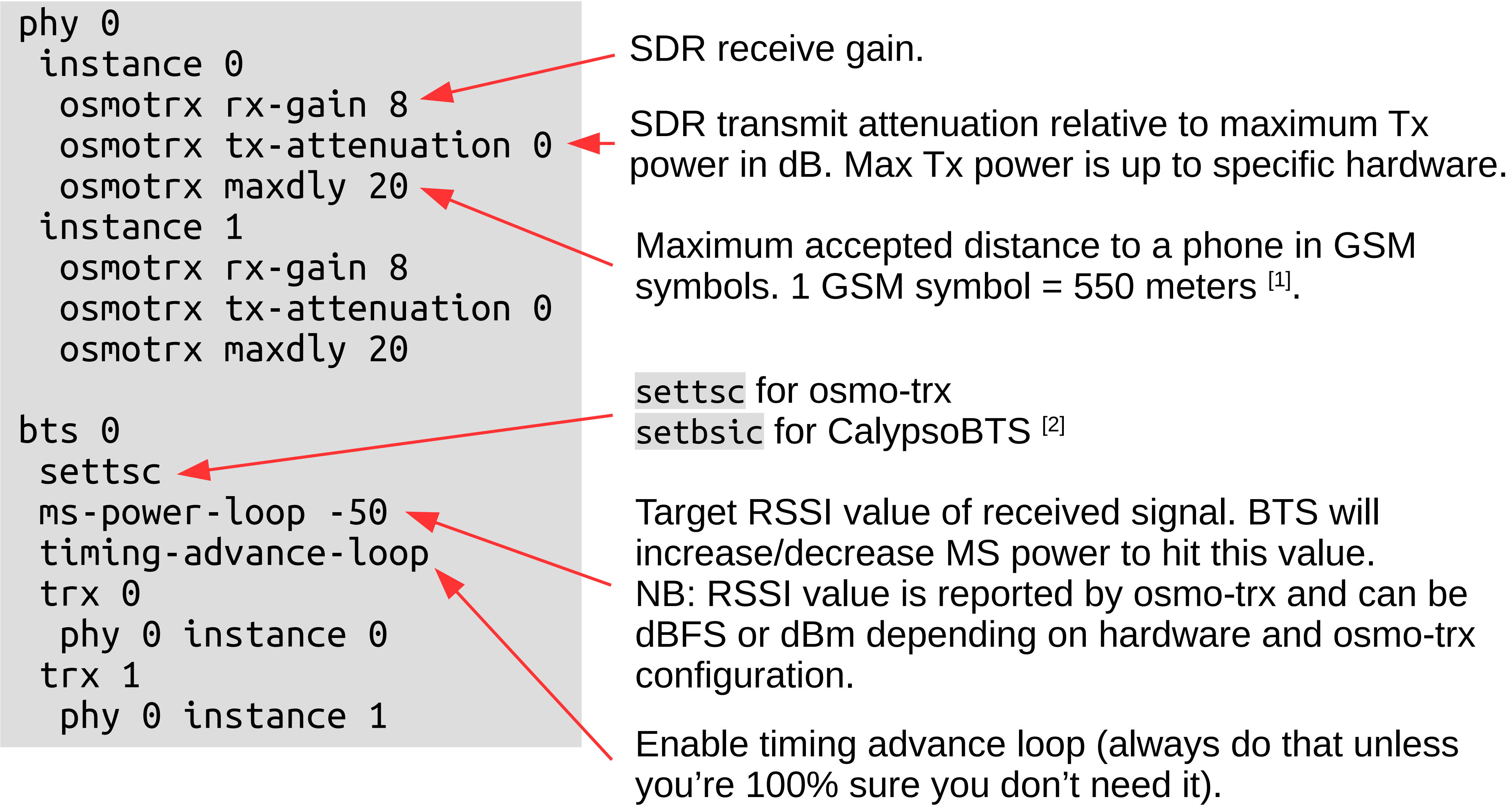
10 dB/div Ref 0.00 dBm
Log

Center 3.755 GHz Span 7.490 GHz
Res BW 3.0 MHz VBW 3.0 MHz Sweep 3.267 ms (1001 pts)

Useful signal Spurs

- Nothing is ideal
- SDR signal always has spurs
- They must be filtered to avoid harmful interference
- E.g. this GSM850 signal interferes with 2.4GHz band
- Again, please, never ever run high power without filters!

Configuring osmo-bts-trx



osmo-bts.conf:

```
phy 0
 instance 0
  osmotrx rx-gain 8
  osmotrx tx-attenuation 0
  osmotrx maxdly 20
 instance 1
  osmotrx rx-gain 8
  osmotrx tx-attenuation 0
  osmotrx maxdly 20

bts 0
 settsc
 ms-power-loop -50
 timing-advance-loop
 trx 0
  phy 0 instance 0
 trx 1
  phy 0 instance 1
```

SDR receive gain.

SDR transmit attenuation relative to maximum Tx power in dB. Max Tx power is up to specific hardware.

Maximum accepted distance to a phone in GSM symbols. 1 GSM symbol = 550 meters [1].

`settsc` for osmo-trx
`setbsic` for CalypsoBTS [2]

Target RSSI value of received signal. BTS will increase/decrease MS power to hit this value.
NB: RSSI value is reported by osmo-trx and can be dBFS or dBm depending on hardware and osmo-trx configuration.

Enable timing advance loop (always do that unless you're 100% sure you don't need it).

1. https://en.wikipedia.org/wiki/Timing_advance
2. https://osmocom.org/projects/baseband/wiki/CalypsoBTS

Configuring osmo-trx

- No VTY, no file configuration
- Only command line options
- Most important ones:
  - l   Logging level (EMERG, ALERT, CRT, ERR, WARNING, ...)
  - e   Enable EDGE receiver
  - m   Enable multi-ARFCN transceiver (default=disabled)
  - x   Enable external 10 MHz reference
  - g   Enable GPSDO reference
  - c   Number of ARFCN channels (default=1)
  - R   RSSI to dBm offset in dB (default=0)
- Refer to `` `osmo-trx -h` `` for more options

Configuring osmo-trx: startup

On startup `osmo-trx` shows its configuration, so you could check if you’re doing it right:

```
$ ./osmo-trx
linux; GNU C++ version 4.8.4; Boost_105400; UHD_003.010.001.001-release

opening configuration table from path :memory:
Config Settings
   Log Level............... NOTICE
   Device args.............
   TRX Base Port........... 5700
   TRX Address............. 127.0.0.1
   Channels................ 1
   Tx Samples-per-Symbol... 4
   Rx Samples-per-Symbol... 1
   EDGE support............ Disabled
   Reference............... Internal
   C0 Filler Table......... Disabled
   Multi-Carrier........... Disabled
   Diversity............... Disabled
   Tuning offset........... 0
   RSSI to dBm offset...... 0
   Swap channels........... 0
```

Configuring osmo-trx: examples

- UmTRX with EDGE:
  - osmo-trx -c 2 -R 38 -e
- USRP B200 single channel with GPSDO:
  - osmo-trx -g
- USRP B200 multi-channel with GPSDO:
  - osmo-trx -m -c 3 -g -e
- USRP B210 with GPSDO:
  - osmo-trx -c 2 -g