OsmoTRX - Feature #4583

generate tx-power correlation table values for different sdr boards

06/05/2020 12:58 PM - roh

Status: Resolved  Start date: 06/05/2020
Priority: Normal  Due date:
Assignee: pespin  % Done: 100%
Category:  Target version:
Spec Reference:

Description

to be able to support the new NOMTXPOWER command its necessary that each 'bts type' can report sane values, thus some kind of 'correlation' is necessary.
real bts hardware has calibration tables for this, so calling it 'correlation' makes clear that we are not talking about exact values.
for the beginning we only do one measurement per band, and only sample a few values throughout the possible dynamic range
i have selected one arfcn per band:

<table>
<thead>
<tr>
<th>band</th>
<th>arfcn</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSM850</td>
<td>190</td>
</tr>
<tr>
<td>GSM900</td>
<td>35</td>
</tr>
<tr>
<td>DCS1800</td>
<td>698</td>
</tr>
<tr>
<td>PCS1900</td>
<td>661</td>
</tr>
</tbody>
</table>

Related issues:
Related to OsmoTRX - Bug #4468: "RSSI offset" default of 0 is not useful

Revision 056ce136 - 06/09/2020 09:43 AM - pespin
UHDDevice: Implement getNominalTxPower() based on TxFrequency

The table with nominal UHD Tx Gains and real transmit power is filled with values measured experimentally. More information can be found in OS#4583.

Related: OS#4583
Change-Id: lf7ef5bf95f4e4afe5864c0f051853aa38b9639eb

Revision 390db47b - 06/19/2020 08:35 PM - pespin
bsc: Allow setting negative nominal tx power through VTY

Some SDRs may provide tx power below 0 dBm on some bands.

Related: OS#4583
Change-Id: la53c30331bea23e219de4e1242cb8a5b357a2ae

Revision f68f19b1 - 06/25/2020 11:05 AM - pespin
LMSDevice: Compute TxGain on LimeSuite API based on expected Tx output power

Right now, according to a few measurements taken on LimeMicro devices, we expect the Tx Gain at UHD level to relate 1:1 with the slope in Tx output power given a specific band.

If more fine-grained results are wanted or some device doesn't follow a 1:1 slope relationship, functions TxGain2TxPower and TxPower2TxGain need to be adapted/improved.

This patch is basically doing the same thing as was done previously for UHDDevice in 992c9bd1cea410e2dd42ce7566299104b5648af.
Related: OS#4583
Change-Id: If154fe4d4cd118aa30ea43c22ee7119117b77da6

Revision f23d8358 - 06/27/2020 02:23 PM - pespin

**tx_power:** Support controlling BTS with nominal tx pwr < 0dBm

Related: OS#4583
Change-Id: l88d59d47837105d52e2b4d8b819511cd360c50a1

**History**

#1 - 06/05/2020 12:58 PM - roh
- Checklist item [ ] USRP B210 added

#2 - 06/05/2020 01:00 PM - roh

#4468 is about the same for RX

#3 - 06/05/2020 01:03 PM - laforge
- Checklist item [ ] LimeSDR-USB added
  Checklist item [ ] LimeSDR-mini added

#4 - 06/05/2020 01:03 PM - laforge
- Related to Bug #4468: "RSSI offset" default of 0 is not useful added

#5 - 06/05/2020 02:53 PM - roh

B210:

- osmtrx tx-attenuation 0 'max power'
  - Set TX gain to 89.75dB (asked for 89.75dB)
- osmtrx tx-attenuation 10
  - Set TX gain to 79.75dB (asked for 79.75dB)
- osmtrx tx-attenuation 20
  - Set TX gain to 69.75dB (asked for 69.75dB)
- osmtrx tx-attenuation 30
  - Set TX gain to 59.75dB (asked for 59.75dB)
- osmtrx tx-attenuation 40
  - Set TX gain to 49.75dB (asked for 49.75dB)
- osmtrx tx-attenuation 50
  - Set TX gain to 39.75dB (asked for 39.75dB)

850 - all PASS
- att 0 -- 13.3dBm
- att 10 -- 3.9dBm
- att 20 -- -5.6dBm
- att 30 -- -15.7dBm
- att 40 -- -25.7dBm
- att 50 -- -36.0dBm

900 - all PASS
- att 0 -- 13.3dBm
- att 10 -- 3.8dBm
- att 20 -- -1.8dBm
- att 30 -- -11.3dBm
- att 40 -- -22.3dBm
- att 50 -- -31.4dBm

1800 - all PASS
- att 0 -- 7.5dBm
- att 10 -- -1.8dBm
- att 20 -- -11.3dBm
- att 30 -- -22.3dBm
- att 40 -- -31.4dBm
- att 50 -- -42.5dBm

1900 - all PASS
- att 0 -- 7.7dBm
- att 10 -- -1.6dBm
I'm plotting stuff using gnuplot which does the linear regression for me and give me information about slope, etc.

Eg csv file (first column is real tx power, second is UHD's tx gain):

<table>
<thead>
<tr>
<th>Power</th>
<th>Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>13.3</td>
<td>89.75</td>
</tr>
<tr>
<td>3.9</td>
<td>79.75</td>
</tr>
<tr>
<td>-5.6</td>
<td>69.75</td>
</tr>
<tr>
<td>-15.7</td>
<td>59.75</td>
</tr>
<tr>
<td>-25.7</td>
<td>49.75</td>
</tr>
<tr>
<td>-36.0</td>
<td>39.75</td>
</tr>
</tbody>
</table>

Then I use this gnuplot script:

```gnuplot
f(x) = a*x + b
fit f(x) '/tmp/bla' u 1:2 via a, b
title_f(a,b) = sprintf('f(x) = \%.2fx + \%.2f', a, b)
plot "'/tmp/bla" u 1:2 w linespoints title 'uhd tx gain', f(x) t title_f(a,b), "'/tmp/bla" using 1:(column(2)-column(1)) with linespoints title 'cmp'
```

which generates attached png.

So according to gnuplot regression, in osmo-trx when we are asked to set a given tx power, in 850mhz the UHD TxGain we need to apply is:

```
hd_tx_gain = 1.01 * requested_tx_power + 75.86
```

So if all follow up more or less lineary, we can simply keep 2 default values per device and band: slope and offset, and make does configurable in VTY too.

#8 - 06/07/2020 10:07 PM - pespin
- File uhd_850mhz.png added

#9 - 06/08/2020 08:29 AM - laforge

Hi Pau,

you are thinking again way too complex here. Do we have Ettus or any related SDR user funding our work in trying to create calibration function abstractions? No.

Please KISS and simply use one fixed value per band here, like I have by now writing and talking about for several weeks.

#10 - 06/08/2020 02:08 PM - pespin

I have not applied the slope + offset implementation as requested. I simply assumed slope to be 1 for all bands and I'm simply using the nominal tx power and the tx gain set at that point and using that fixed offset when computing new gains/txpower:

remote: https://gerrit.osmocom.org/c/osmo-trx/+/18586 Transceiver: Implement TRXC cmd NOMTXPOWER
remote: https://gerrit.osmocom.org/c/osmo-trx/+/18721 UHDDevice: Implement getNominalTxPower() based on TxFrequency
remote: https://gerrit.osmocom.org/c/osmo-trx/+/18722 radioInterface: Operate on real Tx power attenuation rather than on device ...
remote: https://gerrit.osmocom.org/c/osmo-trx/+/18723 UHDDevice: Compute TxGain on UHD API based on expected Tx output power
I did some rewriting of some APIs between radioInterface and radioDevice to clearly differentiate between tx gain values used by low level APIs (like UHD or limesuite) and real output power. radioInterface here simply cares about real output power and lets each radioDevice handle internally the gain values/ranges to reach the requested output tx power.

Some bits are left there to easily extend in the future with more complex calibrations methods (like having a slope attribute per band as I mentioned before) or calibration tables or whatever.

#11 - 06/11/2020 02:45 PM - ipse

Just a quick question without looking into the patches. Is there a compatibility with the old semantic of specifying Tx/Rx gain corrections? Or is there a clear path to move to the new syntax? We're using this in our UmSITES to specify corrections to adjust for the PA/LNA we're using there and have calibrated values for that.

#12 - 06/11/2020 02:53 PM - pespin

First of all, these patches have nothing to do with Rx side, so just to be clear we leave that aside (it continues the same).

Can you share what do you mean with “old semantics”? How do you currently set Tx gain corrections?

With the new patches you can still overwrite the Nominal Tx power in osmo-bts-trx by using eg VTY command “nominal-tx-power 40” in the "trx 0" node.

The idea is that later on it will be possible to tweak nominal Tx value and calibration directly in osmo-trx using some VTY config there, and osmo-bts-trx will simply request it with NOMTXPOWER, but that's out of the scope of current features since no customer is paying for that, so we are only providing working measured values for our most usual cases.

#13 - 06/16/2020 02:17 PM - roh

- Checklist item [] LimeNet micro added

#14 - 06/17/2020 06:38 PM - roh

LimeSDR USB:

all measurements were done with an apu2 and the following versions:

- OsmoTRX version 1.2.0.39-1b3a
- OsmoBTS version 1.2.0.80-1851
- OsmoBSC version 1.6.0.123-c1a73

- Lime: Library version: v20.01.0-1

HW:

Device name: LimeSDR-USB
Expansion name: UNSUPPORTED
Firmware version: 4
Hardware version: 4
Protocol version: 1
Gateware version: 2
Gateware revision: 22
Gateware target: LimeSDR-USB
Serial number: 0x90726074d341c

4406 on TX1_1

osmo-trx-lms.cfg:

...  
    tx-sps 4  
    rx-sps 4  
    clock-ref external  
    rt-prio 18  
    chan 0  
    ! tx-path BAND1  
    ! tx-path BAND2  
    rx-path LNAW

measurements:
- osmotrx tx-attenuation 0 'max power'
  - Set TX gain to 73dB
- osmotrx tx-attenuation 10
  - Set TX gain to 63dB
- osmotrx tx-attenuation 20
  - Set TX gain to 53dB
- osmotrx tx-attenuation 30
  - Set TX gain to 43dB
- osmotrx tx-attenuation 40
  - Set TX gain to 33dB
- osmotrx tx-attenuation 50
  - Set TX gain to 23dB

- 850 - all PASS
  - att 0 -- 11.2dBm
  - att 10 -- 1.0dBm
  - att 20 -- -8.0dBm
  - att 30 -- -17.8dBm
  - att 40 -- -28.0dBm
  - att 50 -- -38.0dBm

- 900 - all PASS
  - att 0 -- 10.8dBm
  - att 10 -- 0.5dBm
  - att 20 -- -8.5dBm
  - att 30 -- -18.3dBm
  - att 40 -- -28.6dBm
  - att 50 -- -38.7dBm

- 1800 - known issues with max power. see #3341, otherwise PASS
  - att 0 -- 3.6dBm FAIL
  - att 10 -- -5.9dBm PASS
  - att 20 -- -14.0dBm PASS
  - att 30 -- -23.6dBm PASS
  - att 40 -- -33.8dBm PASS
  - att 50 -- -43.7dBm PASS

- 1900 - massive phase errors - does this need another connector? does this need band2?
  - att 0 -- 1.7dBm FAIL 24deg rms phase error
  - att 10 -- -8.1dBm FAIL 8.5deg rms phase error
  - att 20 -- -16.3dBm FAIL 8.5deg rms phase error
  - att 30 -- -26.0dBm FAIL 8.5deg rms phase error
  - att 40 -- -36.3dBm FAIL 8.5deg rms phase error
  - att 50 -- -46.1dBm FAIL 8.1deg rms phase error

#15 - 06/17/2020 06:38 PM - roh
- Checklist item [x] LimeSDR-USB set to Done

#16 - 06/17/2020 08:15 PM - pespin

As a side note, those attenuations are currently based on current MaxGain from LimeSuite, which for each Lime dev is (3rd field, type double):

```c
static const std::map<enum lms_dev_type, struct dev_desc> dev_param_map {     { LMS_DEV_SDR_USB,   { true,  true,  73.0, GSMRATE, MCBTS_SPACING, 8.9e-5, 7.9e-5, LMS_DEV_SDR_USB_PREFIX_NAME } },     { LMS_DEV_SDR_MINI,  { false, true,  66.0, GSMRATE, MCBTS_SPACING, 8.9e-5, 8.2e-5, LMS_DEV_SDR_MINI_PREFIX_NAME } },     { LMS_DEV_NET_MICRO, { true,  false, 71.0, GSMRATE, MCBTS_SPACING, 8.9e-5, 7.9e-5, LMS_DEV_NET_MICRO_PREFIX_NAME } },     { LMS_DEV_UNKNOWN,   { true,  true,  73.0, GSMRATE, MCBTS_SPACING, 8.9e-5, 7.9e-5, "UNKNOWN" } },    };
```

So for these values the attenuation applied to LimeSDR-USB MaxGain=73.0.

So AFAIU it makes sense to consider nominal power for band1800 as -5.9dBm instead of the higher value? Or could try to find something between att 0..10 which PASSES and we select that as nominal power?

#17 - 06/18/2020 03:27 PM - roh

LimeSDR USB:

for 1800 i see 'ok' values from 8dB attenuation on. 1900 should be simply not used before its fixed. (and i'm not sure how and if possible)

so far we did not have any band dependent limits for any lime platform, but maybe this makes sense?

73 for gsm850 and gsm900
65 for gsm1800
error out on pcs1900 for now.

#18 - 06/18/2020 04:33 PM - roh

LimeSDR-mini:

all measurements were done with an apu2 and the following versions:

- OsmoTRX version 1.2.0.39-1b3a
- OsmoBTS version 1.2.0.80-1851
- OsmoBSC version 1.6.0.123-c1a73

- Lime: Library version: v20.01.0-1

HW:

Device name: LimeSDR-Mini
Expansion name: UNSUPPORTED
Firmware version: 4
Hardware version: 2
Protocol version: 1
Gateware version: 1
Gateware revision: 30
Gateware target: LimeSDR-Mini
Serial number: 0x1d3b7aa1a9f5cc

4406 on TX

osmo-trx-lms.cfg:

```
...
  tx-sps 4
  rx-sps 4
  ! clock-ref external
  rt-prio 18
  chan 0
  ! tx-path BAND1
  tx-path BAND2
  rx-path LNAW
```

note: BAND2 is important, or you’ll loose ~16-20dB on the low bands (<1800)
for 1800 it doesn’t seem to matter, for 1900 BAND1 is correct

also important: force usb down to usb2. (simple usb extension cord) - usb3 will loose packets without end on the -mini busbridge.(does not work at all for me)

measurements:

- osmotr x-attenuation 0 ‘max power’
  - Set TX gain to 66dB
- osmotr x-attenuation 10
  - Set TX gain to 56dB
- osmotr x-attenuation 20
  - Set TX gain to 46dB
- osmotr x-attenuation 30
  - Set TX gain to 36dB
- osmotr x-attenuation 40
  - Set TX gain to 26dB
- osmotr x-attenuation 50
  - Set TX gain to 16dB

- 850 - all PASS, BAND2
  - att 0 -- 3.1dBm
  - att 10 -- -4.8dBm
  - att 20 -- -14.8dBm
  - att 30 -- -24.8dBm
  - att 40 -- -35.1dBm
  - att 50 -- -45.1dBm

- 900 - all PASS, BAND2
  - att 0 -- 2.8dBm
- Checklist item [x] LimeSDR-mini set to Done

**#20 - 06/18/2020 05:42 PM - roh**

LimeNet micro:

all measurements were done with an apu2 for the bsc and lnm the following versions:

- OsmoBSC version 1.6.0.123-c1a73 (on apu2)
- OsmoTRX version 1.1.1.29-e503
- OsmoBTS version 1.1.0.20-7683

- Lime: Library version: v19.04.0-1

HW:

```
Device name: LimeNET-Micro
Expansion name: UNSUPPORTED
Firmware version: 6
Hardware version: 19
Protocol version: 1
Gateware version: 1
Gateware revision: 2
Gateware target: LimeNET-Micro
Serial number: 0x58399493f179
```

4406 on TRXA/TX

osmo-trx-lms.cfg:

```
... tx-sps 4
   rx-sps 4
   rt-prio 18
   chan 0
   tx-path BAND1
   rx-path LNAH
   ! tx-path BAND2
   ! rx-path LNAL
```

note: BAND1/BAND2 seems not important at all, atleast i cannot see any difference in phase-errors or signal power on this device.
in the high bands it helps reducing the power for 1800 by 6dB.
1900 should be disabled like for limesdr USB

measurements:

- osmotr tx-attenuation 0 'max power'
  - Set TX gain to 71dB

---

05/10/2021
- osmotrx tx-attenuation 10
  - Set TX gain to 61dB
- osmotrx tx-attenuation 20
  - Set TX gain to 51dB
- osmotrx tx-attenuation 30
  - Set TX gain to 41dB
- osmotrx tx-attenuation 40
  - Set TX gain to 31dB
- osmotrx tx-attenuation 50
  - Set TX gain to 21dB

- 850 - all PASS, BAND1/BAND2
  - att 0 -- 6.8dBm
  - att 10 -- -2.3dBm
  - att 20 -- -11.3dBm
  - att 30 -- -21.3dBm
  - att 40 -- -31.6dBm
  - att 50 -- -41.6dBm

- 900 - all PASS, BAND1/BAND2
  - att 0 -- 6.8dBm
  - att 10 -- -2.4dBm
  - att 20 -- -11.4dBm
  - att 30 -- -21.4dBm
  - att 40 -- -31.8dBm
  - att 50 -- -41.7dBm

- 1800 - most PASS, BAND1/BAND2
  - att 0 -- -4.4dBm FAIL rms phase errors ~10°
  - att 6 -- -10.5dBm PASS
  - att 10 -- -13.0dBm PASS
  - att 20 -- -21.6dBm PASS
  - att 30 -- -31.5dBm PASS
  - att 40 -- -41.8dBm PASS
  - att 50 -- -51.5dBm PASS

- 1900 - all FAIL, BAND1/BAND2 rms phase errors >23°
  - att 0 -- -6.3dBm
  - att 10 -- -15.0dBm
  - att 20 -- -23.5dBm
  - att 30 -- -33.5dBm
  - att 40 -- -43.7dBm
  - att 50 -- -53.5dBm

---

#21 - 06/18/2020 05:42 PM - roh
- Checklist item [x] LimeNet micro set to Done

#22 - 06/18/2020 05:43 PM - roh
- % Done changed from 0 to 100

#23 - 06/18/2020 05:43 PM - roh
- Status changed from New to Resolved

#24 - 06/18/2020 06:13 PM - pespin
- Status changed from Resolved to In Progress
- Assignee changed from roh to pespin

Re-assigning task to me since I need to add those values to software ;)

#25 - 06/19/2020 04:17 PM - pespin
- Status changed from In Progress to Feedback

Submitted patch to apply Nominal Tx power per device and band in LMSDevice here:
remote: https://gerrit.osmocom.org/c/osmo-trx/+/18918 LMSDevice: Compute TxGain on LimeSuite API based on expected Tx output power
remote: https://gerrit.osmocom.org/c/osmo-trx/+/18919 Drop old TxGain APIs from parent radioDevice abstract class

I also had to submit a bunch of patches in osmo-trx, osmo-bts-trx and osmo-bsc to allow working with nominal powers less than 0 dBm (negative), since some bands in Lime devices provide such nominal tx power values.
#26 - 07/07/2020 10:57 AM - pespin

- Status changed from Feedback to Resolved

Closing the ticket, this is implemented for the above requested devices.

Files

<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Date</th>
<th>Username</th>
</tr>
</thead>
<tbody>
<tr>
<td>uhd_850mhz.png</td>
<td>20.3 KB</td>
<td>06/07/2020</td>
<td>pespin</td>
</tr>
</tbody>
</table>