OsmoTRX - Feature #3337

osmo-trx: Clarify where RxGain parameter should be set

06/12/2018 05:03 PM - pespin

Status:	Resolved	**Start date:**	06/12/2018
Priority:	Normal	**Due date:**	
Assignee:	pespin	**% Done:**	100%
Category:	LimeSDR		
Target version:			
Spec Reference:			

Description

Right now, it seems we have a hardcoded RxGain in LMSDevice::stRxGain:

```
double LMSDevice::setRxGain(double dB, size_t chan)
{
    if (chan) {
        LOG(ALERT) << "Invalid channel " << chan;
        return 0.0;
    }

    dB = 34.0;
...
```

However, the current setup of osmo-bts-trx and osmo-trx indicates that this cfg should be set in VTY in osmo-bts-trx:

```
static void show_phy_inst_single(struct vty *vty, struct phy_instance *pinst)
{
...

    if (l1h->config.rxgain_valid)
        vty_out(vty, " rx-gain        : %d dB%s",
            l1h->config.rxgain, VTY_NEWLINE);
    else
        vty_out(vty, " rx-gain        : undefined%s", VTY_NEWLINE);
...
```

Which is then sent to osmo-trx via trx-CTRL protocol:

```
int l1if_provision_transceiver_trx(struct trx_l1h *l1h)
{
...
        /* after power on */
        if (l1h->config.rxgain_valid && !l1h->config.rxgain_sent) {
            trx_if_cmd_setrxgain(l1h, l1h->config.rxgain);
            l1h->config.rxgain_sent = 1;
        }
...
```

which is then send with "SETRXGAIN" command.

osmo-trx then receives it in Transceiver::driveControl() and sends it to radioIface and finally ro the radioDevice (like LMSDevice::setRxGain()).

However, RxGain seems to be quite attach to the device, so I'm wondering whether it makes sense to have a VTY cfg option in osmo-trx. I think we can still have them both: The one in VTY of osmo-trx is initially applied, and then if one is set in osmo-bts-trx.cfg it will be sent and override the one in osmo-trx (no need to modify anything in osmo-bts-trx, it already works like that).

This way we can provide sensible defaults for each device since we already have a per-defice .cfg file in osmo-trx/doc/examples/.

Feedback on this proposal is welcome.

History

#1 - 06/13/2018 08:22 PM - laforge

The problem with all those low-level parameters is that they require tons of tuning, and some of them even tuning at runtime.

As a SDR is only one fraction of the RF chain of an actual BTS, there are so many factors we don't know about, particularly what kind of filtering / LNA / PA there is between antenna and SDR. So whatever we do, I doubt there is ever a reasonable default for those values.

Normally, all of this should be abstracted away by the "Layer 0" of the BTS. OsmoBTS wants to

- receive absolute, calibrated dBm levels as RSSI
- transmit as absolute, calibrated dBm levels
- not have to bother with low-level and device/hardware specific bits, such as gain settings for rx/tx, calibration tables (or rather functions) that compensate the non-linearities for each arfcn/band/power-level/...
  - however, we do have hardware/bts specific code for osmo-bts-sysmo + lc15 to load a calibration table into hardware. That's fine, but we're not actually interpreting those calibration tables or need to apply any calibration
  - we do have some unused preparation for calibration tables for external PAs. That also makes sense, as you can attach an external PA to any type of BTS, whether it uses osmo-trx or not. You can use a proprietary nanoBTS and add a PA to it, and we somehow would want to handle such situations.

So in general I agree that it's wrong to have low-level bits like gain settings in OsmoBTS. However, this is the road taken so far, and I don't think there's anyone interested in funding related re-design. I think for now, let's keept it like it is (fixed gain settings in osmo-bts config) and make sure those settings are used across all hardware (uhd, libusrp, lms).

Before anyone can ever think of running a SDR-based (osmo-trx based) real BTS, he will have to add whatever code necessarry to manage all of this. But then, once again, I don't see anyone contributing in this area to develop large code modules like this from scratch.

The makers of actual BTS hadware would be the likely candidates to contribute this. Not sure how e.g. fairwaves are dealing with managing gain values and calibration tables in their products. I haven't seen anything on osmo-trx related to it?

#2 - 06/19/2018 08:42 AM - pespin

As a self reminder: Regarding LMS backend, there's a related ticket opened to ask for {Min,Max}RxGain in https://github.com/myriadrf/LimeSuite/issues/195

#3 - 06/19/2018 08:49 AM - pespin

- Status changed from New to Feedback

- % Done changed from 0 to 80

I added this commit to keep a good default for LMS and still be able to set desired value from osmo-bts:
https://gerrit.osmocom.org/#/c/osmo-trx/+/9678

I think this should be enough for now and we can then close this ticket.

#4 - 06/22/2018 10:00 AM - pespin

- Status changed from Feedback to Resolved

- % Done changed from 80 to 100

Merged, closing