Tune Up procedures for Q2426.

This paper aims to explain the function of each radio parameters which are in the eeprom, the way to calculate each of them and the way to load them in the eeprom.

- **ThinDacValue** : (former THIN_DAC_INIT_VALUE in the file EEPROM.H). The value given to this parameter is loaded in the 10 bits AFC DAC when the mobile is powered on.
For more explanation about this parameter, have a look to the description of the parameter RoughDacValue below.

The global excursion of the thin DAC is about 30 ppm, so the slope is about - 0.03 ppm for one step. It means that when the value loaded in the thin DAC is increased by one, the frequency of the VCXO decrease by 0.03 ppm.

The thin DAC is used by the software to controle the frequency of the VCXO when the mobile is synchronised on the network to compensate doppler and temperature drift.

- **RoughDacValue** : (former CARTE_DAC_INIT_VALUE in the file EEPROM.H). The value given to this parameter is loaded in the 8 bits DAC (GTI radio) 3 bits DAC (GISMO radio) when the mobile is powered on.

This value should be such that the VCXO accuracy is better than 0.1 ppm at a temperature of 20 ° C when both ThinDacValue and RoughDacValue are loaded.

The global excursion of the Rough DAC is about 80 ppm, so the slope is about + 0.3 ppm by step for GTI radio and + 10 ppm by step for GISMO radio. It means that when the value loaded in the rough DAC is increased by one the frequency of the VCXO increase by 0.3 ppm (GTI radio) 10 ppm (GISMO radio).

The rough DAC is used by the software to compensate aging drift of VCXO.

A good way to calculate the value to give to the parameters ThinDacValue and RoughDacValue, is to measure by any mean the frequency of the VCXO while the value loaded in the two DAC are half the excursion (512 for the thin DAC, 128 for the rough DAC for GTI radio, 4 for the rough DAC for GISMO radio).
Then, calculate first the best value to give to RoughDacValue and then the best value to give to the ThinDacValue.
As the slope of the rough DAC and of the thin DAC may vary around the value given, the whole process should be lead again one or two time.

e.g : the measure of the VCXO give 13.0002 MHz
$\Delta f/f$ = 200/13 E6 = +15.4 ppm

For WISMO radio : The value loaded in the rough DAC should be decreased by (15.4 + 5) / 10 = 2
So RoughDacValue = 4 - 2 = 2.
Then there is (15.4 - 10 x 2) = - 4.6 ppm left to be compensated with the thin DAC.
To do so, the value loaded in the thin DAC should be decreased by 4.6 / 0.03 = 153.

So ThinDacValue = 512 - 180 = 358.

- **PwrVsAgcSlope** : (former D_PWR_VS_AGC_SLOPE in the file EEPROM.H). This parameter is proportionate to the slope of the straight line describing the power on the antenna expressed in dBm versus the Agc value loaded in the 8 bits Agc DAC.
To calculate the value to give to this parameter, multiply the previous slope by -128.

Notice that PwrVsAgcSlope = 128x128 / AgcVsPwrSlope.

e.g : if the slope is -0.4 dBm by Agc step, PwrVsAgcSlope = -128 x (-0.4) = 51.

Notice that the straight line describing the power on the antenna versus the Agc must be built when the level on I and Q ADC is constant whatever the power on the antenna is.

To do so, the nominal software can be used when synchronisation is possible. To make the synchronisation possible, the parameters ThinDacValue and RoughDacValue should be calculated as described and likely values should be given to the parameters PwrVsAgcSlope (e.g 51), AgcVsPwrSlope (e.g 320), BoardGain (e.g 40) and VoltageOrder (e.g 90).

Then one can try to synchronise on a test set (e.g HP 8922) and tune the power and then read the Agc order for several value of power. The Agc order can be read in the top of the PC screen while using HAPC software. The right field is the second Agc field of the second line of the parameters displayed. It is just on the left of the Mkr field. Of course the value of the power on the antenna must be read on the test set and not in the Pwr field.

Notice that the Mkr field should be rather constant while the power increase or decrease because of the software control even if the parameters are not good, however, the calculation may be improved by leading again the whole process using, the values obtained for PwrVsAgcSlope and AgcVsPwrSlope.

- **AgcVsPwrSlope** : (former D_AGC_VS_PWR_SLOPE in the file EEPROM.H). This parameter is proportionate to the slope of the straight line describing the Agc value loaded in the 8 bits Agc DAC versus the power on the antenna expressed in dBm.
To calculate the value to give to this parameter, multiply the previous slope by -128.

e.g : if the slope is -2.5 Agc step by dBm , AgcVsPwrSlope = -128 x (-2.5) = 320.

Notice that AgcVsPwrSlope = 128x128 / PwrVsAgcSlope.

See above, description of the PwrVsAgcSlope.

- **VoltageOrder** : (Former VOLTAGE_ORDER_VALUE in the file EEPROM.H). This parameter is
expressed in step of Agc DAC. It allows to adjust the voltage on the I and Q kernel DAC egaliser entrance.
The voltage should be about 1 V pp. If VoltageOrder is increased by one, the voltage on I and Q increased by about 0.05 V.
To calculate the right value, give a likely value to this parameter, and measure the voltage when the mobile is synchronised (see the process described for the calculation of the parameter PwrVsAgcSlope). Then measure the voltage on the kernel DAC egaliser entrance and adjust the value of VoltageOrder until the right voltage is obtained.

Notice that when a voltage of 1 Vpp on I and Q kernel entrance is obtained, the parameter Mkr displayed on the screen while using the HAPC software should be about 20. If not, there certainly some hardware trouble.

- **BoardGain** : (Former BOARD_SPECIFIC_GAIN in the file EEPROM.H). This parameter is expressed in dB and corresponds to the gain of the radio reception channel.
To calculate this parameter, a mean value being given to BoardGain (e.g 40), try to synchronise on a test set (see the process described for the calculation of the parameter PwrVsAgcSlope).
When synchronised, read P1 the value in the field Pwr (HAPC software) and P0 the power level of the test set. Then the value to be given to BoardGain corresponds to the difference between P0 and P1 if there is no signal loss between the test set and the mobile.

e.g : The value given to BoardGain is 40.
The value in Pwr field is 56. It corresponds to a power of P1=-56 dBm.
The power level of the test set is P0 = -50 dBm.
Assume a signal loss L = -1 dBm.

Then BoardGain should be decreased by -P1+ (P0 + L) = 5 dB.
So BoardGain = 40 - 5 = 35 dB.

Notice that as soon as BoardGain is given a likely value (e.g 40 dB) the Agc loop works. BoardGain is only usefull to get a good power measurement.

- **TxPwrRampNbPts** : number of steps of the Hamming power ramping. Up to now, 21 steps for GISMO and GTI V1 radio and 23 steps for GTI V3 radio.
See below the description of the ParamPwRamp parameters.

- **ParamPwRamp** : (former structure ParameterPowerRamp in the file EEPROM.H). The values following describe the Tx power level scheme for each of the 11 levels, level 5 to level 15 (GSM phase 1) with 4 values for each level : PowerMin, PowerMax, DebNMin and DebCourbe.

The ramp up is constituted by 32 values each being loaded every ¼ bit in the kernel TxPower DAC after the beginning of the ramp up. The ramp down is constituted by the same 32 values taken in an inverted order.

The values are calculated by the software as following :

- The DebNMin first values are 0. Notice that DebNMin must be greater than zero otherwise a value different from 0 could be left in the TxPower DAC after the ramp down.

- The n values between DebNMin and DebCourbe are the PowerMin values (n = DebCourbe - DebNMin + 1). Notice that (DebCourbe + TxPwrRampNbPt) must be smaller than 31 otherwise PowerMax couldn't be reached.

- The TxPwrRampNbPts values following DebCourbe are obtained from a Hamming process between PowerMin and PowerMax.

- The value for the (32 - DebNMin - n - TxPwrRampNbPts) = (31 - DebCourbe - TxPwrRampNbPts) steps remaining is PowerMax.

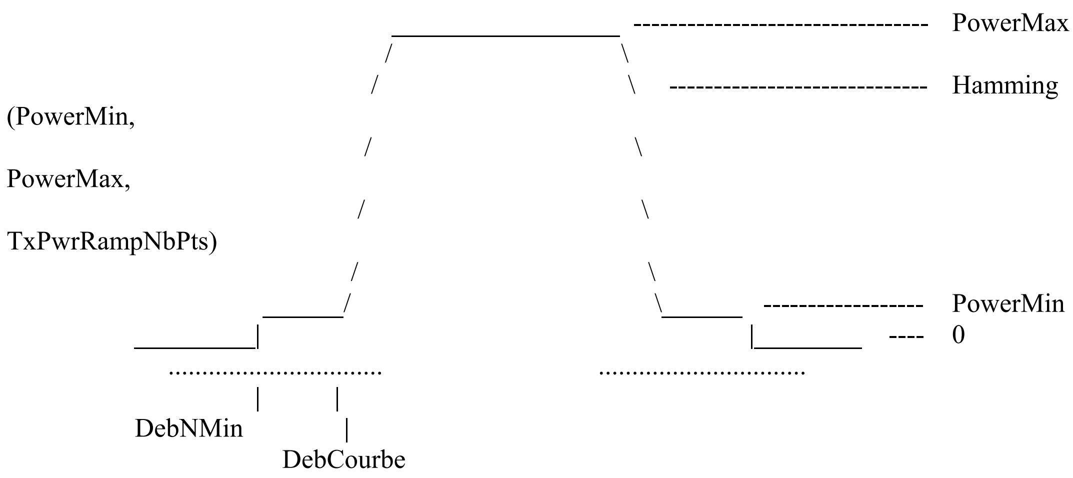

Fig 1 : power ramping scheme.

- **RtkState** : (new parameter). It describes a structured defined in e2p_defs.hi made of two u32 for the mailboxes state and one u32 for the tasks state. The mailboxes can be either local, remote or local and remote. Each mailbox state is described on two bits. The first one is the remote bit and the second the local one. The mailbox number is described in ascending bits order of the two u32.

The last u32 described the task state. One if the task is local and so has to be started and zero otherwise.

This parameter is automatically sent on the serial link when the target software begins.

- **DiagState** : (new parameter). This is an array of the trace levels allowed for every flow (one u32 per flow). See trace.ho to get the number of flows. The ‘d’ command of hapc is used to change the trace level.

- **RxSPErrorTres** : (new parameter). This is an u8 to set the threshold of error over which a TCH block is not sent to the vocoder. The typical value is 61 but this will be fine tuned in the future.

- **FacchTres** : (new parameter). This is an u8 to set the threshold of stealing flags over which a TCH block is rather decoded as a facch block than a speech block. The values can be from 4 to 8 and is curreny set to 8 but should be set to 4 in very soon, after tests.

- **E2pTstPattern** : (new parameter). This u16 parameter is used by the software to determine if the E2prom is virgin or not. This parameter should be written with the value 3C66 for 93C66 E2prom and 3C86 for 93C86 E2prom.
Notice that if the software is compiled with the switch __E2P_AUTO__, as soon as the right pattern is written in the E2prom, the E2prom software driver is able to work.
Otherwise, the switch corresponding to the right E2prom (E2P93C66 for 93C66 or E2P93C86 for 93C86) have to be set.

ooooo