# RTL8822CE
# RTL8822CU
# RTL8822CS
# COB
# MP FLOW

# R12

# Table of Contents



# List of Tables

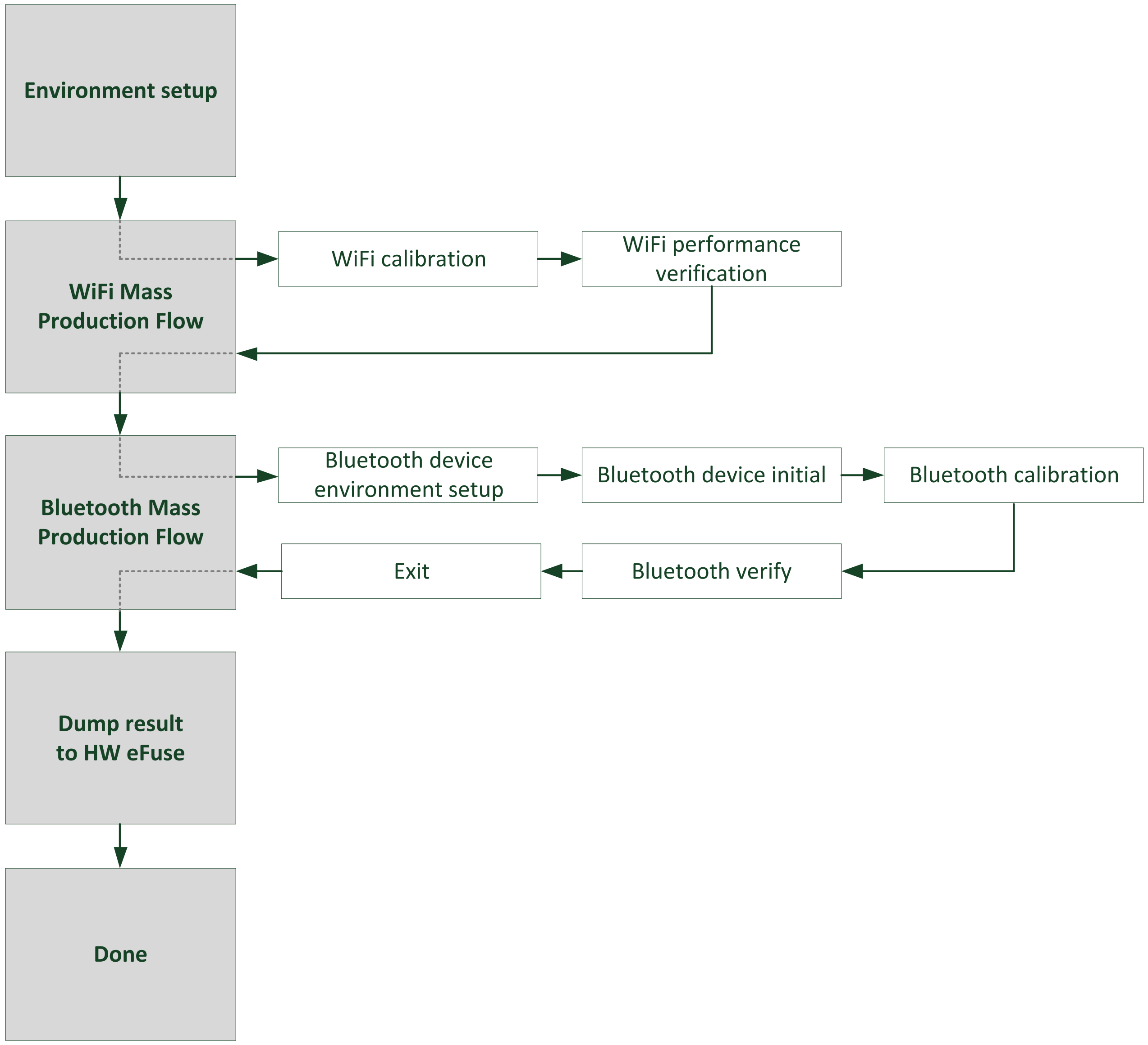
# List of Figures

Revision History:

| Revision | Date | Changes | Author |
|---|---|---|---|
| R12 | 2020/03/12 | Re-write from COB flow R11 | DeanKu |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

# 1. DUT MP Flow

The below diagram shows a global view of Mass-Production-Flow, please refer to the following sub-section to get the detail description of each step.



# 2. Environment setup

## 2.1. Test platform

The calibration flow described in following sections are based on WiFi tester. Realtek had qualified tester is listed below:

| Include 802.11ac Test | | Only 802.11a/b/g/n Test | |
|---|---|---|---|
| Vendor | Modal Name | Vendor | Modal Name |
| LitePoint | IQxel | LitePoint | IQFlex |
| Itest | WT-200 | LitePoint | IQView |
| NI | PXIe-5644R/5645R | LitePoint | IQnxn |
| Anritsu | MT8870A | Agilent | N4010A |
| Aeroflex | PXI 3000 Series | | |

The test environment setup is as below:

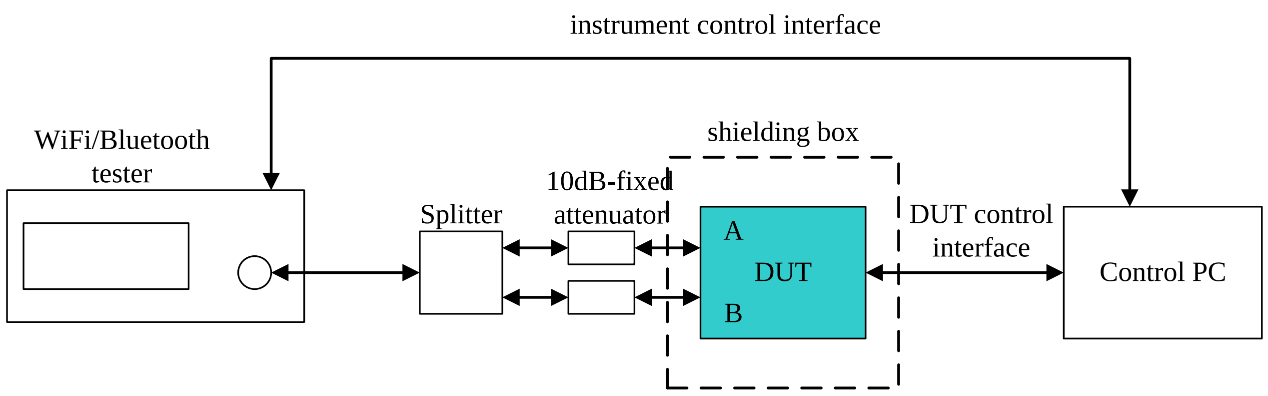



**Figure 1: Brief diagram about test environment setup of DUT**

***To reduce the mismatch effect between DUT and environment,**

**10dB-fixed attenuator is supposed to set closed to DUT as possible.**

## 2.2. CMD environment setup

**Step1 : Use USB cable to connect PC and Android/Linux devices as Figure 3.**

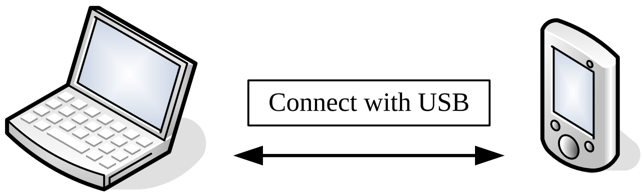



**Figure 2: Connect PC to Android/Linux device**

**Step2 : Use adb command as below:**

**adb devices**

How to check whether the Android/Linux devices were connected or not?

Figure 4 shows a result if the PC connects to Android/Linux devices successfully.


**Figure 3: Check the connection between PC & Android/Linux device**

**Step3 : Use adb command as below:**

**adb shell**

Figure 5 shows the CMD line under shell.


**Figure 4: Enter Android/Linux device command line**

# 3. WiFi Mass Production Flow

## 3.1. WiFi calibration

The WiFi calibration steps are shown as Figure 6:

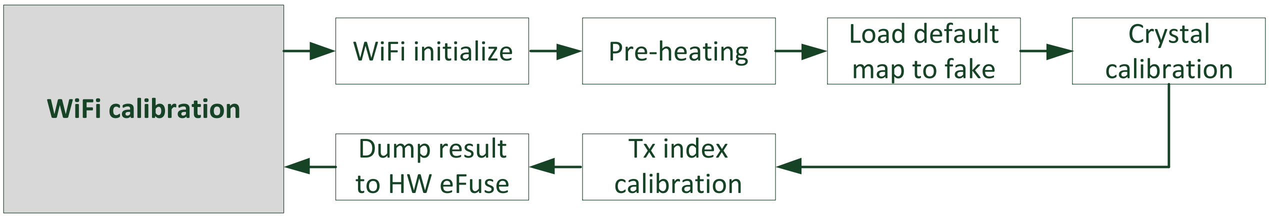



**Figure 5: WiFi calibration flow**

### 3.1.1. WiFi initialize

Refer to Section 2.2, PC can connect to Android/Linux devices with USB port and ADB command. The relative commands of DUT WiFi initializing is listed as below :

**remount**

**root**

**rmmod wlan**

**insmod 8822c.ko rtw_RFE_type=X**

**(delay 5 second)**

***If the efuse of RTL8822C was empty, please make sure your HW-Layout of board type and PA/LNA.**

**ifconfig wlan0 up**

**rtwpriv wlan0 mp_start**

### 3.1.2. Pre-heating

**Pre-heating is an one-time experiment to grab essential setup**, it made DUT closed to the usual temperature and improved the stability of mass production. It will pre-heat the DUT for seconds before mass production.

**Step 1 : Read and record thermal value $THER_{RX}$ at steady Rx status which lasts 5 minutes at room temperature.**

**rtwpriv wlan0 mp_start** **//Enter WiFi MP mode**

Wait for 5 minute.

**rtwpriv wlan0 mp_ther** **//Get the thermal value $THER_{RX}$**

Record the $THER_{RX}$ value.

**Step 2 : Use MP HW Tx to do the following pre-heating operation.**

**rtwpriv wlan0 6 1 a HTMCS7 1 TxPacketInterval TxPacketLength**

**//Supposed initial value of Tx TxPacketInterval = 100**

**//Supposed initial value of Tx TxPacketLength = 1000**

**rtwpriv wlan0 mp_ther** **//Get the heated thermal value $THER_{Heating}$**

**Step 3 : To figure out pre-heat settings (TxPacketInterval/TxPacketLength/$TIME_{Heating}$).**

Trim the TxPacketInterval or TxPacketLength, then do HW Tx and record how long does the DUT be heated to **$THER_{RX}$** temperature. If **$THER_{Heating}$ - $THER_{Rx}$ $\leqq$ 3**, means DUT is heated to desired temperature. The time-spent records as **$TIME_{Heating}$**.

Use these settings (TxPacketInterval/TxPacketLength/**$TIME_{Heating}$**) as pre-heating setup.

**Pre-heating DUT before calibration by the found settings set :**

Do HW Tx **rtwpriv wlan0 6 1 a HTMCS7 1 TxPacketInterval TxPacketLength**
Last for **$TIME_{Heating}$.** So that DUT can be pre-heated to desired thermal range.

**Example :**

**rtwpriv wlan0 mp_start** **//Enter WiFi MP mode**

Wait for 5 minute.

**rtwpriv wlan0 mp_ther** **//MP feedback the thermal value = 2d**

Record the $\mathbf{THER_{RX}}$ = 2d

**rtwpriv wlan0 6 1 a HTMCS7 1 200 1000**

**Example :** Use HW Tx, HT40 HTMCS7 channel 6, TxPacketInterval = 200, TxPacketLength = 1000 to pre-heat DUT.

5 seconds past.

**rtwpriv wlan0 mp_ther** **//MP feedback the thermal value = 2e**

Record the $\mathbf{THER_{Heating}}$ = 2e

2e ($\mathbf{THER_{Heating}}$) – 2d ($\mathbf{THER_{RX}}$) ≦ 3,DUT is heated to desired thermal range!

(5 seconds is just a sample value here. In the operation, please do HW Tx and observe the $\mathbf{THER_{Heating}}$ variance. How long does the DUT be heated to the thermal range.)

Finally, get a pre-heating settings set :

**rtwpriv wlan0 6 1 a HTMCS7 1 200 1000**

Heating time-spent is $\mathbf{TIME_{Heating}}$ = **5** seconds.

**DUT can be heated to desired thermal rang by**
**HW Tx(rtwpriv wlan0 6 1 a HTMCS7 1 200 1000**) for 5 seconds ($\mathbf{TIME_{Heating}}$).

**rtwpriv wlan0 stop** **//Stop HW Tx**

**The pre-heating settings were found.**
**Please pre-heat DUT before calibration.**

### 3.1.3. Load default map to fake map from eFuse/Mask map files.

Load external eFuse map file to driver fake map

**rtwpriv wlan0 efuse_file /../../efuse.map**

Load external mask map file to driver fake map :

**rtwpriv wlan0 efuse_mask /xx/xx/xxmask.txt**

### 3.1.4. Crystal calibration

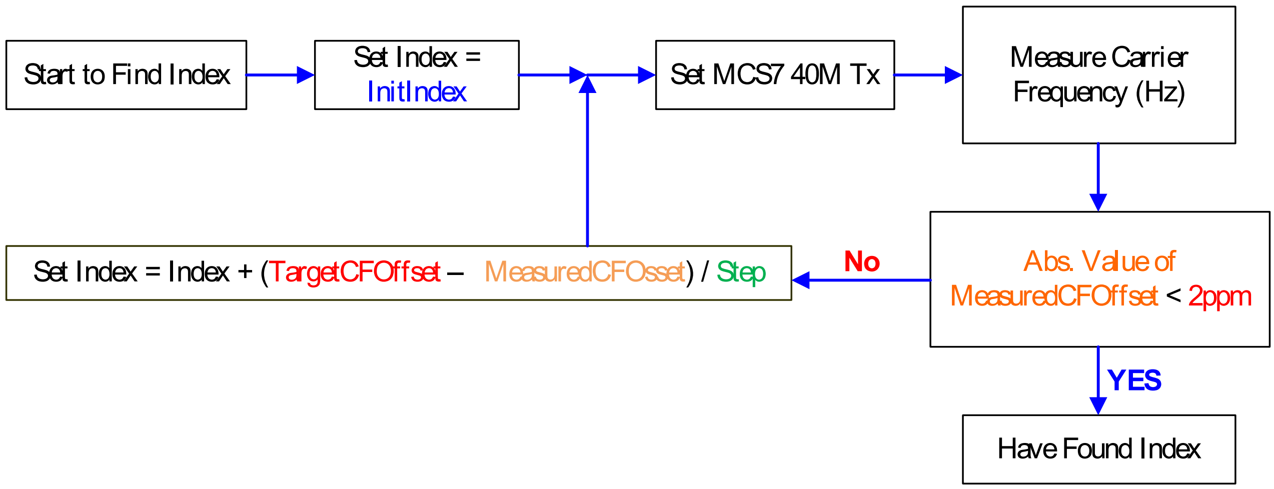



**Figure 6: Finding Crystal Cap. Index Flow**

InitIndex: the default value is 0x40. Index range is 0x0 to 0x7F.
MeasuredCFOffset: Carrier frequency measured by instrument - ideal carrier frequency
target range Abs. Value of 2ppm in 2.4GHz band is about 10KHz(±5KHz).
TargetCFOffset: generally is 0 ppm.
Step: The value will be different from modules dominated by external capacitor beside the crystal, so it must be able to modify simply in the initial file of test program. By experience, the value variation is about +2 ~ +3KHz. The value should be checked by hardware RD and be filled in the setup file of test program. The plus symbol means that the crystal Cap. index and carrier frequency is positive-dependent (The larger index is relative to major carrier frequency).

**Command is as below:**

**Step 1 :**

**rtwpriv wlan0 mp_setrfpath 1**

**//WiFi MP driver will set RF path automatically, too.**

**rtwpriv wlan0 7 1 a HTMCS7 1 2000** **//start HW Tx**

**Step 2 :**

1. To measure frequency error (ppm), Freq_Err.
2. Freq_Err should be limited to ±2ppm.
3. If Freq_Err is between ±2ppm, it means the Index_cry is correct.
4. If the Freq_Err is out of ±2ppm, try the algorithm below to find next index_cry_next until

   Freq_Err is between ±2ppm.

$$\textit{index\_cry\_next} = \textit{Index\_cry} - \frac{Freq\_Err \times 2442}{2500}$$

**Example :** Set Index_cry =32, measured Freq_Err = -23.49ppm. How to find next index ?

$$\textit{The next index is index\_cry\_next} = 32 - \frac{(-23.49) \times 2442}{2500} \fallingdotseq 9$$

**Step3 :**

The index_cry_next must be **rounded to an integer**, and use the following command to update index:

**rtwpriv wlan0 mp_phypara xcap=index_cry_next**

**Example :**

**rtwpriv wlan0 mp_phypara xcap=9** **//update crystal calibration value to MP**

Repeat **Step2, Step3** until the measured Freq_Err is limited to ±2ppm and note down **Index_cry_ok**. And **Index_cry_ok** should be updated to eFuse after calibration.

**Step 4 : After measurement, stop HW Tx.**

**rtwpriv wlan0 stop** **//stop HW Tx**

**Crystal calibration eFuse offset and notice :**

The values must be well-calibrated and filled in the correct eFuse location.

| Crystal Calibration | 0xB9, 0x110, 0x111 |
|---|---|

**Table 1: Crystal calibration offset in eFuse**

1. **0xB9, 0x110, 0x111 should be the same value as each other.**
2. **For the situation of K-free flow ,the eFuse content may be loaded from an external file in the system. These xtal-related-offset must be written in the HW eFuse even the eFuse-content could be set from an external file.**

## 3.1.5. Tx index calibration and thermal

The following flow will show the Tx index calibration flow, criterion, and sample command.

### 3.1.5.1. Tx index location in eFuse

There are the Tx index for each channel group with PHY data rate.

The Tx index value and thermal value should be calibrated and updated to eFuse.

| Tx index location in EFuse of Antenna A | | | | | | |
|---|---|---|---|---|---|---|
| | Group 1<br>CH1 – CH2 | Group 2<br>CH3 – CH5 | Group 3<br>CH6 – CH8 | Group 4<br>CH9 – CH11 | Group 5<br>CH12 – CH13 | Group 6<br>CH14 |
| MCS7<br>B40 | 0x16[7:0] | 0x17[7:0] | 0x18[7:0] | 0x19[7:0] | 0x1A[7:0] | |
| CCK | 0x10[7:0] | 0x11[7:0] | 0x12[7:0] | 0x13[7:0] | 0x14[7:0] | 0x15[7:0] |
| | Group 7<br>CH36 – CH40 | Group 8<br>CH44 – CH48 | Group 9<br>CH52 – CH56 | Group 10<br>CH60 – CH64 | | |
| MCS7<br>B40 | 0x22[7:0] | 0x23[7:0] | 0x24[7:0] | 0x25[7:0] | | |
| | Group 11<br>CH100 –<br>CH104 | Group 12<br>CH108 –<br>CH112 | Group 13<br>CH116 –<br>CH120 | Group 14<br>CH124 –<br>CH128 | Group 15<br>CH132 –<br>CH136 | Group 16<br>CH140 –<br>CH144 |
| MCS7<br>B40 | 0x26[7:0] | 0x27[7:0] | 0x28[7:0] | 0x29[7:0] | 0x2A[7:0] | 0x2B[7:0] |
| | Group 17<br>CH149 –<br>CH153 | Group 18<br>CH157 –<br>CH161 | Group 19<br>CH165 –<br>CH169 | Group 20<br>CH173 –<br>CH177 | | |
| MCS7<br>B40 | 0x2C[7:0] | 0x2D[7:0] | 0x2E[7:0] | 0x2F[7:0] | | |

| Tx index location in EFuse of Antenna B | | | | | | |
|---|---|---|---|---|---|---|
| | **Group 1**<br>**CH1 – CH2** | **Group 2**<br>**CH3 – CH5** | **Group 3**<br>**CH6 – CH8** | **Group 4**<br>**CH9 – CH11** | **Group 5**<br>**CH12 – CH13** | **Group 6**<br>**CH14** |
| **MCS7 B40** | 0x40[7:0] | 0x41[7:0] | 0x42[7:0] | 0x43[7:0] | 0x44[7:0] | |
| **CCK** | 0x3A[7:0] | 0x3B[7:0] | 0x3C[7:0] | 0x3D[7:0] | 0x3E[7:0] | 0x3F[7:0] |
| | **Group 7**<br>**CH36 – CH40** | **Group 8**<br>**CH44 – CH48** | **Group 9**<br>**CH52 – CH56** | **Group 10**<br>**CH60 – CH64** | | |
| **MCS7 B40** | 0x4C[7:0] | 0x4D[7:0] | 0x4E[7:0] | 0x4F[7:0] | | |
| | **Group 11**<br>**CH100 – CH104** | **Group 12**<br>**CH108 – CH112** | **Group 13**<br>**CH116 – CH120** | **Group 14**<br>**CH124 – CH128** | **Group 15**<br>**CH132 – CH136** | **Group 16**<br>**CH140 – CH144** |
| **MCS7 B40** | 0x50[7:0] | 0x51[7:0] | 0x52[7:0] | 0x53[7:0] | 0x54[7:0] | 0x55[7:0] |
| | **Group 17**<br>**CH149 – CH153** | **Group 18**<br>**CH157 – CH161** | **Group 19**<br>**CH165 – CH169** | **Group 20**<br>**CH173 – CH177** | | |
| **MCS7 B40** | 0x56[7:0] | 0x57[7:0] | 0x58[7:0] | 0x59[7:0] | | |

| | |
|---|---|
| **Thermal Meter PathA** | 0xD0[7:0] |
| **Thermal Meter PathB** | 0xD1[7:0] |

**Table 2: Tx index and Thermal meter offset in eFuse**

### 3.1.5.2. Defined target power

According to EMI/EMC regulatory IEEE TX EVM / Spectrum Mask requirement, the target power of each channel group and PHY data rate can be defined.

Assuming that all the channels have the same target power for each PHY data rate, the recommended target power is listed as below :

| Data Rate | MCS9-Nss1-B80 | MCS7-B40 | MCS7-B20 | 54M | CCK |
|---|---|---|---|---|---|
| Target Power 2G | | 16dBm | 16dBm | 17dBm | 18dBm |
| Target Power 5G | 14dBm | 16dBm | 16dBm | 17dBm | |

**Table 3: The default target power**

The target power in the table is RTK default value. The value can be defined by yourself.
Power-by-rate table and target power must be the same value.

***If there is any concern, or want to know the more details, please contact Realtek FAE.**

### 3.1.5.3. Tx index calibration flow

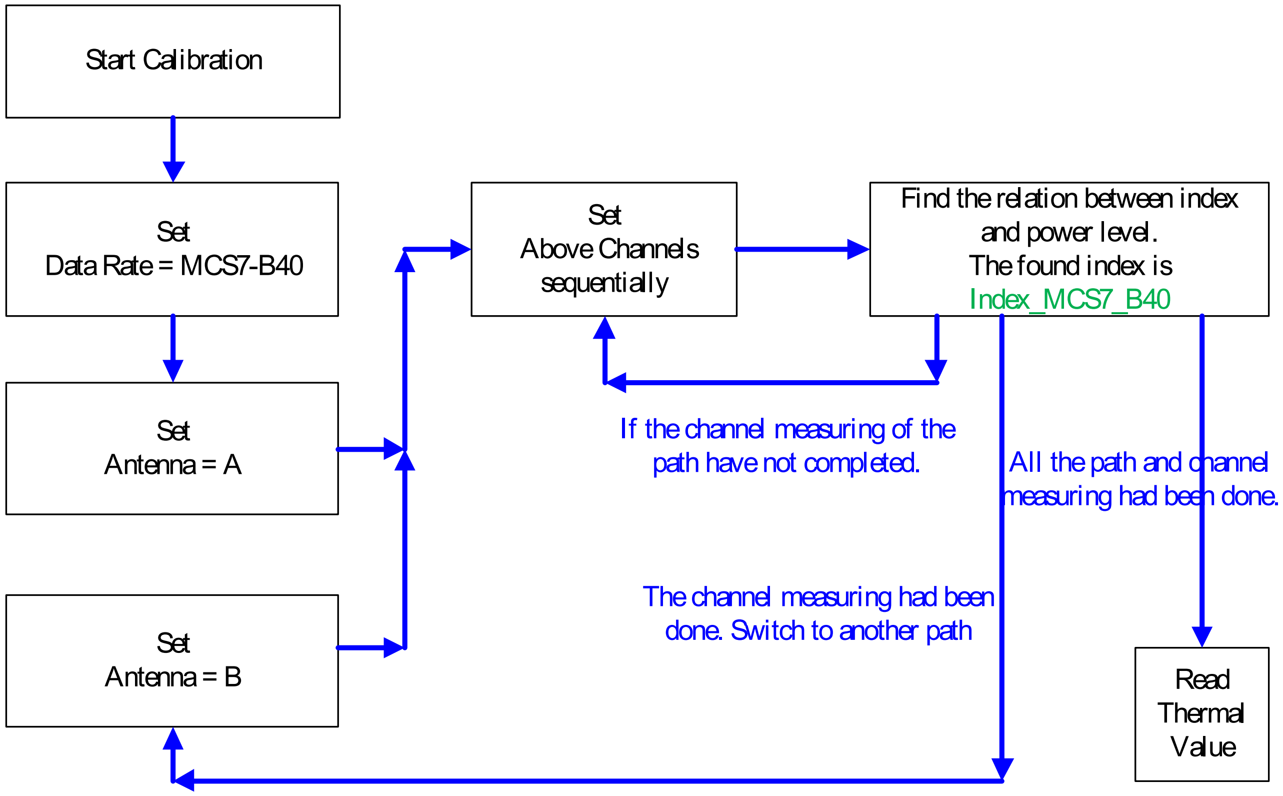



**Figure 7: Tx calibration flow**

The flow to find each index is shown as below:

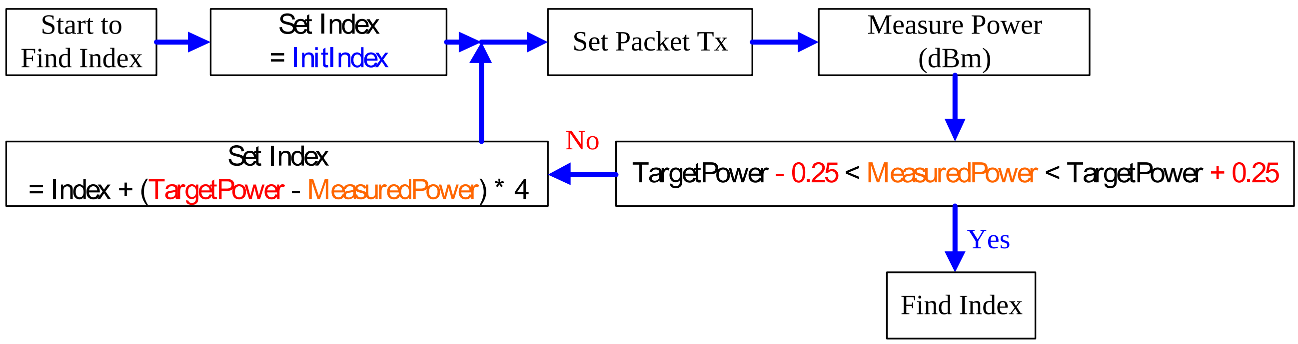



**Figure 8: Finding Tx index flow**

**Init Index :** The power value of **init index should be closed to target power**.

#### 3.1.5.4. Tx index calibration command and sample

**Step 1 : Switch the Tx power tracking state to the Tx index calibration state.**

**insmod 8822c.ko rtw_RFE_type=X**

Tx index for each channel group with PHY data rate in path a, path b should be calibrated.

The following command **take " Path A, Channel 7, HT40, HTMCS7" as example**.

**Step 2 : MP start and enter Tx index calibration mode.**

**rtwpriv wlan0 mp_start**

**rtwpriv wlan0 7 1 a HTMCS7 1 2000 //start HW Tx**

***About HW Tx function parameters detail, please refer to appendix.**

**Step 3 : Make Tx power to fit target power by adjusting Tx power index.**

**rtwpriv wlan0 mp_txpower patha=63,pathb=0 //set a Tx power index 63**

Target power values were defined in section 3.5.2.
The variation between defined Tx target power and measured Tx power should be adjusted to ±0.25dB.

**Defined Tx target power – Measured Tx power = ± 0.25 dB**

The sample command uses the index 63, it is the center index.

**Step 4 : Record the Tx index value after measured power had fit target power.**

Record the Tx index value. Find out all the Tx index value for each channel group and each PHY data rate. **There is a recommended interpolation method which described at section 3.1.5.4.1**

The Tx index value should be updated to HW eFuse.

**Step 5 : Before measuring the next channel, stop HW Tx.**

**rtwpriv wlan0 stop //Stop HW packet Tx**

#### 3.1.5.4.1. Interpolation for finding Tx index

Theoretically, we need to measure all value defined in eFuse to calibrate the Tx power level. But since it needs too much time, we only measure several channels with MCS7-B40 signal and figure out the other non-measured value by interpolation method.

Usually, the recommended measured channels are listed below:

| 2G Band | | 5G Band1 | | 5G Band2 | | 5G Band3 | | | 5G Band4 | |
|---|---|---|---|---|---|---|---|---|---|---|
| CH4 | CH10 | CH38 | CH46 | CH54 | CH62 | CH102 | CH126 | CH142 | CH151 | CH175 |

**Table 4: The recommended measured channel for Tx power calibration**

**Example :** If the measured Tx index in CH4 is 1 and the measured Tx index in CH10 is 3, all 2G group MCS7-B40 TSSI DE is shown as below:

CH4 is group 2, CH10 is group 4.

| Group 1 | Group 2 | Group 3 | Group 4 | Group 5 |
|---|---|---|---|---|
| 70<br>(Calculated by<br>Interpolation) | 71<br>(Measured) | 72<br>(Calculated by<br>Interpolation) | 73<br>(Measured) | 74<br>(Calculated by<br>Interpolation) |

**Table 5: The example of finding Tx index in 2G band by interpolation**

We have measured all groups of 5G band 1 and 5G band 2, so just use the interpolation for 5G band 3 and 5G band4.

#### 3.1.5.4.2. The Tx power difference

The power value difference are between +7 and -8.

The value 0x0 ~ 0x7 in eFuse means 0 ~ +7 and the value 0x8 ~ 0xF in eFuse means -8 ~ -1.

The +1 power difference will plus 0.5dB power.

**We take Table 7 as an example :**

The 2G power difference are shown as below:

| | |
|---|---|
| **MCS7-B20 to MCS7-B40** | (MCS7-B20_Tatget_Power - MCS7-B40_Tatget_Power)x2<br>= ((13 – 13) x2) = 0 |
| **54M-1T to MCS7-B40** | (54M_Tatget_Power - MCS7-B40_Tatget_Power)x2<br>= ((14 – 13) x2) = 2 |
| **MCS15-B40 to MCS7-B40** | (MCS15-B40_Tatget_Power - MCS7-B40_Tatget_Power)x2<br>= ((12 – 13) x2) = -2<br>==>0xE |
| **MCS15-B20 to MCS7-B20** | 0xE |
| **54M-2T to 54M-1T** | 0xE |
| **CCK-2T to CCK-1T** | 0xE |

**Table 6: The example of finding power difference**

| Power Difference Location in EFuse of Antenna A | | |
|---|---|---|
| 2G Band | 54M-1T to MCS7-B40 | 0x1B[3:0] |
| | MCS7-B20 to MCS7-B40 | 0x1B[7:4] |
| | MCS15-B40 to MCS7-B40 | 0x1C[7:4] |
| | MCS15-B20 to MCS7-B20 | 0x1C[3:0] |
| | 54M-2T to 54M-1T | 0x1D[7:4] |
| | CCK-2T to CCK-1T | 0x1D[3:0] |
| 5G Band | 54M-1T to MCS7-B40 | 0x30[3:0] |
| | MCS7-B20 to MCS7-B40 | 0x30[7:4] |
| | MCS7-Nss1-B80 to MCS7-B40 | 0x36[7:4] |
| | MCS15-B40 to MCS7-B40 | 0x31[7:4] |
| | MCS15-B20 to MCS7-B20 | 0x31[3:0] |
| | 54M-2T to 54M-1T | 0x34[7:4] |
| | MCS7-Nss2-B80 to MCS7-Nss1-B80 | 0x37[7:4] |

| Power Difference Location in EFuse of Antenna B | | |
|---|---|---|
| 2G Band | 54M-1T to MCS7-B40 | 0x45[3:0] |
| | MCS7-B20 to MCS7-B40 | 0x45[7:4] |
| | MCS15-B40 to MCS7-B40 | 0x46[7:4] |
| | MCS15-B20 to MCS7-B20 | 0x46[3:0] |
| | 54M-2T to 54M-1T | 0x47[7:4] |
| | CCK-2T to CCK-1T | 0x47[3:0] |
| 5G Band | 54M-1T to MCS7-B40 | 0x5A[3:0] |
| | MCS7-B20 to MCS7-B40 | 0x5A[7:4] |
| | MCS7-Nss1-B80 to MCS7-B40 | 0x60[7:4] |
| | MCS15-B40 to MCS7-B40 | 0x5B[7:4] |
| | MCS15-B20 to MCS7-B20 | 0x5B[3:0] |
| | 54M-2T to 54M-1T | 0x5E[7:4] |
| | MCS7-Nss2-B80 to MCS7-Nss1-B80 | 0x61[7:4] |

### 3.1.6. Thermal meter

Normal driver will load thermal meter to do power-tracking, so the value must be filled in correct eFuse location. Use the below MP API function to get thermal meter value:

**Read Path A thermal vaule :**

**rtwpriv wlan0 mp_ther 0**

**Read Path B thermal vaule :**

**rtwpriv wlan0 mp_ther 1**

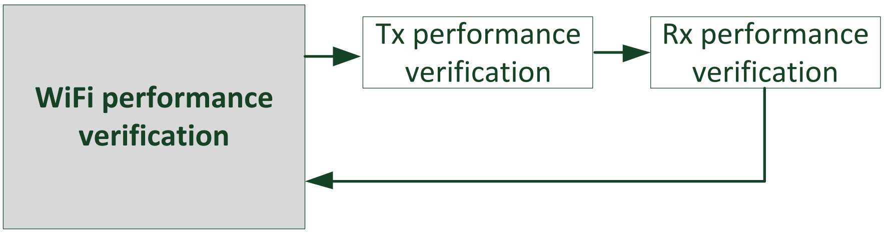
And the feedback information would show like below, 29 is the thermal value.

**wlan0 mp_ther:29**

Write the thermal value into efuse. PathA writes to 0xD0. PathB writes to 0xD1.

| | |
|---|---|
| **Thermal Meter PathA** | 0xD0[7:0] |
| **Thermal Meter PathB** | 0xD1[7:0] |

## 3.2. WiFi performance verification




### 3.2.1. Tx performance verification

The following flow will show the Tx performance verification flow, criterion, and sample command.

#### 3.2.1.1. Tx performance verification criterion

Use the calibrated Tx index in previous step and measure Tx power, EVM, frequency offset and LO leakage to check Tx performance is ok or not. The recommended test items are listed below:

| Data Rate | Antenna | Channel | Item | Criterion |
|---|---|---|---|---|
| MCS7-B40 | Antenna A<br>Antenna B | CH10 | Power | Typical: 16dBm, Acceptable Range: +1/-1.5dB |
| | | | EVM | < -28dB |
| | | | Freq. Err. | ±15ppm |
| | | | Leakage | < -20dBtotal |
| | | | Mask | IEEE spec. defined |
| OFDM 54M | Antenna A<br>Antenna B | CH1 | Power | Typical: 17dBm, Acceptable Range: +1/-1.5dB |
| | | | EVM | < -25dB |
| | | | Freq. Err. | ±15ppm |
| | | | Leakage | < -15dBtotal |
| | | | Mask | IEEE spec. defined |
| CCK 11M | Antenna A<br>Antenna B | CH7 | Power | Typical: 18dBm, Acceptable Range: +1/-1.5dB |
| | | | EVM | < 8% |
| | | | Freq. Err. | ±15ppm |
| | | | Mask | IEEE spec. defined |

**Table 7: The recommended test items of WiFi Tx**

| Data Rate | Antenna | Channel | Item | Criterion |
|---|---|---|---|---|
| MCS7-Nss1-B80 | Antenna A<br>Antenna B | CH42<br>CH58<br>CH138<br>CH155 | Power | Typical: 16dBm, Acceptable Range: +1.5/-2dB |
| | | | EVM | < -28dB |
| | | | Freq. Err. | ±10ppm |
| | | | Leakage | < -25dBtotal |
| | | | Mask | IEEE spec. defined |
| MCS7-B40 | Antenna A<br>Antenna B | CH118 | Power | Typical: 16dBm, Acceptable Range: +1.5/-2dB |
| | | | EVM | < -28dB |
| | | | Freq. Err. | ±10ppm |
| | | | Leakage | < -20dBtotal |
| | | | Mask | IEEE spec. defined |
| OFDM 54M | Antenna A<br>Antenna B | CH100 | Power | Typical: 17dBm, Acceptable Range: +1.5/-2dB |
| | | | EVM | < -25dB |
| | | | Freq. Err. | ±10ppm |
| | | | Leakage | < -15dBtotal |
| | | | Mask | IEEE spec. defined |

**Table 8: The recommended test items of 5G WiFi Tx**

If the instrument supports true MIMO, switch MIMO or composite EVM analysis method, the VHT and HT items (OFDM is as same as above) criterion can be changed to below table:

<table>
<tr><th>Data Rate</th><th>Antenna</th><th>Channel</th><th>Item</th><th>Criterion</th></tr>
<tr><td rowspan="5">MCS7-Nss2-B80</td><td rowspan="5">Antenna AB</td><td rowspan="5">CH42<br>CH58<br>CH138<br>CH155</td><td>Power</td><td>Typical: 15dBm, Acceptable Range: +1.5/-2dB</td></tr>
<tr><td>EVM</td><td>< -28dB</td></tr>
<tr><td>Freq. Err.</td><td>±10ppm</td></tr>
<tr><td>Leakage</td><td>< -25dBtotal</td></tr>
<tr><td>Mask</td><td>IEEE spec. defined</td></tr>
<tr><td rowspan="5">MCS15-B40</td><td rowspan="5">Antenna AB</td><td rowspan="5">CH118</td><td>Power</td><td>Typical: 15dBm, Acceptable Range: +1.5/-2dB</td></tr>
<tr><td>EVM</td><td>< -28dB</td></tr>
<tr><td>Freq. Err.</td><td>±10ppm</td></tr>
<tr><td>Leakage</td><td>< -20dBtotal</td></tr>
<tr><td>Mask</td><td>IEEE spec. defined</td></tr>
<tr><td rowspan="5">MCS15-B40</td><td rowspan="5">Antenna AB</td><td rowspan="5">CH10</td><td>Power</td><td>Typical: 15dBm, Acceptable Range: +1/-1.5dB</td></tr>
<tr><td>EVM</td><td>< -28dB</td></tr>
<tr><td>Freq. Err.</td><td>±15ppm</td></tr>
<tr><td>Leakage</td><td>< -20dBtotal</td></tr>
<tr><td>Mask</td><td>IEEE spec. defined</td></tr>
</table>

**Table 9: The recommended test items of WiFi Tx with 2-stream measurement**

While MP driver is going to measure 256-QAM, Tx index will take the power value in power-by-rate table as measuring reference itself. The criterion would be changed as below table :

<table>
<tr><th>Data Rate</th><th>Antenna</th><th>Channel</th><th>Item</th><th>Criterion</th></tr>
<tr><td rowspan="5">MCS9-Nss1-B80</td><td rowspan="5">Antenna A<br>Antenna B</td><td rowspan="5">CH42<br>CH58<br>CH138<br>CH155</td><td>Power</td><td>Typical: 14dBm, Acceptable Range: +1.5/-2dB</td></tr>
<tr><td>EVM</td><td>< -32dB</td></tr>
<tr><td>Freq. Err.</td><td>±10ppm</td></tr>
<tr><td>Leakage</td><td>< -25dBtotal</td></tr>
<tr><td>Mask</td><td>IEEE spec. defined</td></tr>
</table>

**Table 10: The recommended test items of WiFi Tx with 256-QAM measurement**

*** The table target power is RTK default value , the user can define traget power by yourself.**

### 3.2.1.2. Verify Tx performance

The following steps are the Tx performance verify command sample :

**Step 1 : Enter WL MP mode.**

**rtwpriv wlan0 mp_start**

**Step 2 : Set HW Tx and Tx index to the case which is going to do the verification.**

**Example :** To verify path A, CH10, HT40, MCS7, Calibrate Tx index

**rtwpriv wlan0 mp_txpower patha=calibrated_tx_index, pathb=calibrated_tx_index**

**rtwpriv wlan0 10 1 a HTMCS7 1 2000**

**Step 3 : Measure Tx performance and check the performance with criterion table.**

Path A CH10 HT40 MCS7 criterion :

| Data Rate | Antenna | Channel | Item | Criterion |
|---|---|---|---|---|
| MCS7-B40 | Antenna A<br>Antenna B | CH10 | Power | Typical: 16dBm, Acceptable Range: +1/-1.5dB |
| | | | EVM | < -28dB |
| | | | Freq. Err. | ±15ppm |
| | | | Leakage | < -20dBtotal |
| | | | Mask | IEEE spec. defined |

**Check whether the measured result is fit the criterion or not.**

**Step 4 : Before change the measuring channel, stop HW Tx.**

**rtwpriv wlan0 stop** **//Stop HW packet Tx**

### 3.2.2. Rx performance verification

The following flow will show the Rx performance verification flow, criterion, and sample command.

#### 3.2.2.1. Rx performance verification criterion

Measure the DUT Rx sensitivity to check Rx performance is ok or not. The recommended test items are listed below:

| Data Rate | Antenna | Channel | Item | Criterion |
|---|---|---|---|---|
| **MCS7-B40** | **Antenna A**<br>**Antenna B** | **CH10** | **Sensitivity** | < -61dBm (1R) |
| **OFDM 54M** | **Antenna AB** | **CH1** | | < -68dBm (2R) |
| **CCK 11M** | **Antenna A**<br>**Antenna B** | **CH7** | | < -76dBm (1R) |

| Data Rate | Antenna | Channel | Item | Criterion |
|---|---|---|---|---|
| **MCS9-Nss1-B80** | **Antenna A**<br>**Antenna B** | **CH42** | **Sensitivity** | < -51dBm (1R) |
| **MCS9-Nss1-B80** | **Antenna AB** | **CH58**<br>**CH138**<br>**CH155** | | < -54dBm (2R) |
| **MCS7-B40** | **Antenna A**<br>**Antenna B** | **CH118** | | < -64dBm (2R) |
| **OFDM 54M** | **Antenna A**<br>**Antenna B** | **CH100** | | < -68dBm (2R) |

**Table 11: The recommended test items of WiFi Rx**

The recommended packet-error-rate is < 10%(MCS, OFDM), < 8% (CCK).

***The criterion can be modified by customer requirement.**

***If there is any concern, or want to know the more details, please contact Realtek FAE.**

#### 3.2.2.2. Verify Rx performance

Set the instrument to Tx packets, and DUT MP Rx the packets.
Then parsing the error rate to verify Rx performance.

The following steps are the Rx performance verify command sample :

**Step 1 : Start SW(Software) Rx**

**Example :** Verify path A, MCS7, bandwidth 40, CH10 Rx.

```
rtwpriv wlan0 mp_start              //Start MP mode
rtwpriv wlan0 mp_channel 10         //Set Rx channel
rtwpriv wlan0 mp_ant_rx a           //Set path a
rtwpriv wlan0 mp_bandwidth 40M=1,shortGI=0    //Set bandwidth
                    //40M→40M=1, 20M→40M=0, 80M→40M=2
rtwpriv wlan0 mp_arx start          //Start Rx
rtwpriv wlan0 mp_reset_stats        //Reset the Rx report
```

Instrument Tx packets at MCS7, bandwidth 40, CH10 and
**adjust the Tx power let DUT input power < -61dBm (MCS7-B40 criteria)**.

| Data Rate | Antenna | Channel | Item | Criterion |
|---|---|---|---|---|
| **MCS7-B40** | **Antenna A**<br>**Antenna B** | **CH10** | **Sensitivity** | < -61dBm (1R) |
| **OFDM 54M** | **Antenna AB** | **CH1** | | < -68dBm (2R) |
| **CCK 11M** | **Antenna A**<br>**Antenna B** | **CH7** | | < -76dBm (1R) |

**Step 2 : While the instrument Tx is finished, parsing the Rx report.**

```
rtwpriv wlan0 mp_arx phy
```

The command will report Rx OK/CRC_error counter.

**Step 3 : Calculate packet-error-rate**

Calculate packet-error-rate with step2 result.

The recommended **packet-error-rate is < 10%(MCS, OFDM), < 8% (CCK).**

**Example :**

(1) Instrument Tx 1000 packets at MCS7, bandwidth 40, CH10.

(2) Step 2 reports DUT had Rx 920 packets which Tx from instrument and its input power < -61dBm successfully.

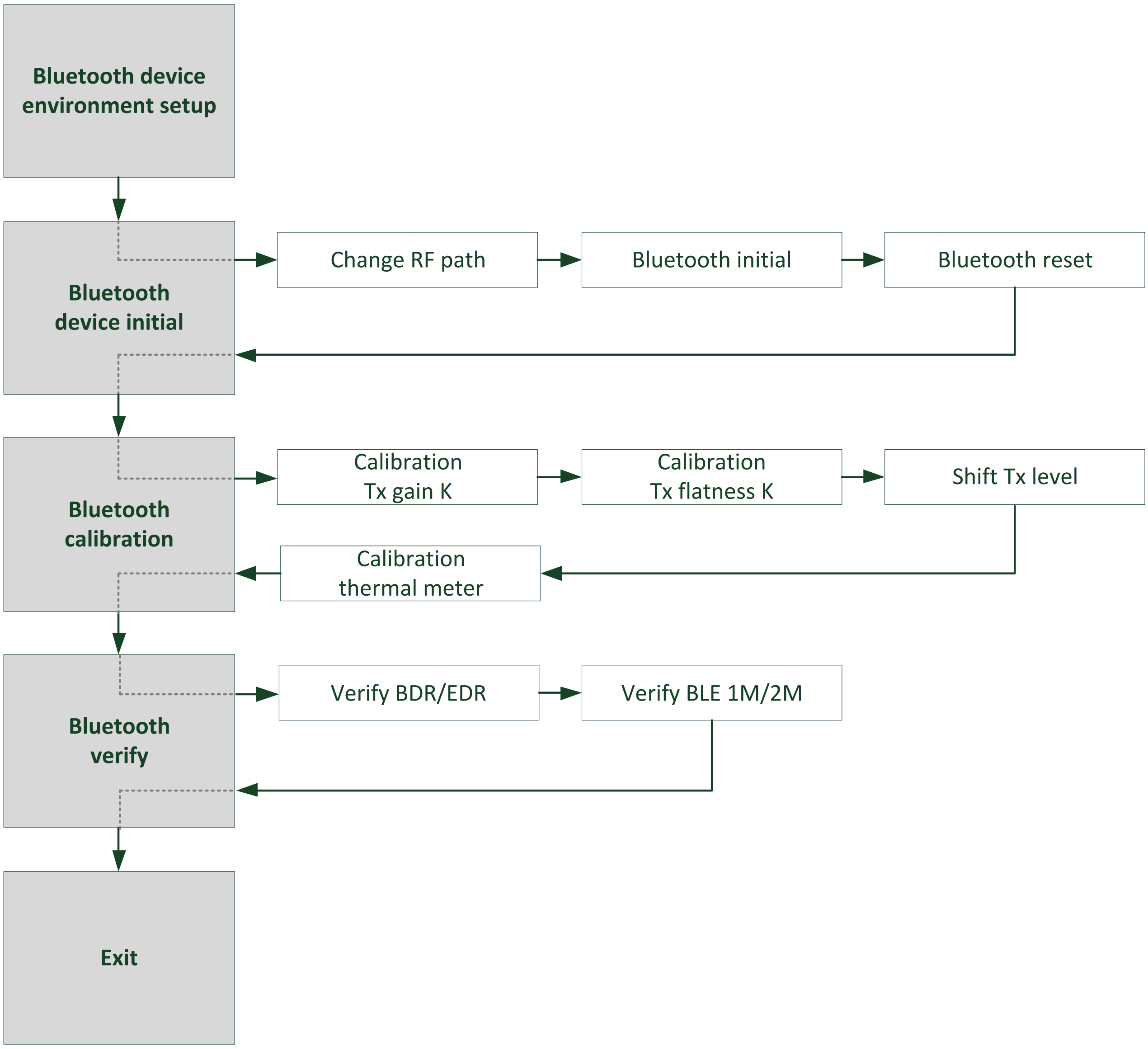
(3) ( 1000 – 920 ) / 1000 = 8%

(4) 8% < 10% → Path A, MCS7, bandwidth 40, CH10 Rx verification pass!

# 4. Bluetooth Mass Production Flow

The below diagram shows a global view of Bluetooth mass production flow.




**Bluetooth Set Tx power Note:**

**1. For general purpose, Tx value for MP is set to 5 dBm to obtain the best power consumption performance and meet different certification (FCC, CE, etc.)**

**2. If customer who uses our module may need to provide permissive change files (Efuse map)during certification of your own product for different Tx power setting. Here listed the main power restraint due to CE compliance, EIRP should less than 10 dBm.**

## 4.1. Bluetooth device environment setup

To do BT RF test, “rtlbtmp” is a necessary tool.

### 4.1.1. Android OS

**MP tool package is provided to customers in binary format:**

MP ADB tool → rtlbtmp

MP library → btmp.default.so

**Customers should copy these binary files to respective directories on target production:**

rtlbtmp → save to → **/system/bin/**

chmod rtlbtmp → **chmod 755 /system/bin/rtlbtmp**

btmp.default.so → save to → **/system/lib/hw/**

mp_rtlxxx_fw, mp_rtlxxx_config → save to → **/system/etc/firmware/**

**Notice: If the system is android o or above:**

rtlbtmp → save to → **/vendor/bin/**

chmod rtlbtmp → **chmod 755 /vendor/bin/rtlbtmp**

btmp.default.so → save to → **/vendor/lib/hw/**

mp_rtlxxx_fw, mp_rtlxxx_config → save to → **/vendor/firmware/**

**Notice: The above files are recommended to be put into the SDK, which will not affect the using of normal Bluetooth.**

**Notice: Before using RTLBTMP, normal BT must be turned off . The following methods is for turn off normal BT.**

**Method 1:** Turn off bluetooth on the UI

**Method 2:** **service call bluetooth_manager 8(close) /6(open)**

**Method 3:** Using command to rename driver file :

**cd /system/lib/hw**

**mv bluetooth.default.so bluetooth.default.so_ORG**

(for android P : **mv libbluetooth.so libbluetooth.so_ORG**)

**Reboot**

#### 4.1.2. Linux OS

**MP tool package is provided to customers in binary format:**

MP CMD tool → rtlbtmp

MP firmware and configure files → mp_rtlxxx_fw, mp_rtlxxx_config

**Customers should copy these binary files to respective directories on target production:**

rtlbtmp → save to → **/usr/sbin/**

chmod rtlbtmp → **chmod 755 /usr/sbin/rtlbtmp**

mp_rtlxxx_fw, mp_rtlxxx_config → save to → **/lib/firmware/**

**Notice: The above files are recommended to be put into the SDK, which will not affect the using of normal Bluetooth.**

**Notice: UART interface chip preparation :**

Please turn off normal Bluetooth. That is rtk_hciattach and the other processes will not be loaded by default after starting up. The command are as below :

**killall rtk_hciattach**

**killall bluetoothd**

**echo 0 > /sys/class/rfkill/rfkill0/state**

**sleep 1**

**echo 1 > /sys/class/rfkill/rfkill0/state**

**sleep 1**

**Notice: USB interface chip preparation :**

Turn on normal Bluetooth first,

**then**

**hciconfig hci0 up**

## 4.2. Bluetooth device initial

In the initial stage, the Bluetooth device must set to factory default and set antenna patch .

**The Initial command list as :**

**Step1 : Change BT RF path**

**ifconfig wlan0 up**

**sleep 1**

**rtwpriv wlan0 mp_start**

**rtwpriv wlan0 mp_setrfpath 0 // 0 or 1**

**rtwpriv wlan0 mp_ant_tx a // a or b**

**Step2 : Bluetooth initial**

**Start MP CMD Tool**

**rtlbtmp**

```
root@tristan-PORTEGE-R700:~# rtlbtmp

::::::::::::::::::::::::::::::::::::::::::::::::
:::::::: Bluetooth MP Test Tool Starting ::::::::
>
```

**Enable MP Stack**

Check Bluetooth stack HCI interface first, then run the enable MP stack CMD.

Please refer "7.1.2. : Bluetooth MP initialize commands at Linux/Android platform".

**enable usb:/dev/rtk_btusb //usb I/F**

**enable uart:/dev/ttyS0 // uart I/F, device node specified by vendor (platform)**

**enable sdio:/dev/sdio //sdio I/F**

```
root@tristan-PORTEGE-R700:~# rtlbtmp

::::::::::::::::::::::::::::::::::::::::::::::::
:::::::: Bluetooth MP Test Tool Starting ::::::::
> enable uart:/dev/ttyUSB0
> > > enable[Success:0]
```

**Step3 : Bluetooth reset**
**Disable Thermal power tracking**

**bt_mp_SetParam 18,0,0**

**bt_mp_Exec 42**

**Set Tx gain K to 0 to " Reset Tx Gain K "**

**bt_mp_SetParam 18,1,0x00 // set ;The value (2's complement) is TX Gain k value**

**bt_mp_Exec 45**

**Set Flatness K to 0 to " Reset Flatness K "**

**bt_mp_SetParam 18,1,0,0**

**bt_mp_Exec 46**

**Set antenna path**

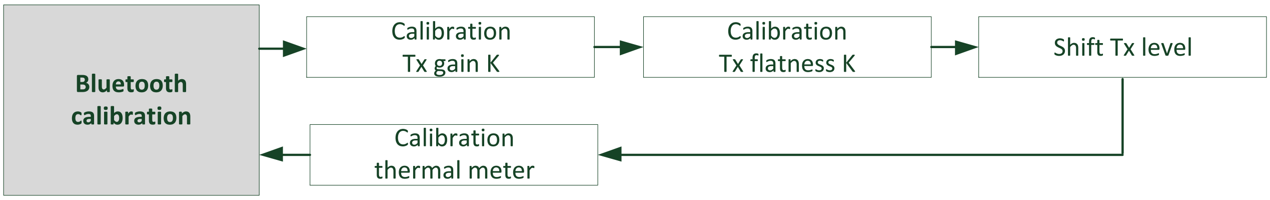
**bt_mp_SetParam 18,0 //The value is antenna path : 0:S0 1:S1**

**bt_mp_Exec 40**

## 4.3. Bluetooth calibration

The flow show as below:




The schematic diagram of this Tx gain K, flatness K and shift Tx level is as follows :

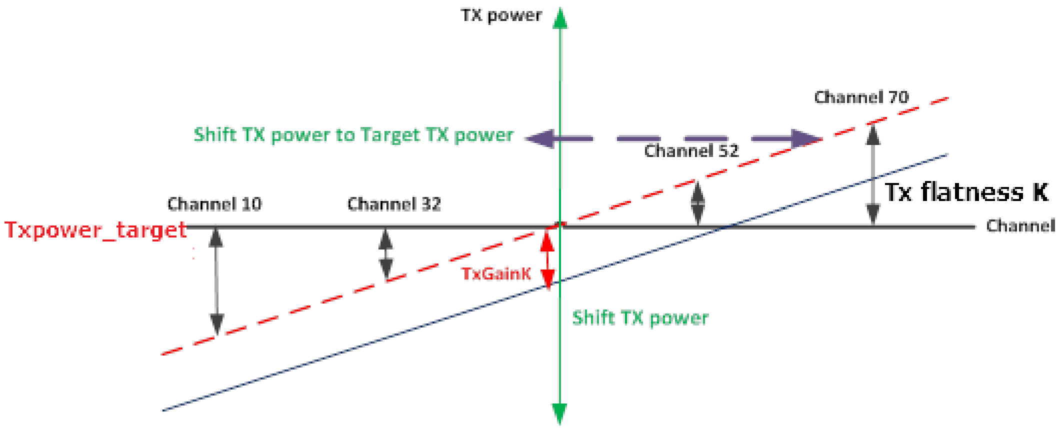



**Figure 9:** Bluetooth TxPower/TxGainK/TxFlatnessK relationship

Now ,We should control the bluetooth device then to enter 2 or 3 command :


**Step1 : Set parameters :**

**bt_mp_SetParam**

The Format is :

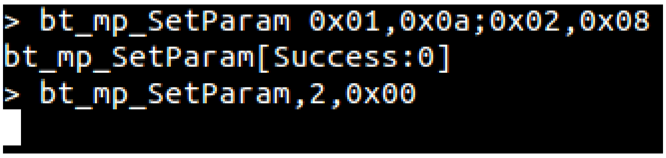
**bt_mp_SetParam Index0,value0; Index1,..;IndexN,valueN**

If there is a requirement to check the parameters, enter the command below :

**bt_mp_GetParam**

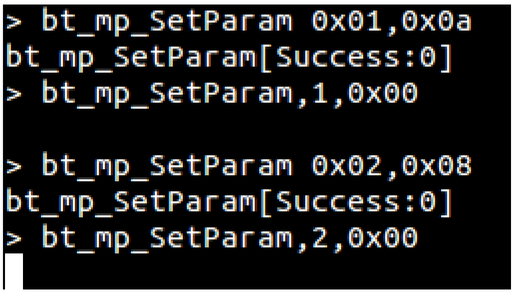
**Example :** If you want to set the channel 10 and packet type "BT_PKT_3DH5" :

**bt_mp_SetParam 0x01,0x0a;0x02,0x08**



```
> bt_mp_SetParam 0x01,0x0a;0x02,0x08
bt_mp_SetParam[Success:0]
> bt_mp_SetParam,2,0x00
```

Or set channel and packet type separately.



```
> bt_mp_SetParam 0x01,0x0a
bt_mp_SetParam[Success:0]
> bt_mp_SetParam,1,0x00

> bt_mp_SetParam 0x02,0x08
bt_mp_SetParam[Success:0]
> bt_mp_SetParam,2,0x00
```

**Step2 : Execute commands**

To execute the parameters/command set in **Step1** :

**bt_mp_Exec ACTION_INDEX**

**Example:** To run " **PACKET_TX_START** " and "**PACKET_TX_STOP**" actions, please use :

**bt_mp_Exec 12 //PACKET_TX_STRAT**

**bt_mp_Exec 14 //PACKET_Tx_STOP**

```
> bt_mp_Exec 12
bt_mp_Exec[Success:0]
> bt_mp_Exec,12,0x00

> bt_mp_Exec 14
bt_mp_Exec[Success:0]
> bt_mp_Exec,14,0x00
```

**For the detail parameters/indexs/command-IDs,**
**please refer to appendix "7.1.4. : Bluetooth MP mode execute commands".**

**Step3 : Report result**

These commands used to report Bluetooth DUT Tx/Rx status are as below:

**bt_mp_Report Item_Index**

**For the whole Item_Index indexs for result report,**
**please refer to appendix "7.1.5. : Bluetooth MP mode report commands".**

#### 4.3.1. Calibration Tx gain K

Here demonstrate the flow about calibrating Tx power to target power. We supposed to use desired target power as calibration power in this step. This Tx gain K is a signed value.

| Tx gain K value | value |
|---|---|
| 12 | 0x0C |
| … | … |
| 4 | 0x04 |
| 3 | 0x03 |
| 2 | 0x02 |
| 1 | 0x01 |
| 0 | 0 (Default) |
| -1 | 0xFF |
| -2 | 0xFE |
| -3 | 0xFD |
| -4 | 0xFC |
| … | … |
| -12 | 0xF4 |

**Table 12: Tx gain K mapping table**

**Step 1 : Begin 2DH5 packet Tx in channel "0x27".**

**bt_mp_SetParam 1,0x27;2,0x06;3,0x07;4,0x00;6,0x7F;7,0x0;11,0x0000009e8b33**

**bt_mp_Exec 30**

**Step 2 : measured by Bluetooth test instrument (e.g. Litepoint IQNxN)**

Txpower_measure is measured value.

**Step 3 : Calculate Tx Gain K value and keep the TxgainK.**

***TxgainK = round((Txpower_target – Txpower_measure)/ 0.5)***

**Example :** Txpower_target = 6 dBm

**Step 4 : Stop Packet Tx (FW_PACKET_TX_STOP=31)**

**bt_mp_Exec 31**

Keep the Tx gain k value write to device in last work.

### 4.3.2. Calibration Tx flatness K

Let's correct the flatness of the channel. The step by step commands are in the below.

**Step 1 : Begin 2DH5 packet TX in channel 12/32/52/70**

**If Channel 12 :**

**bt_mp_SetParam 1,0x0B;2,0x06;3,0x07;4,0x00;6,0x7F;7,0x0;11,0x0000009e8b33**

**If Channel 20 :**

**bt_mp_SetParam 1,0x20;2,0x06;3,0x07;4,0x00;6,0x7F;7,0x0;11,0x0000009e8b33**

**If Channle 52 :**

**bt_mp_SetParam 1,0x34;2,0x06;3,0x07;4,0x00;6,0x7F;7,0x0;11,0x0000009e8b33**

**If Channel 70 :**

**bt_mp_SetParam 1,0x46;2,0x06;3,0x07;4,0x00;6,0x7F;7,0x0;11,0x0000009e8b33**

**bt_mp_Exec 30**

**Step 2 : measured by Bluetooth test instrument (e.g. Litepoint IQNxN).**

**Step 3 : Stop Packet Tx (FW_PACKET_TX_STOP=31),**

**then repeat to 4 channels to complete the measurement**

**bt_mp_Exec 31**

**Step 4 : Calculate flatness value and keep the <u>flatnessK value</u>.**

***flatnessK value [0~3 ] =floor((Txpower_flat_target- Txpower_flat_measure[0~3])*2)***

**Step 5 : Write to efuse of device .( This action can be placed after the final verification)**

This tx flatness k is a signed value, please refer to section "7.3.Tx flatness K mapping table".

### 4.3.3. Shift Tx Level

After finished "Calibrate Tx power to target power" and "Calibrate Tx power flatness", then use the table below to set the target Tx power for different antenna paths. Please refer "7.4. : Bluetooth Tx power table".

The Offset Tx power to RAM command show as:

**Step 1 : Get TX power level:**

**bt_mp_Exec 38**

**bt_mp_Report 17**

**Step 2 : Set TX power level to ram:**

**(1M=0x38);(2M=0x3F);(3M=0x3F),(BLE1M=0x38),(BLE2M=0x38)**

**bt_mp_SetParam 18,0x38,0x3F,0x3F,0x38,0x38;**

**bt_mp_Exec 51**

### 4.3.4. Calibration thermal meter

Normal driver will load thermal meter to do power tracking. So this value must be filled in correct eFuse location. Use the command below to read Bluetooth thermal meter once.
**For Linux/Android platform to get the thermal value from device.**

**bt_mp_Report 7**

***Note: The Thermal Value is RAW Data of Realtek define**

## 4.4. Bluetooth verify

The flow show as:


### 4.4.1. Verify Bluetooth BDR/EDR

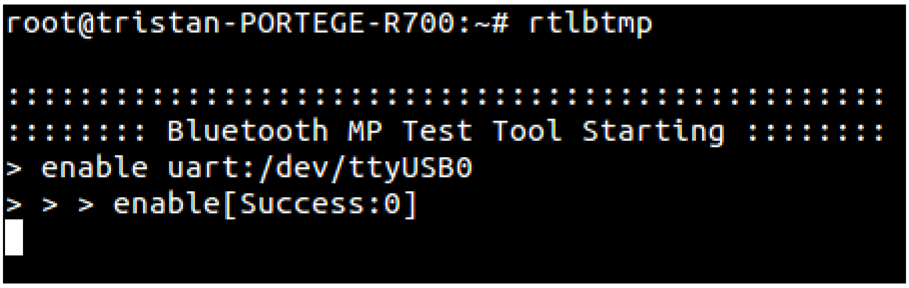
The verify need to verify Tx and RX.

#### 4.4.1.1. Verify Bluetooth BDR/EDR Tx

If the device is UART interface, use below commands at android and Linux platform. Please refer the "7.5. : Verify Bluetooth BDR/EDR Tx SPEC" to check performance SPEC.

The step by step command:

**Step 1 : Enter MP Mode and download patch code**




**Step 2 : Set antenna path.**

If the path is S1, then :

**bt_mp_SetParam 18,1** **//The vale is antenna path : 0:S0 1:S1**

**bt_mp_Exec 40**

**Step 3 : Set Parameter : For example , Only show some settings for DH1 and 3DH5.**

<table>
<tr><th colspan="2" rowspan="2">Test Item</th><th>Command</th></tr>
<tr><th>If Channel = 0x27</th></tr>
<tr><td rowspan="3">DH1</td><td>Maximum Power</td><td>bt_mp_SetParam 1,0x27;2,0x00;3,0x07;4,0x00;6,0x7F;7,0x0;11,0x0000009e8b33</td></tr>
<tr><td>Delta F1</td><td>bt_mp_SetParam 1,0x27;2,0x00;3,0x05;4,0x00;6,0xFF;7,0x0;11,0x0000009e8b33</td></tr>
<tr><td>Delta F2</td><td>bt_mp_SetParam 1,0x27;2,0x00;3,0x02;4,0x00;6,0xFF;7,0x0;11,0x0000009e8b33</td></tr>
<tr><td>3DH5</td><td>ALL</td><td>bt_mp_SetParam 1,0x27;2,0x06;3,0x07;4,0x00;6,0x7F;7,0x0;11,0x0000009e8b33</td></tr>
</table>

**Step 4 : Run Packe Tx (FW_PACKET_TX_START=30)**

**bt_mp_Exec 30**

**Step 5 : measured by Bluetooth test instrument (e.g. Litepoint IQNxN)**

**Step 6 : Stop Packet Tx (FW_PACKET_TX_STOP=31)**

**bt_mp_Exec 31**

**If you need to test other parameters, please stop packet tx and go back to step 2.**

### 4.4.1.2. Verify Bluetooth BDR/EDR Rx

For android and Linux ,you can use below command to test. Please refer the "7.6. : Verify Bluetooth BDR/EDR Rx SPEC" to check performance SPEC.

The below command only shows some settings for DH1 and 3DH1.

**Step 1 : Enter MP Mode and download patch code**

```
root@tristan-PORTEGE-R700:~# rtlbtmp

::::::::::::::::::::::::::::::::::::::::::::::::::
:::::::: Bluetooth MP Test Tool Starting ::::::::
> enable uart:/dev/ttyUSB0
> > > enable[Success:0]
```

**Step 2 : Set Parameter**

| **Test Item** | | **adb command** |
|---|---|---|
| **Channel** | **Packet type** | PayloadType = PRBS9;<br>WhiteningCoeffValue = 0xFF(disable);<br>PacketHeader = **PACKET_HEADER Table**;<br>HitTarget = 0x000000c6967e |
| **6** | **DH1** | **bt_mp_SetParam 1,0x06;2,0x00;3,0x07;6,0xFF;11,0x000000c6967e** |
| **42** | **DH1** | **bt_mp_SetParam 1,0x2a;2,0x00;3,0x07;6,0xFF;11,0x000000c6967e** |
| **70** | **DH1** | **bt_mp_SetParam 1,0x46;2,0x00;3,0x07;6,0xFF;11,0x000000c6967e** |
| **6** | **3DH1** | **bt_mp_SetParam 1,0x06;2,0x06;3,0x07;6,0xFF;11,0x000000c6967e** |
| **42** | **3DH1** | **bt_mp_SetParam 1,0x2a;2,0x06;3,0x07;6,0xFF;11,0x000000c6967e** |
| **70** | **3DH1** | **bt_mp_SetParam 1,0x46;2,0x06;3,0x07;6,0xFF;11,0x000000c6967e** |

Please refer 7.1.3. : Bluetooth MP mode control parameters commands.

**Step 3 : Run Packe Rx (FW_PACKET_RX_START=32)**

**Step 4 : To setting parameter with the Bluetooth test instrument. Bluetooth test instrument begin transmit.**

**bt_mp_Exec 32**

**Step 5 : Report Received Result.**

-- "**bt_mp_Report 3**" should be executed every 1s.

**Step 6 : Stop Packet Rx (FW_PACKET_RX_STOP=33)**

**bt_mp_Exec 33**

**Step 7 : Exit MP Mode**

```
> disable
disable[Success:0]
> quit
:::::::: Bluetooth MP Test Tool Terminating ::::::::
root@tristan-PORTEGE-R700:~#
```

### 4.4.2. Verify Bluetooth BLE

This BLE RF performance includes spec for BT4.0 and BT5.0. This test item contains the following items:

- Verify Bluetooth BLE 1M TX
- Verify Bluetooth BLE 2M TX
- Verify Bluetooth BLE 1M RX
- Verify Bluetooth BLE 2M RX

### 4.4.2.1. Verify Bluetooth BLE 1M Tx (BT4.0)

To measure the DUT BLE TX power and modulation index to check whether BLE TX performance is fine or not. The steps shows the adb commands for UART interface device at Android/Linux platform.

**Step 1 : Enter MP Mode and download patch code**

**Step 2 : Set Parameter :**

| INDEX | VALUE | Value Range (BLE 5.0) |
|---|---|---|
| **1** | **ChannelNumber** | 0~39 |
| **3** | **PayloadType** | BT_LE_PAYLOAD_TYPE_PRBS9 = 0,<br>BT_LE_PAYLOAD_TYPE_1111_0000 = 1,<br>BT_LE_PAYLOAD_TYPE_1010 = 2,<br>BT_LE_PAYLOAD_TYPE_PRBS15 = 3,<br>BT_LE_PAYLOAD_TYPE_ALL1 = 4,<br>BT_LE_PAYLOAD_TYPE_ALL0 = 5,<br>BT_LE_PAYLOAD_TYPE_0000_1111 = 6,<br>BT_LE_PAYLOAD_TYPE_0101 = 7, |
| **15** | **LEDataLen** | 0x00~0xFF |
| **16** | **PHY** | 0x00 : Reserved<br>0x01 : LE 1M PHY<br>0x02 : LE 2M PHY<br>0x03 : LE Coded PHY with S=8 data coding<br>0x04 : LE Coded PHY with S=2 data coding |

| Channel | Test item | Command |
|---|---|---|
| 0 | Avg_power | bt_mp_SetParam 1,0x00;3,0x00;15,0xff;16,1 |
| 19 | Avg_power | bt_mp_SetParam 1,0x13;3,0x00;15,0xff;16,1 |
| 39 | Avg_power | bt_mp_SetParam 1,0x27;3,0x00;15,0xff;16,1 |
| 0 | Delta F1 | bt_mp_SetParam 1,0x00;3,0x01;15,0xff;16,1 |
| 0 | Delta F2 | bt_mp_SetParam 1,0x00;3,0x02;15,0xff;16,1 |

**Step 3 : LE TX TEST**

**bt_mp_Exec 22**

**Step 4 : LE TEST END**

**bt_mp_Exec 24**

Please refer “7.7. : Verify Bluetooth BLE Tx Performance (BLE 1M)”.

### 4.4.2.2. Verify Bluetooth BLE 2M Tx(BT5.0)

If need to verify BLE 5.0, then the step show below:

**Step 3 : Set Parameter**

| Channel | Test item | Command |
|---|---|---|
| **0** | **Avg_power** | **bt_mp_SetParam 1,0x00;3,0x00;15,0xff;16,2** |
| **19** | **Avg_power** | **bt_mp_SetParam 1,0x13;3,0x00;15,0xff;16,2** |
| **39** | **Avg_power** | **bt_mp_SetParam 1,0x27;3,0x00;15,0xff;16,2** |
| **0** | **Delta F1** | **bt_mp_SetParam 1,0x00;3,0x01;15,0xff;16,2** |
| **0** | **Delta F2** | **bt_mp_SetParam 1,0x00;3,0x02;15,0xff;16,2** |

**Step 3 : LE TX TEST**

**bt_mp_Exec 22**

**Step 4 : LE TEST END**

**bt_mp_Exec 24**

Please refer "7.8. : Verify Bluetooth BLE 5.0 Tx Performance".

### 4.4.2.3. Verify Bluetooth BLE 1M Rx(BT4.0)

For android and Linux ,you can use below command to test.

**Step 1 : Set Parameter**

| INDEX | VALUE | Value Range (BLE 5.0) |
|---|---|---|
| 1 | ChannelNumber | 0~39 |
| 16 | PHY | **0x00 :** Reserved<br>**0x01 :** LE 1M PHY<br>**0x02 :** LE 2M PHY<br>**0x03 :** LE Coded PHY |
| 17 | Modulation Index | **0x00 :** Standard<br>**0x01 :** Stable |

| Test Item | | Commands |
|---|---|---|
| Channel | Payload type | |
| 0x00 | PRBS9 | bt_mp_SetParam 1,0x00;16,0x01;17,0x0 |
| 0x13 | PRBS9 | bt_mp_SetParam 1,0x13;16,0x01;17,0x0 |
| 0x22 | PRBS9 | bt_mp_SetParam 1,0x22;16,0x01;17,0x0 |

**Step 2 : Run LE Packet Rx**

**bt_mp_Exec 23** **//LE RX TEST**

**Step 3 : To setting Parameter with the Bluetooth test instrument. Bluetooth test instrument begin transmit.**

**Step 4 : Stop LE Packet Rx and to obtain the receive packet count**

**bt_mp_Exec 24** **//LE TEST END**

**bt_mp_Report 11** **//REPORT LE RX Number of Packets**

### 4.4.2.4. Verify Bluetooth BLE 2M Rx(BT5.0)

The Command step by step list below :

**Step 1 : Enter Hci reset to reset the DUT.**

**Step 2 : Set Parameter & Run LE Packet Rx**

| Test Item | | Command |
|---|---|---|
| Channel | Payload type | |
| 0x00 | PRBS9 | bt_mp_SetParam 1,0x00; 16,0x02;17,0x0 |
| 0x13 | PRBS9 | bt_mp_SetParam 1,0x13; 16,0x02;17,0x0 |
| 0x22 | PRBS9 | bt_mp_SetParam 1,0x22; 16,0x02;17,0x0 |

**Step 3 : LE RX TEST**

**bt_mp_Exec 23**

**Step 4 : LE TEST END**

**bt_mp_Exec 24**

**Step 5 : Report LE Rx number of packets**

**bt_mp_Report 11**

Please refer "7.9. : Verify Bluetooth BLE Rx Performance".

## 4.5. Bluetooth MP Exit

**Disable MP Stack**

**disable**

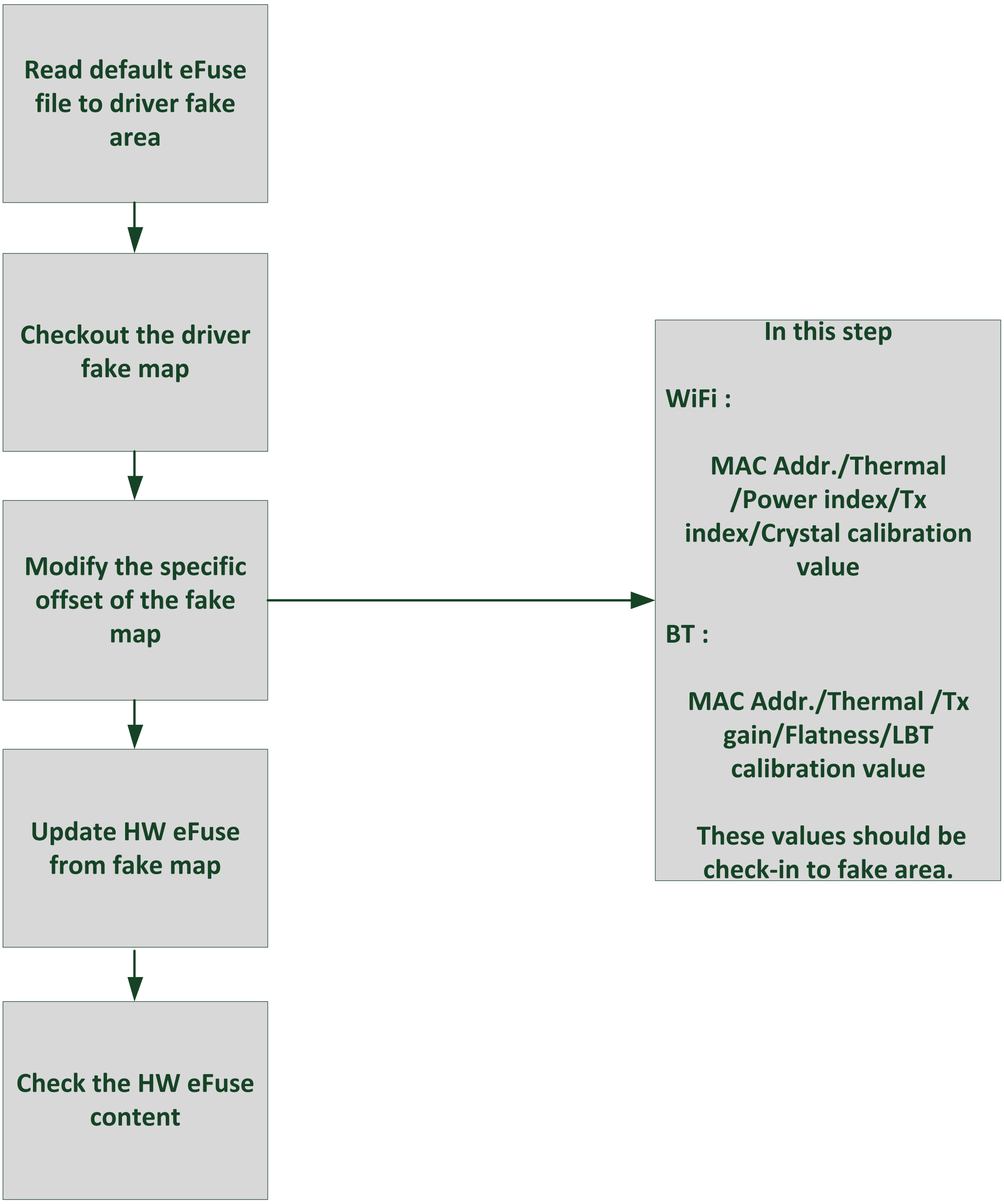
```
> disable
disable[Success:0]
>
```

**Exit MP Tool**

**quit**

```
> quit
:::::::: Bluetooth MP Test Tool Terminating ::::::::
root@tristan-PORTEGE-R700:~#
```

# 5. Dump result to HW efuse



## 5.1. WiFi MP results eFuse check-in flow

### 5.1.1. Read default eFuse file to driver fake area

**If the default map have already loaded in section 3.1.3., please skip 5.1.1..**

**Load external eFuse map file to driver fake map :**

**rtwpriv wlan0 efuse_file /../../efuse.map**

**Load external mask map file to driver fake map :**

**rtwpriv wlan0 efuse_mask /xx/xx/xxmask.txt**

### 5.1.2. Checkout the driver fake map

**Read current driver fake map :**

**rtwpriv wlan0 efuse_get wlrfkmap**

We could check the current driver fake map after the map file was loaded or any modification was applied.

Most of all, please confirm the modified driver fake map cautiously, before updating the driver fake map to fact eFuse map.

### 5.1.3. Modify the specific offset of the fake map

**rtwpriv wlan0 efuse_set wlwfake,Addr,Value(Hex)**

In this step, MAC Addr./Thermal /Power index/Tx index/Crystal calibration value should be check-in to the Fake-Area.

#### 5.1.3.1. Write WiFi MAC address to driver fake map

**EFuse offset address :**

| | |
|---|---|
| **WiFi MAC address** | 0x120 ~ 0x125 |

**Table 13: RTL8822CE WiFi MAC address offset in eFuse**

| | |
|---|---|
| **WiFi MAC address** | 0x157 ~ 0x15c |

**Table 14: RTL8822CU WiFi MAC address offset in eFuse**

| | |
|---|---|
| **WiFi MAC address** | 0x16a ~ 0x16f |

**Table 15: RTL8822CS WiFi MAC address offset in eFuse**

**Example :** To write RTL8822CE MAC address to driver fake map.
(Ex : MAC Addr = 023456789ABC)

**rtwpriv wlan0 efuse_set wlwfake, 120, 023456789ABC**

#### 5.1.3.2. Write thermal value to driver fake map

**EFuse offset address :** Refer to section 3.1.6. : Read thermal meter.
Path A thermal meter : 0xD0[7:0]
Path B thermal meter : 0xD1[7:0]

**rtwpriv wlan0 efuse_set wlwfake,d0,thermal_value_path(Hex)**

**Example :** If path A thermal value = 29, path B thermal value = 30.

**rtwpriv wlan0 efuse_set wlwfake,d0,1d**

**rtwpriv wlan0 efuse_set wlwfake,d1,1e**

### 5.1.3.3. Write calibrated crystal value to driver fake map

**EFuse offset address :** Refer to section 3.1.4. : Crystal calibration.

RTL8822C crystal calibration value should be filled in 0xB9, 0x110, 0x111.

**rtwpriv wlan0 efuse_set wlwfake,Addr,Crystal_value(Hex)**

**Example :** If the calibrated crystal value = 0x40.

**rtwpriv wlan0 efuse_set wlwfake,B9,40**

**rtwpriv wlan0 efuse_set wlwfake,110,40**

**rtwpriv wlan0 efuse_set wlwfake,111,40**

### 5.1.3.4. Write Tx index to driver fake map

**EFuse offset address :** Refer to 3.1.5.1. : Tx index locations in eFuse.

The Tx index for each channel group with PHY data rate should all updated to driver fake map.

**rtwpriv wlan0 efuse_set wlwfake,Addr,index_value(Hex)**

**Example :** If the pathA group 1 MCS7 B40 Tx index value = 72
pathA group 3 MCS7 B40 Tx index value = 71
pathA group 5 MCS7 B40 Tx index value = 70

**rtwpriv wlan0 efuse_set wlwfake,16,48**

**rtwpriv wlan0 efuse_set wlwfake,18,47**

**rtwpriv wlan0 efuse_set wlwfake,1A,46**

### 5.1.4. Update HW eFuse from fake-map

Before updating the modified driver fake map to fact hardware eFuse map, please use below command to confirm whether the modification is completed :

**rtwpriv wlan0 efuse_get wlrfkmap**

If the modification are completed and confirmed, please use below command for updating the driver fake map into the fact hardware eFuse :

**rtwpriv wlan0 efuse_set wlfk2map**

**Attention : It would return error,** if the **mask map** file haven't been loaded simultaneously in the case of the operating MP driver version is over than "rtl8xxx_WiFi_linux_v5.X.X_19292".

### 5.1.5. Check the HW eFuse content

The hardware eFuse map can be gotten by using below command :

**rtwpriv wlan0 efuse_get realmap**

And also check the value which were modified in section 5.1.3.1~5.1.3.4 are all correct.

## 5.2. Bluetooth MP result eFuse check-in flow

### 5.2.1. Read Default E-fuse-File to Driver-Fake-Area

**Load external eFuse map file to driver fake map :**

**rtwpriv wlan0 bt_efuse_file /../../bt_efuse.map**

**Load external mask map file to driver fake map :**

**rtwpriv wlan0 efuse_bt_mask /xx/xx/xx_bt_mask.txt**

### 5.2.2. Checkout the driver fake map

**Read current driver fake map :**

**rtwpriv wlan0 efuse_get btffake //Front part map**

**rtwpriv wlan0 efuse_get btbfake //Back part map**

We could check the current driver fake map after the map file was loaded or any modification was applied.

Most of all, please confirm the modified driver fake map cautiously, before updating the driver fake map to fact eFuse map.

### 5.2.3. Modify the specific offset of the fake map

**rtwpriv wlan0 efuse_set btwfake,Addr,Value(Hex)**

In this step, the BT MAC_Addr, thermal value, Tx Gain K result, Tx power flatness K result should be check-in to the Fake-Area.

### 5.2.3.1. Write BT MAC address to driver fake map:

**EFuse offset address :**

| **BT MAC address** | 0x30 |
|---|---|

**Table 16: Bluetooth MAC address offset in eFuse**

**Example :** To write RTL8822CE BT MAC address to driver fake map.
(Ex : MAC Addr. = 00E04CAABBCC)

**rtwpriv wlan0 efuse_set btwfake, 30,CCBBAA4CE000**

### 5.2.3.2. Write thermal value to driver fake map

**EFuse offset address :** Refer to 7.2. : Bluetooth eFuse definition about calibrates of Tx power.

BT thermal meter : 0x28C[7:0]

**rtwpriv wlan0 efuse_set btwfake,28c,thermal_valu (Hex)**

**Example :** If BT thermal value = 80

**rtwpriv wlan0 efuse_set btwfake, 28c,50**

### 5.2.3.3. Write Tx gain/Flatness/LBT calibration value to driver fake map

**EFuse offset address :**

Refer to 7.2. : Bluetooth eFuse definition about calibrates of Tx power

Tx gain K valid bit = 0x278[1].
Tx flatness K valid bit = 0x278[2].
LBT valid bit = 0x278[5]
Tx gain K result saves to 0x279[7:0].
Tx Flatness K result saves to 0x27A[7:0]~0x27B[7:0].

**rtwpriv wlan0 efuse_set btwfake,Addr,value(Hex)**

**Example :** If both of Tx gain K result = 0, Tx flatness K result = 0, and no LBT.

**rtwpriv wlan0 efuse_set btwfake,278,6**

**rtwpriv wlan0 efuse_set btwfake,279,0**

**rtwpriv wlan0 efuse_set btwfake,27a,0**

**rtwpriv wlan0 efuse_set btwfake,27b,0**

### 5.2.4. Update HW E-fuse from fake-map

Before updating the modified driver fake map to fact hardware eFuse map, please use below command to confirm whether the modification is completed :

**rtwpriv wlan0 efuse_get btffake //Front part map**

**rtwpriv wlan0 efuse_get btbfake //Back part map**

If the modification are completed and confirmed, please use below command for updating the driver fake map into the fact hardware eFuse :

**rtwpriv wlan0 efuse_set btfk2map**

### 5.2.5. Check the HW E-fuse content

The hardware eFuse map can be gotten by using below command :

**rtwpriv wlan0 efuse_get btfmap //Front part map**

**rtwpriv wlan0 efuse_get btbmap //Back part map**

And also check the value which were modified in section 5.2.3.1~5.2.3.3 are all correct.

# 6. WiFi MP flow appendix

## 6.1. Hardware Tx parameters

**rtwpriv wlan0 [Channel] [Bandwidth] [ANT_Path] [RateID] [Tx Mode] [Packet Interval] [Packet Length] [Packet Count] [Packet Pattern]**

**[Channel]** : 1~177

**[Bandwidth]** : 20M = 0, 40M = 1, 80M = 2

**[ANT_Path]** : Path A = a, Path B = b, Path C = c, Path D = d, Path AB(2x2) = ab, …

**[Tx Mode]** : Packet Tx = 1, Continuous Tx = 2, OFDM Single Tone Tx = 3

**[Packet Interval]** : (Option) 1~65535us, default is 2000

**[Packet Length]** : (Option) Data length of packet payload, default is 1500

**[Packet Count]** : (Option) For continuous packet Tx. Counting Tx packets.

**[Packet Pattern]** : (Option) 0x00~0xff, default is random HEX.

**[RateID] :**

| | | | | | | |
|---|---|---|---|---|---|---|
| 1M | 2M | 5.5M | 11M | 6M | 9M | 12M |
| 18M | 24M | 36M | 48M | 54M | | |
| HTMCS0 | HTMCS1 | HTMCS2 | HTMCS3 | HTMCS4 | HTMCS5 | HTMCS6 |
| HTMCS7 | HTMCS8 | HTMCS9 | HTMCS10 | HTMCS11 | HTMCS12 | HTMCS13 |
| HTMCS14 | HTMCS15 | HTMCS16 | HTMCS17 | HTMCS18 | HTMCS19 | HTMCS20 |
| HTMCS21 | HTMCS22 | HTMCS23 | HTMCS24 | HTMCS25 | HTMCS26 | HTMCS27 |
| HTMCS28 | HTMCS29 | HTMCS30 | HTMCS31 | | | |
| VHT1MCS0 | VHT1MCS1 | VHT1MCS2 | VHT1MCS3 | VHT1MCS4 | VHT1MCS5 | VHT1MCS6 |
| VHT1MCS7 | VHT1MCS8 | VHT1MCS9 | | | | |
| VHT2MCS0 | VHT2MCS1 | VHT2MCS2 | VHT2MCS3 | VHT2MCS4 | VHT2MCS5 | VHT2MCS6 |

VHT2MCS7 VHT2MCS8 VHT2MCS9

After measuring, use the below command to stop Tx.

**rtwpriv wlan0 stop** **//stop HW Tx**

## 6.2. Software Rx parameters

**rtwpriv wlan0 mp_start** **//Start MP mode**

**rtwpriv wlan0 mp_channel [Channel]** **//Set Rx channel**

**rtwpriv wlan0 mp_ant_rx [ANT_Path]** **//Set path**

**rtwpriv wlan0 mp_bandwidth 40M=[Bandwidth], shortGI=[GI]** **//Set bandwidth**

**rtwpriv wlan0 mp_arx start** **//Start SW Rx**

**rtwpriv wlan0 mp_reset_stats** **//Reset the Rx report**

**rtwpriv wlan0 mp_arx phy** **//Get Rx phy packet count.**

**rtwpriv wlan0 mp_query** **//Get Tx/Rx packet counter.**

| | | |
|---|---|---|
| **[Channel]** | : | 1~177 |
| **[Bandwidth]** | : | 20M = 0, 40M = 1, 80M = 2 |
| **[GI]** | : | Long GI = 0, Short GI = 1 |
| **[ANT_Path]** | : | Path A = a, Path B = b, Path C = c, Path D = d, Path AB(2x2) = ab, … |

After measuring, use the below command to stop MP.

**rtwpriv wlan0 mp_stop** **//Stop MP mode**

# 7. Bluetooth MP flow appendix

Contains the following

- Tx gain K mapping table
- Tx flatness K mapping table
- Bluetooth Tx power table

## 7.1. Bluetooth MP tool command usage

- Bluetooth start , exit and help command
- Bluetooth MP initialize commands at Linux/Android platform
- Bluetooth control parameters commands
- Bluetooth execute commands
- Bluetooth report commands

### 7.1.1. Bluetooth start , exit and help command

**Bluetooth start MP CMD tool example**

**chmod 755 /system/bin/rtlbtmp**

**rtlbtmp**

```
127|root@android:/system/bin # chmod 755 /system/bin/rtlbtmp
chmod 755 /system/bin/rtlbtmp
root@android:/system/bin # rtlbtmp
rtlbtmp

::::::::::::::::::::::::::::::::::::::::::::::::
:::::::: Bluetooth MP Test Tool Starting ::::::::
>
```

**Enable MP stack**

Check Bluetooth stack HCI interface first, then run the enable MP stack CMD.

**enable usb:/dev/rtk_btusb // USB I/F**

**enable uart:/dev/ttyS0 // UART I/F, device node specified by vendor (platform)**

```
::::::::::::::::::::::::::::::::::::::::::::::::
:::::::: Bluetooth MP Test Tool Starting ::::::::
> enable uart:/dev/ttyS0
enable uart:/dev/ttyS0
> > > enable[Success:0]
```

**Disable MP Stack**

**disable**

```
> disable
disable
disable[Success:0]
>
```

## Exit MP Tool

**quit**

```
> quit
quit
:::::::: Bluetooth MP Test Tool Terminating ::::::::
```

## Lookup MP CMDs

**help**

```
> help
help
help :: Lists all available console commands
quit :: Abort the MP tool test app
enable :: Enable bluetooth
disable :: Disable bluetooth
bt_mp_HciCmd :: Send HCI Commands
bt_mp_GetParam :: Get all/individual exposed parameters
bt_mp_SetParam :: Set specific parameters<index,value>
bt_mp_SetParam1 :: Set series 1 parameters
bt_mp_SetParam2 :: Set series 2 parameters
bt_mp_SetConfig :: Set configurations to the specific file
bt_mp_Exec :: Execute specific action<action id>
bt_mp_Report :: Report specific info according to item selected
bt_mp_RegRW :: R/W Modem, RF, SYS & BB registers
>
```

### 7.1.2. Bluetooth MP initialize commands at Linux/Android platform

These commands used to initialize Bluetooth DUT in MP mode are listed as below:

| MP Command | Parameters | Return | Description |
|---|---|---|---|
| **rtlbtmp** | None | [success] | Start the MP CMD tool. |
| **enable** | USB:<br>enable usb:/dev/rtk_btusb | [success] | Enable USB I/F Bluetooth MP stack and download FW code. Device node is **fixed as /dev/rtk_btusb**. |
| | UART5:<br>enable uart:/dev/ttyS0<br>enable uart5:/dev/ttyS0<br>UART4:<br>enable uart4:/dev/ttyS0 | [success] | Enable Uart I/F Bluetooth MP stack and download FW code. Device node is **chosen by vendor specifically**. H5 (UART5) or H4 (UART4) is determined by the chip configurations. Customers can consult FAE for detailed information. |
| **disable** | None | [success] | Disable Bluetooth MP stack and close the device. |
| **quit** | None | None | Exit from the MP CMD tool. |
| **help** | None | None | List all MP CMDs supported. |

**Table 17: BT_MP_initialize_CMD**

**NOTE: Before running the MP CMD tool, BT on UI settings should be disabled; otherwise, MP tool will be at abnormal status.**

### 7.1.3. Bluetooth MP mode control parameters commands

These commands used to set/get Bluetooth DUT parameters.

| MP Command | Parameters | Return | | |
|---|---|---|---|---|
| **bt_mp_SetParam** | **Index0,value0;Index1,value1;...;IndexN,valueN** | Return Index | Return Status | |
| **bt_mp_GetParam** | Index | Return Index | Return Status | Return Value |

**Table 18: BT MP CONTROL_PARAM_CMD**

**Bluetooth Control Commands: Set/Get parameter command**

**Set parameters :**

**bt_mp_SetParam**

**Check the parameters :**

**bt_mp_GetParam**

**The Format is :**

**bt_mp_SetParam Index0,value0; Index1,..;IndexN,valueN**

| INDEX | VALUE | Length (Byte) | Value Range | Table Index |
|---|---|---|---|---|
| **0** | PGRawData | 256 | Row data | None<br>(Not support Combo Chip WIFI+BT series ) |
| **1** | ChannelNumber | 1 | 0~78 | None |
| **2** | PacketType | 1 | 0~9 | Refer to **PKT_TYPE** |
| **3** | PayloadType | 1 | 0~7 | Refer to **PAYLOAD_TYPE** and **LE_PAYLOAD_TYPE** |
| **4** | TxPacketCount<br>(only for packet tx) | 2 | 0~0xFFF | Refer to **2.3.3.1.3** |
| **5** | TxGainValue | 1 | | |
| **6** | WhiteningCoeffValue | 1 | 0x00~0x7F | **Enable Whitening :** 0x00~0x7f<br>**Disable Whitening :** 0x80 |
| **7** | TxGainIndex | 1 | 0x00: | Refer to **2.3.3.1.5** |

| | | | | |
|---|---|---|---|---|
| | | | Default tx gain | |
| **9** | PacketHeader | 4 | 0x0~0x3FFFF | Refer to **PACKET_HEADER** |
| **10** | HoppingFixChannel (for Hopping mode) | 1 | 0 : Disable<br>1 : Enable fix Channel | None |
| **11** | HitTarget | 6 | 6 bytes | None |
| **14** | Xtal | 4 | 0~0x7F | Depend on chip serials.<br>(Not support Combo Chip WIFI+BT series ) |
| **15** | LEDataLen | 1 | BT4.0 :<br>0x00~0x25<br>BT5.0 :<br>0x00~0xFF | BLE TX Data Length |
| **16** | PHY | 1 | 1~4 | Refer to **BLE5.0_TX_PKT_TYPE** and **BLE5.0_RX_PKT_TYPE** |
| **17** | ModulationIndex | 1 | 0,1 | Refer to **BLE5.0_MODULATION_TYPE** |
| **18** | a. set/get enable/disable tx power tracking<br>b. bt_diff_s0s1<br>c. set gain K<br>d. set flatness<br>e. set tx default power<br>f. SET_K_TX_CH_PWR<br>g. TX_PATH_LOSS_MODULE<br>h. CONFIG_EXTEND<br>i. set Antenna s0/s1<br>j. set hopping start channel and stop channel | | | **TBD** |
| **19** | calculate_XtalBoundary<br>calculate_XtalNormal | 2 | | **TBD** |

**Table 19: BT PARAM_INDEX**

**BT PKT_TYPE**

| NAME | INDEX | Payload Length in bits |
|---|---|---|
| BT_PKT_DH1 | 0 | 0~27*8 |
| BT_PKT_DH3 | 1 | 0~183*8 |
| BT_PKT_DH5 | 2 | 0~339*8 |
| BT_PKT_2DH1 | 3 | 0~54*8 |
| BT_PKT_2DH3 | 4 | 0~367*8 |
| BT_PKT_2DH5 | 5 | 0~679*8 |
| BT_PKT_3DH1 | 6 | 0~83*8 |
| BT_PKT_3DH3 | 7 | 0~552*8 |
| BT_PKT_3DH5 | 8 | 0~1021*8 |
| BT_PKT_LE | 9 | 0~39*8 |

**Table 20: BT PKT_TYPE**

**BLE5.0_Tx_PKT_TYPE**

| NAME | INDEX |
|---|---|
| BLE5_TX_1M_PHY | 1 |
| BLE5_TX_2M_PHY | 2 |
| LE5_TX_CODED_PHY_S8 | 3 |
| LE5_TX_CODED_PHY_S2 | 4 |

**Table 21: BT BLE5.0_Tx_PKT_TYPE**

**BLE5.0_Rx_PKT_TYPE**

| NAME | INDEX |
|---|---|
| BLE5_TX_1M_PHY | 1 |
| BLE5_TX_2M_PHY | 2 |
| LE5_TX_CODED_PHY_LR | 3 |

**Table 22: BT BLE5.0_Rx_PKT_TYPE**

**BLE5.0_MODULATION_TYPE**

| NAME | INDEX |
|---|---|
| STANDARD_MODULATION_INDEX | 0 |
| STABLE_MODULATION_INDEX | 1 |

**Table 23: BT BLE5.0_MODULATION_TYPE**

**The payload types are defined in Table PAYLOAD_TYPE.**

| NAME | INDEX |
|---|---|
| **BT_PAYLOAD_TYPE_ALL0** | 0 |
| **BT_PAYLOAD_TYPE_ALL1** | 1 |
| **BT_PAYLOAD_TYPE_0101** | 2 |
| **BT_PAYLOAD_TYPE_1010** | 3 |
| **BT_PAYLOAD_TYPE_0x0_0xF** | 4 |
| **BT_PAYLOAD_TYPE_0000_1111** | 5 |
| **BT_PAYLOAD_TYPE_1111_0000** | 6 |
| **BT_PAYLOAD_TYPE_PRBS9** | 7 |

**Table 24: BT PAYLOAD_TYPE**

**LE_PAYLOAD_TYPE**

| NAME | INDEX |
|---|---|
| **BT_LE_PAYLOAD_TYPE_PRBS9** | 0 |
| **BT_LE_PAYLOAD_TYPE_1111_0000** | 1 |
| **BT_LE_PAYLOAD_TYPE_1010** | 2 |
| **BT_LE_PAYLOAD_TYPE_PRBS15** | 3 |
| **BT_LE_PAYLOAD_TYPE_ALL1** | 4 |
| **BT_LE_PAYLOAD_TYPE_ALL0** | 5 |
| **BT_LE_PAYLOAD_TYPE_0000_1111** | 6 |
| **BT_LE_PAYLOAD_TYPE_0101** | 7 |

**Table 25: BT LE_PAYLOAD_TYPE**

**PACKET TYPE**

| PACKET TYPE | PAYLOAD(Bits) | PACKET HEADER HEX |
|---|---|---|
| **DH1** | 216 | 33820 |
| **DH3** | 1464 | 39858 |
| **DH5** | 2712 | A078 |
| **2DH1** | 432 | 33820 |
| **2DH3** | 2936 | C050 |
| **2DH5** | 5432 | 3F870 |
| **3DH1** | 664 | 15C40 |
| **3DH3** | 4416 | 39858 |
| **3DH5** | 8168 | A078 |

**Table 26: BT PACKET_HEADER**

**Bluetooth TxPacketCount parameter**

**TxPacketCount** is used to set how many Tx packets will be transmitted. The range of TxPacketCount is from **0 to 0xFFF**. If TxPacketCount value set to "0", it means to send Tx packet counts continuously.

**Bluetooth WhiteningCoeffValue parameter**

The range of **WhiteningCoeffValue** is from **0 to 0x7F**. If WhiteningCoeffValue is "0x80", it means to disable whitening.

**Bluetooth TX gain cal (K) value**

Set Tx gain Cal value will change BR/EDR(1/2/3M) and BLE TX power at the same time.

### 7.1.4. Bluetooth MP mode execute commands

Using this command to control bt mp action, and get current report.

| MP Command | Parameters | Return | |
|---|---|---|---|
| **bt_mp_Exec** | Action | Return Action | Return Status |

You can use

**bt_mp_Exec ACTION_INDEX**

The definition of ACTION_INDEX can refer to "BT_ACTIONCONTROL_TAG" Table.

**Example:** If you want to run FW packet Tx, please use

**bt_mp_Exec 30 // Start FW_PACKET_TX_START**

**bt_mp_Exec 31 //Stop FW_PACKET_TX_STOP**

```
> bt_mp_Exec 12
bt_mp_Exec[Success:0]
> bt_mp_Exec,12,0x00

> bt_mp_Exec 14
bt_mp_Exec[Success:0]
> bt_mp_Exec,14,0x00
```

| **Command /define** | **Index** | **Support Chip RTL8761B** |
|---|---|---|
| HCI_RESET | 0 | ▓ |
| TEST_MODE_ENABLE | 1 | ▓ |
| WRITE_EFUSE_DATA | 2 | ▓ |
| SET_TX_GAIN_TABLE | 3 | ▓ |
| SET_TX_DAC_TABLE | 4 | ▓ |
| SET_DEFAULT_TX_GAIN_TABLE | 5 | ▓ |
| SET_DEFAULT_TX_DAC_TABLE | 6 | ▓ |
| SET_POWER_GAIN_INDEX | 7 | ▓ |
| SET_POWER_GAIN | 8 | ▓ |
| SET_POWER_DAC | 9 | ▓ |
| SET_XTAL | 10 | ▓ |
| REPORT_CLEAR | 11 | ▓ |
| PACKET_TX_START | 12 | |
| PACKET_TX_UPDATE | 13 | |
| PACKET_TX_STOP | 14 | |
| CONTINUE_TX_START | 15 | |
| CONTINUE_TX_UPDATE | 16 | |
| CONTINUE_TX_STOP | 17 | |
| PACKET_RX_START | 18 | |
| PACKET_RX_UPDATE | 19 | |
| PACKET_RX_STOP | 20 | |
| HOPPING_DWELL_TIME | 21 | ▓ |
| LE_TX_DUT_TEST_CMD | 22 | ▓ |
| LE_RX_DUT_TEST_CMD | 23 | ▓ |
| LE_DUT_TEST_END_CMD | 24 | ▓ |
| READ_EFUSE_DATA | 25 | ▓ |
| LE_ CONTINUE _TX _START | 28 | ▓ |
| LE CONTINUE _TX _STOP | 29 | ▓ |
| FW_PACKET_TX_START | 30 | ▓ |
| FW_PACKET_TX_STOP | 31 | ▓ |
| FW_PACKET_RX_START | 32 | ▓ |
| FW_PACKET_RX_STOP | 33 | ▓ |
| FW_CONTINUE_TX_START | 34 | ▓ |
| FW_CONTINUE_TX_STOP | 35 | ▓ |
| FW_LE_CONTINUE_TX_START | 36 | ▓ |

| | | |
|---|---|---|
| FW_LE_CONTINUE_TX_STOP | 37 | ▓ |
| FW_READ_TX_POWER_INFO | 38 | ▓ |
| SET_GPIO3_0 | 39 | |
| SET_ANT_INFO | 40 | |
| SET_ANT_DIFF_S0S1 | 41 | |
| TX_POWER_TRACKING | 42 | ▓ |
| SET_K_TX_CH_PWR | 43 | ▓ |
| WRITE_FLASH_CONFIG | 44 | ▓ |
| TX_POWER_GAIN_K | 45 | ▓ |
| TX_POWER_FLATNESS | 46 | ▓ |
| TX_PATH_LOSS_MODULE | 47 | ▓ |
| CONFIG_EXTEND | 48 | |
| READ_FLASH_CONFIG | 49 | |
| UNLOCK_8822C | 50 | |
| SetTxPower_8822C_8761B | 51 | ▓ |

**Table 27: BT_ACTIONCONTROL_TAG**

## 7.1.5. Bluetooth MP mode report commands

These commands used to report Bluetooth DUT Tx/Rx status are listed as below:

**bt_mp_Report "Item Index"**

| Item Index | Item Index | Return | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **PKT TX = 1** | 1 | | TXBits | TxCounts | | | | | |
| **CONT TX = 2** | 2 | | TXBits | TxCounts | | | | | |
| **PKT RX = 3** | 3 | | RxRssi | RXBits | RxCounts | RxErrorBits | | | |
| **Tx Gain Table = 4** | 4 | | Tx Gain Table | | | | | | |
| **Tx DAC Table = 5** | 5 | | Tx DAC Table | | | | | | |
| **Xtal = 6** | 6 | | Xtal | | | | | | |
| **Thermal = 7** | 7 | | Thermal | | | | | | |
| **Stage = 8** | 8 | | Stage | | | | | | |
| **Efuse = 10** | 10 | | Efuse | | | | | | |
| **LE RX = 11** | 11 | | RxCounts | | | | | | |
| **LE CONT TX=12** | 12 | | TXBits | TxCounts | | | | | |
| **FW_PKT_TX=13** | 13 | | TXBits | TxCounts | | | | | |
| **FW_CONT_TX=14** | 14 | **Status** | TXBits | TxCounts | | | | | |
| **FW_PKT_RX=15** | 15 | | RxRssi | RXBits | RxCounts | RxErrorBits | | | |
| **FW_LE_CONT_TX=16** | 16 | | TXBits | TxCounts | | | | | |
| **TX_POWER_INFO=17** | 17 | | Max tx power index | default tx power index | | | | | |
| | | | | 1M | 2M | 3M | BLE 1M | BLE 2M | LR |
| **REPORT_GPIO3_0** | 18 | | | | | | | | |
| **REPORT_MP_DEBUG_MESSAGE** | 19 | | | | | | | | |
| **REPORT_MP_FT_VALUE** | 20 | | | | | | | | |
| **REPORT_POWER_TRACKING** | 21 | | | | | | | | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **REPORT_MP_TXCAL_INFO** | 22 | | **TxGainK** | **Flatness(LSB)** | **Flatness (MSB)** | **TXPathLoss** | | | |
| **REPORT_FLASH_CONFIG** | 23 | | | | | | | | |
| **REPORT_XTALBOUNDARY** | 24 | | **DIV_int** | **DIV_frac(LSB)** | **DIV_frac (MSB)** | | | | |
| **REPORT_XTALNORMAL** | 25 | | NewXtalIndex | | | | | | |

## 7.2. Bluetooth eFuse definition about calibrates of Tx power

First, view the eFuse (config file) content about setting of power index and channel adjust value. Normal driver will load this value in initial step. So this value must be well-calibrated and filled on correct eFuse location. Please check next page for detail introduction of Tx power setting of efuse (config file).

| Efuse Offset | Explanation |
|---|---|
| 0x278[1] | Tx gain K valid bit* |
| 0x278[2] | Flatness K valid bit** |
| 0x279[7:0] | TX Gain K |
| 0x27A[7:0]~0x27B[7:0] | Flatness K |
| 0x280[7:0] | 1M Max TX Power# |
| 0x281[7:0] | 2M Max TX Power# |
| 0x282[7:0] | 3M Max TX Power# |
| 0x283[7:0] | LE 1M Max TX Power# |
| 0x28C[7:0] | Tmeter module K |

**Table 28: BT Tx gain index offset in eFuse**

***After TX gain K flow please remember to enable this bit which 0x278 will be 02**

****After TX flatness K flow please remember to enable this bit which 0x278 will be 04**

**Note: If both TX gain K & TX flatness K are been done please enable both valid bits which 0x278 will be 06**

**# Please note that RTK adapt max Tx setting for 8822C series combo chip in different certification (FCC, CE, etc.) in order to confirm radio characteristic of our work could still pass compliance rule under extreme condition. For general purpose, Tx value for MP is set to 5 dBm to obtain the best power consumption performance and user experience. Therefore, customer who uses our module may need to provide permissive change files during certification of your own product for different Tx power setting. Here listed the main power restraint due to CE compliance, EIRP should less than 10 dBm.**

**<u>Efuse notification:</u> Default Tx setting of efuse map release by RTK is set to 5 dBm which the setting is shown below as Fig A. If customer need to adopt other Tx target power for their own product please following the power table listed in "[7.4.Bluetooth Tx power table](#)" and <u>revise the efuse map by themselves</u>. For example 4 dBm setting is shown in Fig B. This efuse map file need to burn in DUT after passing through Tx calibration flow. Please note that Max Tx power variation could achieve +- 2 dB due to PCB or component variation even after calibration. Therefore we suggest customer take +- 2dB margin during changing Tx value of efuse map file. For example, if the Max power of the product could not exceed 6 dBm then Tx value for 4 dBm (Fig. 11) is a better choice.**

S0_5dBm efuse map Tx value

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |
| FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |
| FF | FF | FF | FF | FF | FF | FF | FF | 00 | FF | FF | FF | FF | FF | FF | FF |
| BR1M 31 | EDR2M 38 | EDR3M 38 | LE1M 31 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |
| FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |
| FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |

S1_5dBm efuse map Tx value

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |
| FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |
| FF | FF | FF | FF | FF | FF | FF | FF | 00 | FF | FF | FF | FF | FF | FF | FF |
| BR1M 38 | EDR2M 3F | EDR3M 3F | LE1M 38 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |
| FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |
| FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |

**Figure 10:** **BT_5 dBm setting example (Capture image from efuse map file) RTK default map**

S0_4dBm efuse map Tx value

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |
| FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |
| FF | FF | FF | FF | FF | FF | FF | FF | 00 | FF | FF | FF | FF | FF | FF | FF |
| BR1M 2F | EDR2M 36 | EDR3M 36 | LE1M 2F | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |
| FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |
| FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |

S1_4dBm efuse map Tx value

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |
| FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |
| FF | FF | FF | FF | FF | FF | FF | FF | 00 | FF | FF | FF | FF | FF | FF | FF |
| BR1M 36 | EDR2M 3D | EDR3M 3D | LE1M 36 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |
| FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |
| FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |

**Figure 11:** **BT_4 dBm setting example (Capture image from efuse map file)**

## 7.3. Tx flatness K mapping table

**All flatness value can map to below table:**

| Txpower_flat_diff_value | Efuse Flatness value |
|---|---|
| 6 | 0x6 |
| 5 | 0x5 |
| 4 | 0x4 |
| 3 | 0x3 |
| 2 | 0x2 |
| 1 | 0x1 |
| 0 | 0x0 |
| -1 | 0xF |
| -2 | 0xE |
| -3 | 0xD |
| -4 | 0xC |
| -5 | 0xB |
| -6 | 0xA |

**The Txpower_flatness value is defined as :**

| Bits | Define | |
|---|---|---|
| **0~3** | Low Channel Flatness | CH 0~21 |
| **4~7** | MidLow Channel Flatness | CH 22~43 |
| **8~11** | MidHigh Channel, Flatness | CH 44~61 |
| **12~15** | High Channl Flatness | CH 62~78 |

**For example :**

| Channel | Measure Tx power(dBm) | Txpower_flat_diff calculate | Txpower_flat_diff | Value | Flatness value |
|---|---|---|---|---|---|
| 10 | -0.6 | floor((0-(-0.6))*2) | 1 | 0x1 | **0xDE01** |
| 32 | -0.3 | floor((0-(-0.3))*2) | 0 | 0x0 | |
| 52 | 0.7 | floor((0-(0.7))*2) | -2 | 0xE | |
| 70 | 1.2 | floor((0-(1.2))*2) | -3 | 0xD | |

**Note: Efuse offset 0x27A[7:0] = 0x01 , Efuse offset 0x27B[7:0] = 0xDE**

## 7.4. Bluetooth Tx power table

**For S1 <BT/WL Port>**

| Tx power | GFSK/BLE power (dBm) | EDR config power (dBm) |
|---|---|---|
| 0x20 | -7 | -10.5 |
| 0x21 | -6.5 | -10 |
| 0x22 | -6 | -9.5 |
| 0x23 | -5.5 | -9 |
| 0x24 | -5 | -8.5 |
| 0x25 | -4.5 | -8 |
| 0x26 | -4 | -7.5 |
| 0x27 | -3.5 | -7 |
| 0x28 | -3 | -6.5 |
| 0x29 | -2.5 | -6 |
| 0x2A | -2 | -5.5 |
| 0x2B | -1.5 | -5 |
| 0x2C | -1 | -4.5 |
| 0x2D | -0.5 | -4 |
| 0x2E | **0** | -3.5 |
| 0x2F | 0.5 | -3 |
| 0x30 | 1 | -2.5 |
| 0x31 | 1.5 | -2 |
| 0x32 | 2 | -1.5 |
| 0x33 | 2.5 | -1 |
| 0x34 | 3 | -0.5 |
| 0x35 | 3.5 | **0** |
| 0x36 | 4 | 0.5 |
| 0x37 | 4.5 | 1 |
| 0x38 | **5** | 1.5 |
| 0x39 | 5.5 | 2 |
| 0x3A | 6 | 2.5 |
| 0x3B | 6.5 | 3 |
| 0x3C | 7 | 3.5 |
| 0x3D | 7.5 | 4 |
| 0x3E | 8 | 4.5 |
| 0x3F | 8.5 | **5** |
| 0x40 | 9 | 5.5 |
| 0x41 | 9.5 | 6 |
| 0x42 | 10 | 6.5 |
| 0x43 | 10.5 | 7 |
| 0x44 | 11 | 7.5 |
| 0x45 | 11.5 | 8 |
| 0x46 | 12 | 8.5 |
| 0x47 | 12.5 | 9 |
| 0x48 | 13 | 9.5 |
| 0x49 | 13.5 | 10 |

**For S0 <BT Only Port>**

| Tx power | GFSK/BLE power (dBm) | EDR config power (dBm) | Tx power | GFSK/BLE power (dBm) | EDR config power (dBm) |
|---|---|---|---|---|---|
| 0x18 | -7.5 | -11 | 0x3A | 9.5 | 6 |
| 0x19 | -7 | -10.5 | 0x3B | 10 | 6.5 |
| 0x1A | -6.5 | -10 | 0x3C | 10.5 | 7 |
| 0x1B | -6 | -9.5 | 0x3D | 11 | 7.5 |
| 0x1C | -5.5 | -9 | 0x3E | 11.5 | 8 |
| 0x1D | -5 | -8.5 | 0x3F | 12 | 8.5 |
| 0x1E | -4.5 | -8 | 0x40 | 12.5 | 9 |
| 0x1F | -4 | -7.5 | 0x41 | 13 | 9.5 |
| 0x20 | -3.5 | -7 | 0x42 | 13.5 | 10 |
| 0x21 | -3 | -6.5 | | | |
| 0x22 | -2.5 | -6 | | | |
| 0x23 | -2 | -5.5 | | | |
| 0x24 | -1.5 | -5 | | | |
| 0x25 | -1 | -4.5 | | | |
| 0x26 | -0.5 | -4 | | | |
| 0x27 | **0** | -3.5 | | | |
| 0x28 | 0.5 | -3 | | | |
| 0x29 | 1 | -2.5 | | | |
| 0x2A | 1.5 | -2 | | | |
| 0x2B | 2 | -1.5 | | | |
| 0x2C | 2.5 | -1 | | | |
| 0x2D | 3 | -0.5 | | | |
| 0x2E | 3.5 | **0** | | | |
| 0x2F | 4 | 0.5 | | | |
| 0x30 | 4.5 | 1 | | | |
| 0x31 | **5** | 1.5 | | | |
| 0x32 | 5.5 | 2 | | | |
| 0x33 | 6 | 2.5 | | | |
| 0x34 | 6.5 | 3 | | | |
| 0x35 | 7 | 3.5 | | | |
| 0x36 | 7.5 | 4 | | | |
| 0x37 | 8 | 4.5 | | | |
| 0x38 | 8.5 | **5** | | | |
| 0x39 | 9 | 5.5 | | | |

## 7.5. Verify Bluetooth BDR/EDR Tx SPEC

To measure the DUT Bluetooth legacy Tx power/initial Carrier offset/modulation characteristics to check whether Tx performance is fine or not. Bluetooth legacy Tx criterion is shown as below. Below SPEC is defined by using 5 dBm as target power at normal temperature 25 degree C. **For vender’s Tx verification, please adapt target power value as your own SPEC.**

<table>
<tr><th colspan="6">Verify Bluetooth Tx Basic 1M</th></tr>
<tr><th>Test Item</th><th>Sub Test item</th><th>Channel</th><th>Packet Type</th><th>Payload Type</th><th>SPEC</th></tr>
<tr><td rowspan="2">Output Power</td><td>Peak Power</td><td rowspan="10">0<br>6<br>39<br>42<br>70<br>78</td><td rowspan="10">DH1<br>DH3<br>DH5</td><td>PRBS9</td><td>< 9.5 dBm<br>NOTE: Max variation = target power value +/- 4.5 dB*</td></tr>
<tr><td>Average Power</td><td>PRBS9</td><td>3 dBm ~ 7 dBm<br>NOTE: Max variation = target power variation +/- 2 dB*</td></tr>
<tr><td rowspan="4">Modulation Characteristics</td><td>Delta F1 Avg.</td><td>00001111</td><td>140KHz ~ 175KHz</td></tr>
<tr><td>Delta F2 Avg</td><td>10101010</td><td>None</td></tr>
<tr><td>Delta F2 Max.</td><td>10101010</td><td>> 115KHz</td></tr>
<tr><td>Modulation Index</td><td>F2avg/F1avg</td><td>> 0.8</td></tr>
<tr><td colspan="2">Initial Carrier Frequency</td><td>PRBS9</td><td>abs. < 20KHz</td></tr>
<tr><td colspan="2">20dB BW</td><td>PRBS9</td><td>< 1000KHz</td></tr>
<tr><td colspan="2">CF. Drift</td><td>PRBS9</td><td>abs. < 25KHz</td></tr>
<tr><td colspan="2">Max. Drift Rate</td><td>PRBS9</td><td>< 20KHz/50us</td></tr>
</table>

**Table 29: Verify Bluetooth Tx Basic 1M**

***Max variation SPEC only available after doing Tx calibration flow**

| Verify Bluetooth Tx EDR 2M | | | | | |
|---|---|---|---|---|---|
| **Test Item** | **Sub Test item** | **Channel** | **Packet Type** | **Payload Type** | **SPEC** |
| Output Power | Peak Power | 0<br>6<br>39<br>42<br>70<br>78 | 2DH1<br>2DH3<br>2DH5 | PRBS9 | < 9 dBm<br>NOTE: Max variation = target power value +/- 4dB* |
| | Average Power | | | | 3 dBm ~ 7 dBm<br>NOTE: Max variation = target power value +/- 2 dB* |
| Relative Transmit Power ($P_{GFSK}$ - $P_{DPSK}$) | | | | | -4 ~ 1 |
| Omega I | | | | | abs. < 20KHz |
| Omega 0 | | | | | abs. < 10KHz |
| Omega I + | | | | | abs. < 20KHz |
| Enhanced data rate carrier frequency stability and modulation accuracy | RMS DEVM | | | | < 0.20 |
| | Peak DEVM | | | | < 0.35 |
| | 0.99 DEVM | | | | > 0.99 |

**Table 30: Verify Bluetooth Tx EDR 2M**

***Max variation SPEC only available after doing Tx calibration flow**

<table>
<tr><th colspan="6">Verify Bluetooth Tx EDR 3M</th></tr>
<tr><th>Test Item</th><th>Sub Test item</th><th>Channel</th><th>Packet Type</th><th>Payload Type</th><th>SPEC</th></tr>
<tr><td rowspan="2">Output Power</td><td>Peak Power</td><td rowspan="9">0<br>6<br>39<br>42<br>70<br>78</td><td rowspan="9">3DH1<br>3DH3<br>3DH5</td><td rowspan="9">PRBS9</td><td>< 9 dBm<br>NOTE: Max variation = target power value + / - 4dB*</td></tr>
<tr><td>Average Power</td><td>3 dBm ~ 7 dBm<br>NOTE: Max variation = target power value +/- 2 dB*</td></tr>
<tr><td colspan="2">Relative Transmit Power<br>($P_{GFSK}$ - $P_{DPSK}$)</td><td>-4.00dB - 1.00dB</td></tr>
<tr><td colspan="2">Omega I</td><td>abs. < 20KHz</td></tr>
<tr><td colspan="2">Omega 0</td><td>abs. < 10KHz</td></tr>
<tr><td colspan="2">Omega I +</td><td>abs. < 20KHz</td></tr>
<tr><td rowspan="3">Enhanced data rate carrier frequency stability and modulation accuracy</td><td>RMS DEVM</td><td>< 0.13</td></tr>
<tr><td>Peak DEVM</td><td>< 0.25</td></tr>
<tr><td>0.99 DEVM</td><td>> 0.99</td></tr>
</table>

**Table 31: Verify Bluetooth Tx EDR 3M**

## 7.6. Verify Bluetooth BDR/EDR Rx SPEC

To Measure the DUT Rx sensitivity to check whether Rx performance is fine or not. The Rx performance test can be measured in signaling mode (ex: Anritsu 8852B, Agilent N4010A) or non-signaling mode (ex: LitePoint IQNxN). Bluetooth Rx criterion is shown as below :

| Channel | Payload Type | Legacy sensitivity limit | Criterion |
|---|---|---|---|
| | | | Bluetooth SPEC |
| 0/6/39/42/70/78 | PRBS9 | EDR1M_BER < 0.01% | **< -85 dBm** |
| 0/6/39/42/70/78 | PRBS9 | EDR2M/3M_BER < 0.01% | **< -80 dBm** |

**Table 32: Bluetooth Legacy Rx criterion**

# 7.7. Verify Bluetooth BLE Tx Performance (BLE 1M)

To measure the DUT BLE Tx power and modulation index to check whether BLE Tx performance is fine or not. Bluetooth BLE Tx criterion is shown as below. Below Spec is defined by using 5 dBm as target power at normal temperature 25 degree C. **For vender's Tx verification, please adapt target power value as your own SPEC.**

<table>
<tr><th rowspan="2">Test Item</th><th rowspan="2">Sub Test Item</th><th rowspan="2">Payload Type</th><th rowspan="2">Channel</th><th>Criterion</th></tr>
<tr><th>Bluetooth SPEC</th></tr>
<tr><td rowspan="2">BLE Output Power</td><td>Average Power</td><td rowspan="2">PRBS9</td><td rowspan="6">0/3/19/22/36/39</td><td>3 dBm ~ 7dBm<br>NOTE: Max variation = target power value +/- 2 dB*</td></tr>
<tr><td>Peak Power</td><td>< 9 dBm<br>NOTE: Max variation = target power value + / - 4dB*</td></tr>
<tr><td>Carrie freq. offset & drift</td><td>None</td><td>PRBS9</td><td>< 20KHz</td></tr>
<tr><td rowspan="3">Modulation Characteristics</td><td>Delta F1 Avg.</td><td>BT_PAYLOAD_TYPE_1111_0000</td><td>225 kHz ~ 275 kHz</td></tr>
<tr><td>Delta F2 Max.</td><td>BT_PAYLOAD_TYPE_1010</td><td>> 185 kHz</td></tr>
<tr><td>F2avg/F1avg</td><td>None</td><td>> 0.8</td></tr>
</table>

**Table 33: Bluetooth BLE Tx criterion**

***Max variation SPEC only available after doing Tx calibration flow**

## 7.8. Verify Bluetooth BLE 5.0 Tx Performance

To measure the DUT BLE 5.0 Tx power and modulation index to check whether BLE Tx performance fine or not. Bluetooth BLE 5.0 Tx criterion is shown as below. Below SPEC is defined by using 5 dBm as target power at normal temperature 25 degree C. **For vender's Tx verification, please adapt target power value as your own spec.**

<table>
<tr><th rowspan="2">PHY SPEC</th><th rowspan="2">Test Item</th><th rowspan="2">Sub Test Item</th><th rowspan="2">Payload Type</th><th rowspan="2">Channel</th><th>Criterion</th></tr>
<tr><th>Bluetooth SPEC</th></tr>
<tr><td>2M</td><td rowspan="4">BLE Output Power</td><td rowspan="3">Average Power</td><td rowspan="4">PRBS9</td><td rowspan="10">0/3/19/22/36/39</td><td rowspan="3">3 dBm ~ 7 dBm<br>NOTE: Max variation = target power value +/- 2 dB*</td></tr>
<tr><td>LR S2</td></tr>
<tr><td rowspan="2">LR S8</td></tr>
<tr><td>Peak<br>Power</td><td>< 9 dBm<br>NOTE: Max variation = target power value + / - 4dB*</td></tr>
<tr><td>2M</td><td rowspan="3">Carrie freq. offset & drift</td><td rowspan="3">None</td><td>PRBS9</td><td>2M: < 20kHz</td></tr>
<tr><td>LR S2</td><td>PRBS9</td><td>S8: < 19.2kHz</td></tr>
<tr><td>LR S8</td><td>PRBS9</td><td></td></tr>
<tr><td>2M</td><td rowspan="3">Modulation Characteristics</td><td>Delta F1 Avg.</td><td>BT_PAYLOAD_TYPE_1111_0000</td><td>2M: 450 ~ 550 kHz<br>S8: 225 ~ 275 kHz</td></tr>
<tr><td>LR S2</td><td>Delta F2 Max.</td><td>BT_PAYLOAD_TYPE_1010</td><td>2M: > 370kHz</td></tr>
<tr><td>LR S8</td><td>F2avg/F1avg</td><td>None</td><td>2M: > 0.8</td></tr>
</table>

**Table 34: Bluetooth BLE 5.0 Tx criterion**

***Max variation SPEC only available after doing Tx calibration flow**

## 7.9. Verify Bluetooth BLE Rx Performance

To calculate the Packet Error Rate (PER)

***PER%= 100*(1-(Packets Received /Packets Send)***

To measure the DUT BLE Rx sensitivity to check whether Rx performance is fine or not.

The Bluetooth Rx criterion is shown as below:

| **Channel** | **SPEC** | **Payload Type** | **BLE sensitivity limit** | **Criterion** |
|---|---|---|---|---|
| | **PHY SPEC** | | | **Bluetooth SPEC** |
| 0/3/19/22/36/39 | 1M | PRBS9 | PER<= 30.800 % | **< -85 dBm** |
| | 2M | PRBS9 | PER<= 30.800 % | **< -85 dBm** |
| | LRS2 | PRBS9 | PER<= 30.800 % | **< -85 dBm** |
| | LRS8 | PRBS9 | PER<= 30.800 % | **< -85 dBm** |

**Table 35: Bluetooth BLE Rx criterion**

# FCC Caution

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions:

(1) This device may not cause harmful interference,
(2) this device must accept any interference received, including interference that may cause undesired operati
Any Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

Note: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

-Reorient or relocate the receiving antenna.
-Increase the separation between the equipment and receiver.
-Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
-Consult the dealer or an experienced radio/TV technician for help.

This equipment complies with FCC radiation exposure limits set forth for an uncontrolled environment.

This equipment should be installed and operated with minimum distance 20cm between the radiator & your body.

## Requirement per KDB996369 D03

### 2.2 List of applicable FCC rules

List the FCC rules that are applicable to the modular transmitter. These are the rules that specifically establish the bands of operation, the power, spurious emissions, and operating fundamental frequencies. DO NOT list compliance to unintentional-radiator rules (Part 15 Subpart B) since that is not a condition of a module grant that is extended to a host manufacturer. See also Section 2.10 below concerning the need to notify host manufacturers that further testing is required.3

**Explanation:** This module meets the requirements of FCC part 15C(15.247).

### 2.3 Summarize the specific operational use conditions

Describe use conditions that are applicable to the modular transmitter, including for example any limits on antennas, etc. For example, if point-to-point antennas are used that require reduction in power or compensation for cable loss, then this information must be in the instructions. If the use condition limitations extend to professional users, then instructions must state that this information also extends to the host manufacturer's instruction manual. In addition, certain information may also be needed, such as peak gain per frequency band and minimum gain, specifically for master devices in 5 GHz DFS bands.

**Explanation:** The EUT has a Ceramic Antenna, and the antenna use a permanently attached antenna which is not replaceable.

### 2.4 Limited module procedures

If a modular transmitter is approved as a "limited module," then the module manufacturer is responsible for approving the host environment that the limited module is used with. The manufacturer of a limited module must describe, both in the filing and in the installation instructions, the alternative means that the limited module manufacturer uses to verify that the host meets the necessary requirements to satisfy the module limiting conditions.

A limited module manufacturer has the flexibility to define its alternative method to address the conditions that limit the initial approval, such as: shielding, minimum signaling amplitude, buffered modulation/data inputs, or power supply regulation. The alternative method could include that the limited module manufacturer reviews detailed test data or host designs prior to giving the host manufacturer approval.

This limited module procedure is also applicable for RF exposure evaluation when it is necessary to demonstrate compliance in a specific host. The module manufacturer must state how control of the product into which the modular transmitter will be installed will be maintained such that full compliance of the product is always ensured. For additional hosts other than the specific host originally granted with a limited module, a Class II permissive change is required on the module grant to register the additional host as a specific host also approved with the module.

**Explanation:** The module is not a limited module.

### 2.5 Trace antenna designs

For a modular transmitter with trace antenna designs, see the guidance in Question 11 of KDB Publication 996369 D02 FAQ – Modules for Micro-Strip Antennas and traces. The integration information shall include for the TCB review the integration instructions for the following aspects: layout of trace design, parts list (BOM), antenna, connectors, and isolation requirements.

a) Information that includes permitted variances (e.g., trace boundary limits, thickness, length, width, shape(s), dielectric constant, and impedance as applicable for each type of antenna);

b) Each design shall be considered a different type (e.g., antenna length in multiple(s) of frequency, the wavelength, and antenna shape (traces in phase) can affect antenna gain and must be considered);

c) The parameters shall be provided in a manner permitting host manufacturers to design the printed circuit (PC) board layout;

d) Appropriate parts by manufacturer and specifications;

e) Test procedures for design verification; and

f) Production test procedures for ensuring compliance.

The module grantee shall provide a notice that any deviation(s) from the defined parameters of the antenna trace, as described by the instructions, require that the host product manufacturer must notify the module grantee that they wish to change the antenna trace design. In this case, a Class II permissive change application is required to be filed by the grantee, or the host manufacturer can take responsibility through the change in FCC ID (new application) procedure followed by a Class II permissive change application.

**Explanation:** Yes, The module with trace antenna designs, and This manual has been shown the layout of trace design,, antenna, connectors, and isolation requirements.

### 2.6 RF exposure considerations

It is essential for module grantees to clearly and explicitly state the RF exposure conditions that permit a host product manufacturer to use the module. Two types of instructions are required for RF exposure information: (1) to the host product manufacturer, to define the application conditions (mobile, portable – xx cm from a person's body); and (2) additional text needed for the host product manufacturer to provide to end users in their end-product manuals. If RF exposure statements and use conditions are not provided, then the host product manufacturer is required to take responsibility of the module through a change in FCC ID (new application).

**Explanation:** This module complies with FCC RF radiation exposure limits set forth for an uncontrolled environment, This equipment should be installed and operated with a minimum distance of 20 centimeters between the radiator and your body." This module is designed to comply with the FCC statement

### 2.7 Antennas

A list of antennas included in the application for certification must be provided in the instructions. For modular transmitters approved as limited modules, all applicable professional installer instructions must be included as part of the information to the host product manufacturer. The antenna list shall also identify the antenna types (monopole, PIFA, dipole, etc. (note that for example an "omni-directional antenna" is not considered to be a specific "antenna type")).

For situations where the host product manufacturer is responsible for an external connector, for example with an RF pin and antenna trace design, the integration instructions shall inform the installer that unique antenna connector must be used on the Part 15 authorized transmitters used in the host product. The module manufacturers shall provide a list of acceptable unique connectors.

**Explanation:** The EUT has a Ceramic Antenna, and the antenna use a permanently attached antenna which is unique.

### 2.8 Label and compliance information

Grantees are responsible for the continued compliance of their modules to the FCC rules. This includes advising host product manufacturers that they need to provide a physical or e-label stating "Contains FCC ID" with their finished product. See Guidelines for Labeling and User Information for RF Devices – KDB Publication 784748.

**Explanation:**The host system using this module, should have label in a visible area indicated the following texts: "Contains FCC ID: AQ5RWF-M68B-UWK1.

### 2.9 Information on test modes and additional testing requirements5

Additional guidance for testing host products is given in KDB Publication 996369 D04 Module Integration Guide. Test modes should take into consideration different operational conditions for a stand-alone modular transmitter in a host, as well as for multiple simultaneously transmitting modules or other transmitters in a host product.

The grantee should provide information on how to configure test modes for host product evaluation for different operational conditions for a stand-alone modular transmitter in a host, versus with multiple, simultaneously transmitting modules or other transmitters in a host.

Grantees can increase the utility of their modular transmitters by providing special means, modes, or instructions that simulates or characterizes a connection by enabling a transmitter. This can greatly simplify a host manufacturer's determination that a module as installed in a host complies with FCC requirements.

**Explanation:** KTC can increase the utility of our modular transmitters by providing instructions that simulates or characterizes a connection by enabling a transmitter.

**2.10 Additional testing, Part 15 Subpart B disclaimer**

The grantee should include a statement that the modular transmitter is **only** FCC authorized for the specific rule parts (i.e., FCC transmitter rules) listed on the grant, and that the host product manufacturer is responsible for compliance to any other FCC rules that apply to the host not covered by the modular transmitter grant of certification. If the grantee markets their product as being Part 15 Subpart B compliant (when it also contains unintentional-radiator digital circuity), then the grantee shall provide a notice stating that the final host product still requires Part 15 Subpart B compliance testing with the modular transmitter installed.

**Explanation:** The module without unintentional-radiator digital circuity, so the module does not require an evaluation by FCC Part 15 Subpart B. The host shoule be evaluated by the FCC Subpart B.