



# BES2600-WM Wi-Fi & BT Test Flow

Version: 2.6 20220112

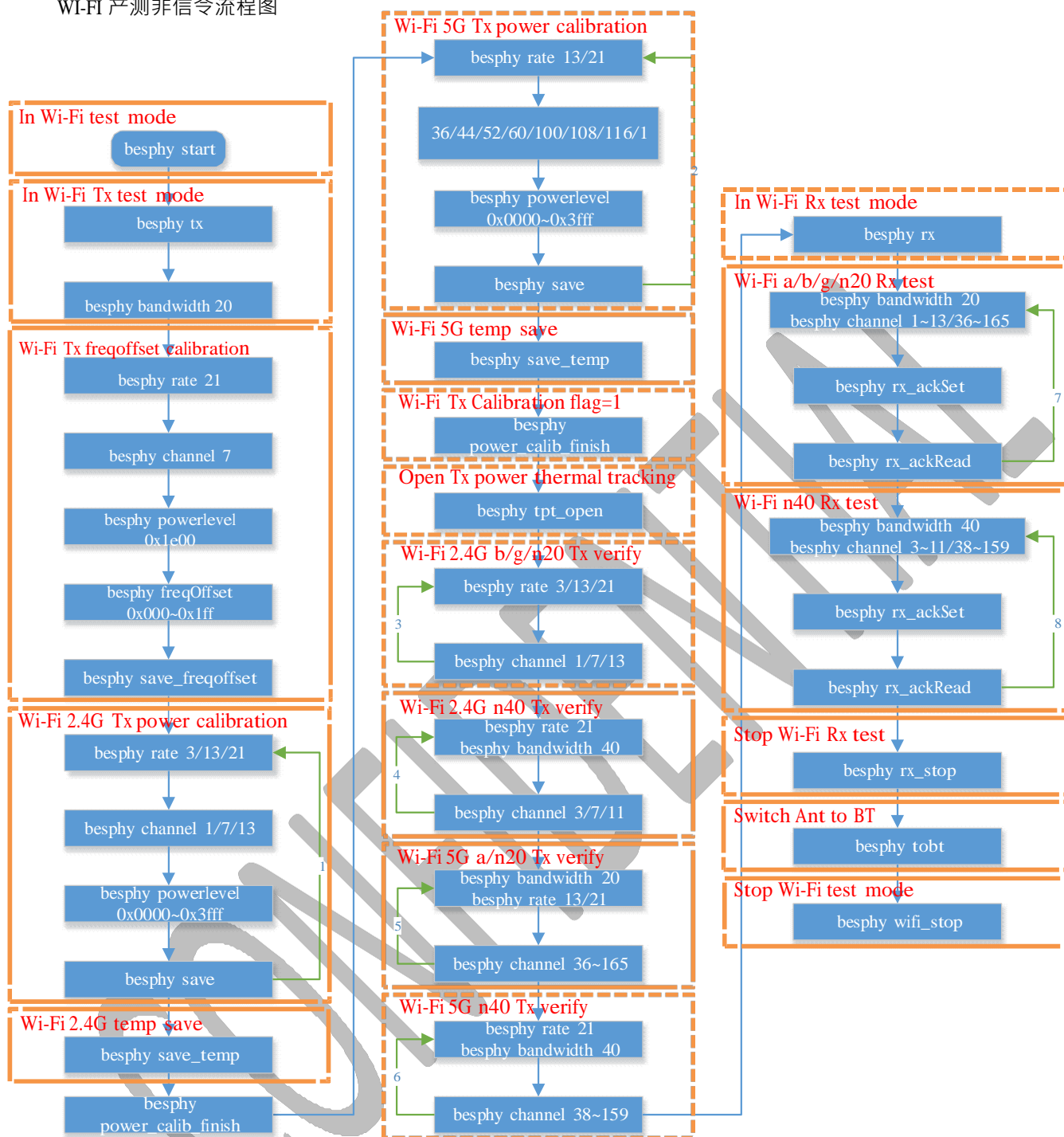
## 目录

1. Wi-Fi Tx Calibration And Tx/Rx Verfy .....	6
1.1 进入非信令测试模式 .....	6
1.2 进入 Wi-Fi Tx 非信令测试模式 .....	6
1.3 Wi-Fi Tx Frequency Calibration ( BT Tx 频偏与 WI-FI 共用校准值 ) .....	6
1.4 Wi-Fi Tx Power Calibration .....	8
1.4.1 CCK11 CH1 功率校准 .....	8
1.3.2 OFDM54 CH7 功率校准 .....	8
1.3.3 HT20-MCS7 CH13 功率校准 .....	8
1.5 Wi-Fi Tx Thermal Save .....	10
1.6 Wi-Fi Tx Verify .....	10
1.6.1 如下 CCK11 CH1 Tx Verify 指令 : .....	10
1.6.2 如下 OFDM54 CH7 Tx Verify 指令 : .....	10
1.6.3 如下 HT20-MCS7 CH13 Tx Verify 指令 : .....	10
1.6.4 如下 MCS7 CH13 Tx Verify 指令 : .....	10
1.7 Wi-Fi Rx Verify .....	10
1.7.1 CCK11/OFDM54/HT20-MCS7 CH1 Rx Verify .....	10
1.7.2 CCK11/OFDM54/HT20-MCS7 CH7 Rx Verify .....	11
1.7.3 CCK11/OFDM54/HT20-MCS7 CH13 Rx Verify .....	11
1.7.4 HT40-MCS7 CH11 Rx Verify .....	11
2. BT Tx Calibration And Tx/Rx Verfy .....	11
2.1 Wi-Fi Switch to BT .....	11
2.2 BT BDR Tx Power Calibration .....	11
2.3 BT EDR Tx Power Calibration .....	12
2.4 BT BLE Tx Power Calibration .....	12
2.5 BT BDR Tx Verify .....	13
2.6 BT EDR Tx Verify .....	13
2.7 BLE Tx Verify .....	14
2.8 BT BDR Rx Verify .....	14

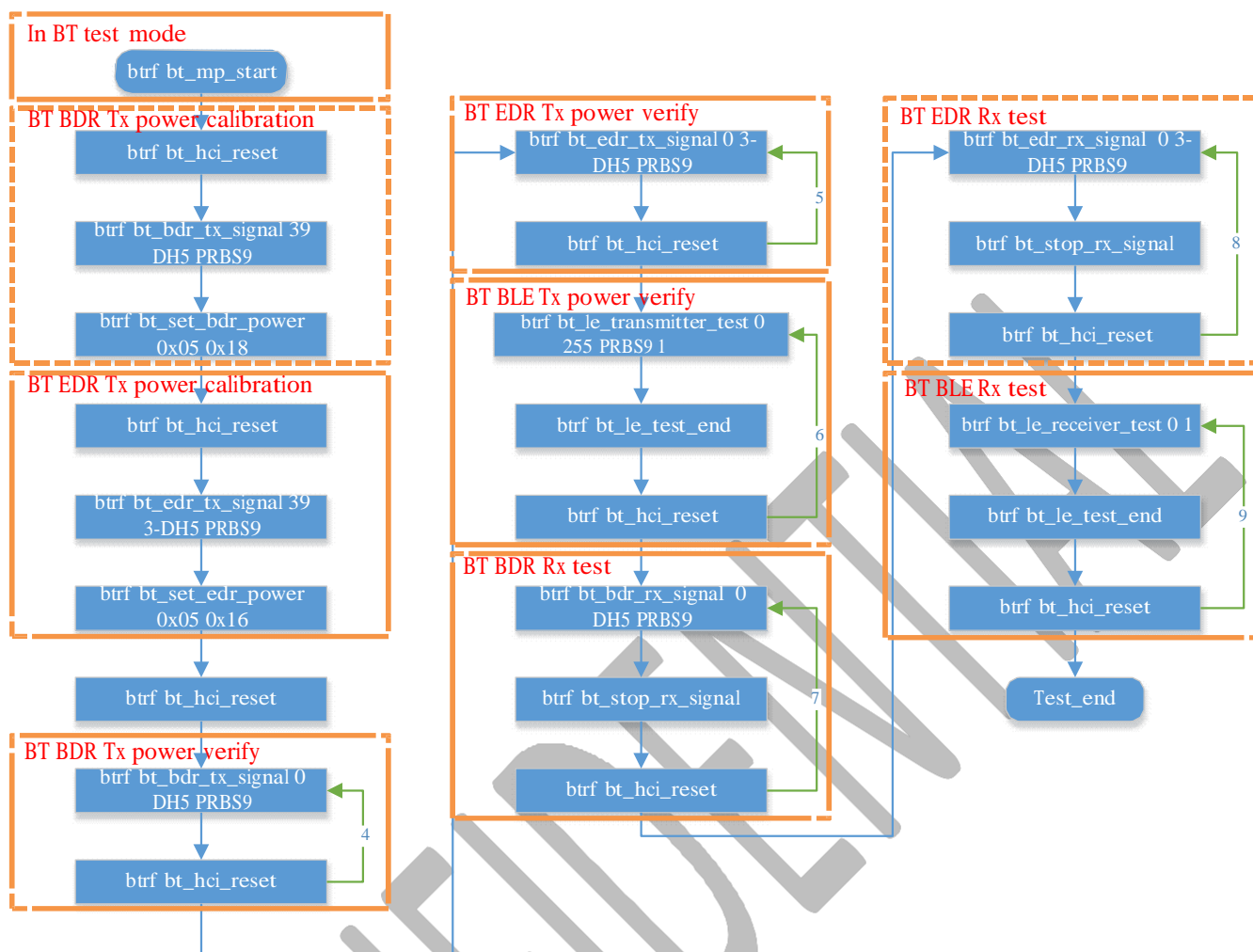
2.9	BT EDR Rx Verify .....	15
2.10	BLE Rx Verify .....	15
3.	Wi-Fi 信令测试 .....	16
4.	BT 信令测试 .....	17
4.1	BDR/EDR 信令测试 .....	18
4.2	BLE 测试 .....	18
5.	Wi-Fi & BT 单载波 .....	19
5.1	Wi-Fi 单载波 ( DC tone ) .....	19
5.2	BT 单载波 .....	19

CONFIDENTIAL

WI-FI 产测非信令流程图



## BT 非信令产测流程图



## 1. Wi-Fi Tx Calibration And Tx/Rx Verfy

### 1.1 进入非信令测试模式

Waiting for Wi-Fi initial done

Send CMD: `besphy start`

### 1.2 进入 Wi-Fi Tx 非信令测试模式

Waiting for “Wi-Fi\_transq\_init” UART0 log

Send CMD: `besphy tx`

Waiting for “OK\_BES\_RF” UART0 log

Note : 进入测试模式后每发一条 CMD UART0 log 都会有 “OK\_BES\_RF” 返回表示指令执行成功，所以在发下一条指令时都需要等待上一指令返回 “OK\_BES\_RF” 后再发

### 1.3 WI-FI Tx Frequency Calibration ( BT Tx 频偏与 WI-FI 共用校准值 )

如下以 Wi-Fi 11n HT20 MCS7 Long-GI 的 Tx 来进行校准频偏

`besphy start`

`besphy tx`

`besphy bandwidth 20` //Value 20 or 40 · 20 BW20; 40 BW40 · 默认为 20M

`besphy rate 21` // Value: 0~21 rate index 详细见 table1; 11b:0~3, 11g:5~13, 11n:14~21

`besphy short_GI 0` //Value: 0 or 1, 0 Long-GI, 1 short-GI; for only 11n,default long-GI

`besphy channel 7` // BW20: 2.4G channel1~13 · 5G channel36~165  
BW40: 2.4G channel3~11 · 5G channel38~159

`//besphy powerlevel 0x1e00` //Value: 0x0000~0x3fff 调整输出功率这里可以省略这条指令用 driver 默认值

`besphy freqOffset 0xda` //Value: 0x000~0x1ff 改变这个值来调整频偏

`besphy save_freqoffset` //保存校准的频偏值

若需要读取写入的校准值则用如下 CMD :

`besphy get_save_freqoffset` //读取保存校准的频偏值，返回值是 32bit 值，0~8bit 才是写入的校准值  
freqoffset 的步进是不均匀的为非线性，如下图

若需要调整 tx duty cycle :

`besphy tx_interval 1` //Value: 1~8 ms · 这里是 tx packet 的周期

`besphy besphy frame_length 1000` //Values: 11b/g 1000~1999bytes;11n 1000~4095bytes

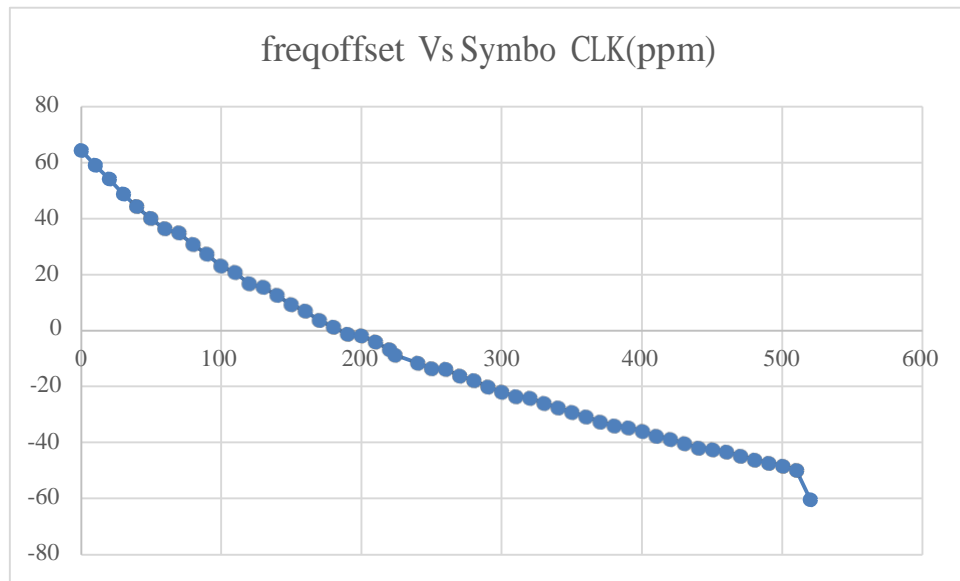


Table 1 Rate Index

Rate Index	Modulation Type	Rate
0	DSSS	CCK1
1	DSSS2	CCK2
2	CCK5.5	CCK5.5
3	CCK11	CCK11
4	CCK22	CCK22(No Support)
5	CCK33	CCK33(No Support)
6	BPSK 1/2	OFDM6
7	BPSK 3/4	OFDM9
8	QPSK 1/2	OFDM12
9	QPSK 3/4	OFDM18
10	16QAM 1/2	OFDM24
11	16QAM 3/4	OFDM36
12	64QAM 2/3	OFDM48
13	64QAM 3/4	OFDM54
14	BPSK 1/2	MCS0
15	QPSK 1/2	MCS1
16	QPSK 3/4	MCS2
17	16QAM 1/2	MCS3
18	16QAM 3/4	MCS4
19	64QAM 2/3	MCS5
20	64QAM 3/4	MCS6
21	64QAM 5/6	MCS7

## 1.4 Wi-Fi Tx Power Calibration

在频偏校准完成后进入 Tx 功率校准，Tx 功率校准只需要校准 CCK11/OFDM54/HT20-MCS7 的 1/7/13 信道，11n 下 HT40 的功率等于 HT20 的功率，所以只需要校准 HT20-MCS7 即可

### 1.4.1 CCK11 CH1 功率校准

```
besphy rate 3           //设定速率为 CCK11
besphy channel 1        //设定信道为 1
besphy powerlevel 0x2000 //设定数字增益
...
besphy powerlevel 0x2500 //调整数字增益至输出功率为目标功率
besphy save              //保存上面校准的数字增益
```

### 1.3.2 OFDM54 CH7 功率校准

```
besphy rate 13          //设定速率为 OFDM54
besphy channel 7        //设定信道为 7，若要 5G 直接设定 5G 对应信道即可
besphy powerlevel 0x1f00 //设定数字增益
...
besphy powerlevel 0x2100 //调整数字增益至输出功率为目标功率
besphy save              //保存上面校准的数字增益
```

### 1.3.3 HT20-MCS7 CH13 功率校准

```
besphy rate 21          //设定速率为 MCS7
besphy channel 13       //设定信道为 13，若要 5G 直接设定 36~165 对应信道即可
besphy powerlevel 0x1a00 //设定数字增益
...
besphy powerlevel 0x1e00 //调整数字增益至输出功率为目标功率
besphy save              //保存上面校准的数字增益 其他信道都
```

是同样的方法

功率校准完成保存后可通过如下 CMD 读取

```
besphy get_save          //读取保存数字增益
```

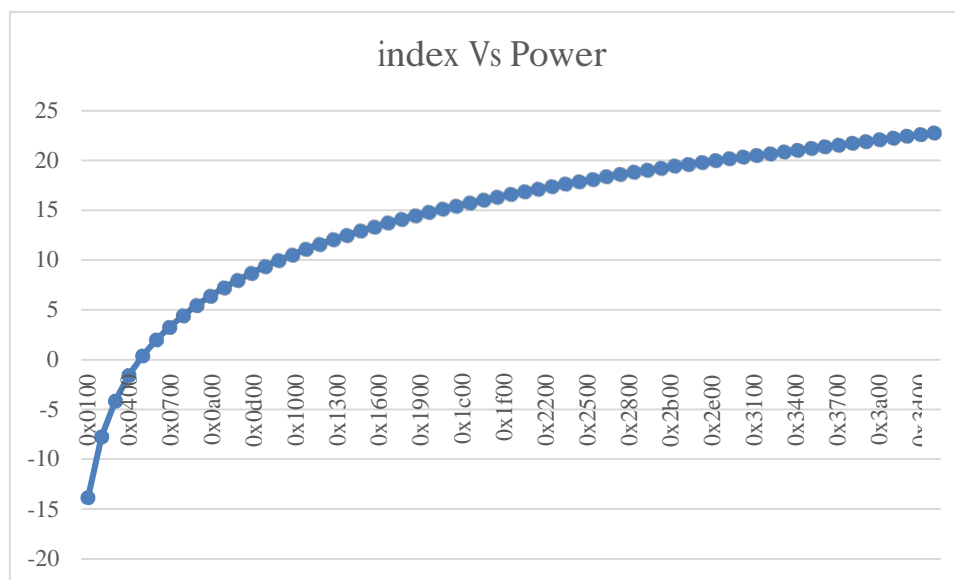
信道在 2.4G 下返回打印内容如下：

```
OK_BES_RF=====>power_table[0]=0x000026e6    //CCK11  CH1~CH2
OK_BES_RF=====>power_table[1]=0x00002736    //CCK11  CH5~7
OK_BES_RF=====>power_table[2]=0x000028da    //CCK11  CH11~13
OK_BES_RF=====>power_table[3]=0x00001ef4    //OFDM54  CH1~CH2
OK_BES_RF=====>power_table[4]=0x00001f44    //OFDM54  CH5~7
OK_BES_RF=====>power_table[5]=0x00002066    //OFDM54  CH11~13
OK_BES_RF=====>power_table[6]=0x00001abe    //HT20&HT40 MCS7  CH1~CH2
OK_BES_RF=====>power_table[7]=0x00001ad2    // HT20&HT40MCS7  CH5~7
OK_BES_RF=====>power_table[8]=0x00001c08    // HT20&HT40MCS7  CH11~13
```



Note: CH3~CH4=(CH1~CH2+CH5~CH7)/2 ; CH8~CH10=(CH5~CH7+CH11~CH13)/2

每个数字增益之间的功率差不是固定的，为非线性



信道在 5G 下返回打印内容如下：

```
OK_BES_RF==>power_table_G[0]=0x1500, power_table_N[0]=0x1300 //G[0]:OFDM54;N[0]:MCS7 CH36~40
OK_BES_RF==>power_table_G[1]=0x1800, power_table_N[1]=0x1500 //G[0]:OFDM54;N[0]:MCS7 CH44~48
OK_BES_RF==>power_table_G[2]=0x1800, power_table_N[2]=0x1600 //G[0]:OFDM54;N[0]:MCS7 CH52~56
OK_BES_RF==>power_table_G[3]=0x1700, power_table_N[3]=0x1500 //G[0]:OFDM54;N[0]:MCS7 CH60~64
OK_BES_RF==>power_table_G[4]=0x1b00, power_table_N[4]=0x1900 //G[0]:OFDM54;N[0]:MCS7 CH100~104
OK_BES_RF==>power_table_G[5]=0x1900, power_table_N[5]=0x1700 //G[0]:OFDM54;N[0]:MCS7 CH108~112
OK_BES_RF==>power_table_G[6]=0x1900, power_table_N[6]=0x1700 //G[0]:OFDM54;N[0]:MCS7 CH116~120
OK_BES_RF==>power_table_G[7]=0x1800, power_table_N[7]=0x1600 //G[0]:OFDM54;N[0]:MCS7 CH124~128
OK_BES_RF==>power_table_G[8]=0x1700, power_table_N[8]=0x1500 //G[0]:OFDM54;N[0]:MCS7 CH132~136
OK_BES_RF==>power_table_G[9]=0x1600, power_table_N[9]=0x1400 //G[0]:OFDM54;N[0]:MCS7 CH140~144
OK_BES_RF==>power_table_G[10]=0x1700, power_table_N[10]=0x1500 //G[0]:OFDM54;N[0]:MCS7 CH149~153
OK_BES_RF==>power_table_G[11]=0x1800, power_table_N[11]=0x1500 //G[0]:OFDM54;N[0]:MCS7 CH157~161
OK_BES_RF==>power_table_G[12]=0x1600, power_table_N[12]=0x1400 //G[0]:OFDM54;N[0]:MCS7 CH165
```

## 1.5 Wi-Fi Tx temp Save

在 Wi-Fi Tx 校准完成后不停止 Tx 条件下立即保存当前的 temp code 值，CMD 如下

```
besphy get_temp           //获取当前实时 temp code
besphy save_temp          //若在 2.4G 信道下保存对应的就是 2.4G 的 tempcode，在 5G 下保存对应的就是 5G 的 tempcode，两个都要保存。
```

若需要读取保存的 temp code，用如下指令

```
besphy get_save_temp      //获取校准保存的 temp code,当前信道是 1~13 获取的就是 2.4G 保存值，当前信道是 36~165 获取的就是 5G 保存值。在所有
```

校准数据保存后需要写入校准完成标志位如下指令，否则校准数据无法生效

```
besphy power_calib_finish //写校准完成标志位,2.4G 校准完写一次，5G 校准完写一次
```

## 1.6 Wi-Fi Tx Verify

在 Wi-Fi 在 Tx Verify 之前需要先通过选 CMD 打开温补

```
besphy tpt_open          //打开温补
```

Wi-Fi Tx Verify 与 Wi-Fi Tx Calibration 在指令上的差异就是省略掉 “besphy powerlevel 0x....” 和 “besphy save” 指令

1.6.1 如下 CCK11 CH1 Tx Verify 指令：

```
besphy rate 3            //设定速率为 CCK11 besphy
channel 1                //设定信道为 1
```

1.6.2 如下 OFDM54 CH7 Tx Verify 指令：

```
besphy rate 13           //设定速率为 OFDM54
besphy channel 7         //设定信道为 7,1~13 or 36~165
```

1.6.3 如下 HT20-MCS7 CH13 Tx Verify 指令：

```
besphy rate 21           //设定速率为 MCS7
besphy channel 13        //设定信道为 13, 1~13 or 36~165
```

1.6.4 如下 MCS7 CH13 Tx Verify 指令：

```
besphy bandwidth 40      //设定带宽为 40M
besphy rate 21           //设定速率为 MCS7
besphy channel 11        //设定信道为 11,3~11 or 38~159
```

Note: 11b/g 以及 11n HT20 都可以不加 besphy bandwidth 这个指令，但 11n HT40 必须要加

## 1.7 Wi-Fi Rx Verify

进入 Rx 测试模式 · 如下指令

```
besphy rx //进入 rx 测试模式
```

Rx 测试模式下不区分速率只区分信道。

**Note:** 因 40M Rx IQ 校准参数需要参考 20M 的，所以在 RX 测试时必须先测完 20M RX 再来测试 40M RX，否则灵敏度会差一些

### 1.7.1 CCK11/OFDM54/HT20-MCS7 CH1 Rx Verify

```
besphy channel 1 //设定信道为 1, 1~13 or 36~165
```

```
besphy rx_ackSet //设置 count=0
```

```
besphy rx_ackRead //获取收到正确的 Packet 数
```

### 1.7.2 CCK11/OFDM54/HT20-MCS7 CH7 Rx Verify

```
besphy channel 7 //设定信道为 7, 1~13 or 36~165 besphy
```

```
rx_ackSet //设置 count=0
```

```
besphy rx_ackRead //获取收到正确的 Packet 数
```

### 1.7.3 CCK11/OFDM54/HT20-MCS7 CH13 Rx Verify

```
besphy channel 13 //设定信道为 13, 1~13 or 36~165 besphy
```

```
rx_ackSet //设置 count=0
```

```
besphy rx_ackRead //获取收到正确的 Packet 数
```

### 1.7.4 HT40-MCS7 CH11 Rx Verify

```
besphy bandwidth 40 //设定带宽为 40M
```

```
besphy channel 11 //设定信道为 11, 3~11 or 38~159 besphy
```

```
rx_ackSet //设置 count=0
```

```
besphy rx_ackRead //获取收到正确的 Packet 数
```

## 2. BT Tx Calibration And Tx/Rx Verfy

### 2.1 Wi-Fi Switch to BT

`besphy rx_stop` //停止 Wi-Fi Rx

`besphy tobt` //切换到 BT

`besphy wifi_stop` //关闭 Wi-Fi

然后进入 BT 测试模式

`btrf bt_mp_start` //进入 bt 测试模式

`btrf ble_config 32` //配置 BLE 增益并保存到 flash, 20 表示 BLE 的功率比 BDR 低 2dB 是海外配置; 若是国内请把 20 改为 32 这样 BLE 和 BDR 的功率是一样的

若要确认这个 BLE 增益是否已正确保存到 flash 可用如下 CMD 读取, 读出来的值要换成十进制

`btrf get_ble_ctry_type`

**Note :** bt 非信令的每项测试再发对应指令前都需要发 `btrf bt_hci_reset`

### 2.2 BT BDR Tx Power Calibration

BT 功率校准都是用 PRBS9 的 Payload · 如下以 DH5 CH39 为例

`btrf bt_hci_reset` //reset bt

`btrf bt_bdr_tx_signal 39 DH5 PRBS9` // 39: channel; DH5: packet type; PRBS9: payload type 详见 table2 和 table3

`btrf bt_set_bdr_power 0x05 0x18` //设定 bt BDR Tx 功率; 0x05: div\_level 保持 0x05 不变

0x18 : 对应的增益 · 可调范围 0x00~0x1f

**Note:** BLE与BDR共用校准值, 校准的时候只需要校准BDR就好了, BLE只需要在后面校验

Table2 : BDR/EDR Packet type Index

Packet type	Index	CMD Code
ID_NULL_TYPE	0x00	ID_NULL
POLL_TYPE	0x01	POLL
FHS_TYPE	0x02	FHS
DM1_TYPE	0x03	DM1
DH1_TYPE	0x04	DH1
DH1_2_TYPE	0x04	2-DH1
DH1_3_TYPE	0x08	3-DH1
HV1_TYPE	0x05	HV1
HV2_TYPE	0x06	HV2
EV3_2_TYPE	0x06	2-EV3
HV3_TYPE	0x07	HV3
EV3_TYPE	0x07	EV3
EV3_3_TYPE	0x07	3-EV3
DV_TYPE	0x08	DV
AUX1_TYPE	0x09	AUX1
DM3_TYPE	0x0A	DM3
DH3_TYPE	0x0B	DH3
DH3_2_TYPE	0x0A	2-DH3
DH3_3_TYPE	0x0B	3-DH3
EV4_TYPE	0x0C	EV4
EV5_2_TYPE	0x0C	2-EV5
EV5_TYPE	0x0D	EV5
EV5_3_TYPE	0x0D	3-EV5
DM5_TYPE	0x0E	DM5
DH5_TYPE	0x0F	DH5
DH5_2_TYPE	0x0E	2-DH5
DH5_3_TYPE	0x0F	3-DH5

Table3 : BDR/EDR Payload Type Index

Payload Type	Index	CMD Code
0x00	0	0x00
ff	1	0xFF
0101101(0x55)	2	0x55
0001111(0x0F)	3	0x0F
prbs9	4	PRBS9

## 2.3 BT EDR Tx Power Calibration

BT 功率校准都是用 PRBS9 的 Payload，如下以 2DH5 CH39 为例

```
btrf bt_hci_reset //reset bt
```

```
btrf bt_edr_tx_signal 39 2-DH5 PRBS9
```

```
btrf bt_set_edr_power 0x05 0x18 //设定 bt EDR Tx 功率; 0x05: div_level 保持 0x05 不变 0x18: 对应的增益, 可调范围 0x00~0x20
```

## 2.4 BT BDR Tx Verify

BT BDR Tx 测试项以 DH5 CH39 为例，列出测试项对应的 CMD 如下 table：

Table4：BDR DH5 CH39 Tx 对应的 CMD

BDR DH5 CH39 Test items		Payload	CMD
TRM/CA/01/C(输出功率)		PRBS9	btrf bt_bdr_tx_signal 39 DH5 PRBS9
TRM/CA/03/C(功率控制)		PRBS9	btrf bt_bdr_tx_signal 39 DH5 PRBS9
TRM/CA/04/C(发射输出频谱—频率范围)		PRBS9	btrf bt_bdr_tx_signal 39 DH5 PRBS9
TRM/CA/05/C(发射输出频谱- 20dB 带宽)		PRBS9	btrf bt_bdr_tx_signal 39 DH5 PRBS9
TRM/CA/06/C(发射输出频谱- 临信道功率)		PRBS9	btrf bt_bdr_tx_signal 39 DH5 PRBS9
TRM/CA/07/C (调制特性)	DeltaF1 Max/Avg	00001111	btrf bt_bdr_tx_signal 39 DH5 0x0F
	DeltaF2 Max/Avg	01010101	btrf bt_bdr_tx_signal 39 DH5 0x55
TRM/CA/08/C (初始载波频率容限)		PRBS9	btrf bt_bdr_tx_signal 39 DH5 PRBS9
TRM/CA/09/C (载波频率漂移)	DH1	PRBS9	btrf bt_bdr_tx_signal 39 DH1 PRBS9
	DH3	PRBS9	btrf bt_bdr_tx_signal 39 DH3 PRBS9
	DH5	PRBS9	btrf bt_bdr_tx_signal 39 DH5 PRBS9

## 2.5 BT EDR Tx Verify

BT EDR Tx 测试项以 2DH5 CH39 为例，列出测试项对应的 CMD 如下 table：

Table5：EDR 2DH5 CH39 Tx 对应的 CMD

EDR 2DH5 CH39 Test items	Payload	CMD
TRM/CA/10/C (EDR 相对发射功率)	PRBS9	btrf bt_edr_tx_signal 39 2-DH5 PRBS9
TRM/CA/11/C (EDR 载波频率稳定度和调制准确度)	PRBS9	btrf bt_edr_tx_signal 39 2-DH5 PRBS9
TRM/CA/12/C (EDR 差分相位编码)	PRBS9	btrf bt_edr_tx_signal 39 2-DH5 PRBS9
TRM/CA/13/C (EDR 带内杂散辐射)	PRBS9	btrf bt_edr_tx_signal 39 2-DH5 PRBS9

## 2.6 BLE Tx Verify

BT BLE Tx 测试项以 CH19 为例，列出测试项对应的 CMD 如下 table：

若要测试 BLE2M 则将指令最后一个数字 1 改为 2 就可切换到 2M，若要测 LE code 则改为 3 或 4

Table6：BLE 1M CH19 Tx 对应的 CMD

BLE CH19 Test items	Payload	CMD
TRM-LE/CA/BV-01-1/C to TRM-LE/CA/BV-01-C (NOC 发射功率)	PRBS9	btrf bt_le_transmitter_test 19 255 PRBS9 1
TRM-LE/CA/BV-01-2/C to TRM-LE/CA/BV-02-C (EOC 发射功率)	PRBS9	btrf bt_le_transmitter_test 19 255 PRBS9 1
TRM-LE/CA/BV-02-1/C to TRM-LE/CA/BV-03-C (NOC 带内杂散)	PRBS9	btrf bt_le_transmitter_test 19 255 PRBS9 1
TRM-LE/CA/BV-02-2/C to TRM-LE/CA/BV-04-C (EOC 带内杂散)	PRBS9	btrf bt_le_transmitter_test 19 255 PRBS9 1
TRM-LE/CA/BV-03-C to TRM-LE/CA/BV-05-C (调制特性)	DeltaF1 Max/Avg 00001111(0F)	btrf bt_le_transmitter_test 19 255 0x0F 1
	DeltaF2 Max/Avg 01010101(55)	btrf bt_le_transmitter_test 19 255 0x55 1
TRM-LE/CA/BV-04-1/C to TRM-LE/CA/BV-06-C	01010101(55)	btrf bt_le_transmitter_test 19 255 0x55 1
TRM-LE/CA/BV-04-2/C to TRM-LE/CA/BV-07-C	01010101(55)	btrf bt_le_transmitter_test 19 255 0x55 1

BLE 1MTx Power 以 CH19 为例：

```

btrf bt_hci_reset //reset bt

btrf bt_le_transmitter_test 19 255 PRBS9 1 // 19: channel19; 255:datalength Bytes;
PRBS9:payload; 1: BLE 1M

btrf bt_le_test_end //停止 ble 测试

```

BLE 2MTx Power 以 CH19 为例：

```

btrf bt_hci_reset //reset bt

btrf bt_le_transmitter_test 19 255 PRBS9 2 // 19: channel19; 255:datalength Bytes;
PRBS9:payload; 2: BLE 2M

btrf bt_le_test_end //停止 ble 测试

```

BLE code s=8 Tx Power 以 CH19 为例：

```

btrf bt_hci_reset //reset bt

btrf bt_le_transmitter_test 19 255 PRBS9 3 // 19: channel19; 255:datalength Bytes;
PRBS9:payload; 3: le code s=8

btrf bt_le_test_end //停止 ble 测试

```

BLE code s=2 Tx Power 以 CH19 为例：

```

btrf bt_hci_reset //reset bt

btrf bt_le_transmitter_test 19 255 PRBS9 4 // 19: channel19; 255:datalength Bytes;
PRBS9:payload; 4: le code s=2

btrf bt_le_test_end //停止 ble 测试

```

table7: BLE Payload type index

Payload Type	Index	CMD Code
PRBS9	0x00	PRBS9
11110000(0xF0)	0x01	0xF0
10101010(0xAA)	0x02	0xAA
PRBS15	0x03	PRBS15
11111111(0xFF)	0x04	0xFF
00000000(0x00)	0x05	0x00
00001111(0x0F)	0x06	0x0F
01010101(0x55)	0x07	0x55



## 2.7 BDR Rx Verify

BT Rx 协议规范 Payload 均使用 PRBS9。以 DH5 CH0 Rx 为例，指令如下：

`btrf bt_hci_reset` //reset bt

`btrf bt_bdr_rx_signal 0 DH5 PRBS9`

等待仪器发包完成后输入如下指令

`btrf bt_stop_rx_signal` //停止 Rx，获取收包信息

如下输入 `bt_stop_rx_signal` 指令查看收到的正确包的输量是 `0x03ec(hex2dec)=1004`

```
cmd_bt_stop_rx_signal ...
355617/I/NONE / 8 | hci_tx_p 0x20066df0 hci_rx_p 0x2007158c hci_rx_len 0x20071588
01 87 fc 1c 02 e8 03 00 00 00 00 03 55 55 55 55 00 00 01 02 0f 04 fd 03 00 00 00 00 ff ff ff f
HciRxBuffLen 21
04 0e 12 05 87 fc 00 ec 03 04 00 3e 00 0f ff 30 03 00 00 00 00
success:0 receive packet num length:1004
head_error:4
payload_error:62
adv_estsw:65295
adv_esttpt:816
payload_bit_error:0
success
```

*03ec(hex2dec)=1004*

以 R&S 仪器为例，通过 `payload_error` 和 `header_error` 来判断 Rx Sensitivity。即 PLER，

$PLER = \frac{(\text{payload\_error} + \text{header\_error})}{\text{仪器发包数}} < 6.5\%$

同时要求 PER<80%

也就是发 1000 个包，`payload_error+header_error`<65，同时要求收到的包数量>

## 2.8 BT EDR Rx Verify

BT EDR Rx 以 2DH5 CH0 Rx 为例，指令如下：

`btrf bt_hci_reset` //reset bt

`btrf bt_edr_rx_signal 0 2-DH5 PRBS9`

// 0: channel0; 2DH5: paket type; PRBS9: payload type

等待仪器发包完成后输入如下指令

`btrf bt_stop_rx_signal` //停止 Rx，获取收包信息

EDR 统计包的方式和 BDR 一样，参考 BDR

## 2.9 BLE Rx Verify

BLERx Sensitivity 按照协议标准 PER<30.8%,测试用的 Payload 为 PRBS9 37bytes, 以 CH0 为例, 指令如下

```
btrf bt_hci_rest //reset bt
btrf bt_le_receiver_test 0 1 // 0: channel0; 1: LE 1M 若要测试 LE 2M 将 1 改为 2 就可以了,
3: LE code(包含 S2 和 S8)
```

等待仪器发包完成后输入如下指令

```
btrf bt_le_test_end //停止 Rx, 获取收包信息
```

如下为收到包数

新版软件统计方式

```
bt_le_test_end
[proc_onecmd] ==> enter
01 1f 20 00
HciRxBuffLen 9
04 0e 06 05 1f 20 00 e6 03
success:0 length 998
[proc_onecmd] ==> leave
```

0x03e6(hex2dec)=998

旧版软件统计方式

```
bt_le_test_end
01 1f 20 00
HciRxBuffLen 9
04 0e 06 05 1f 20 00 e7 03
success:0 length:0 param:
```

0x03e7(hex2dec)=999

## 3. WIFI BT MAC 地址写入

```
AT+SETWMAC=00:11:22:33:44:55 //写入 WiFi MAC
```

```
AT+ SETBTMAC=00:11:22:33:44:55 //写入 BT MAC
```

### 3. Wi-Fi 信令测试

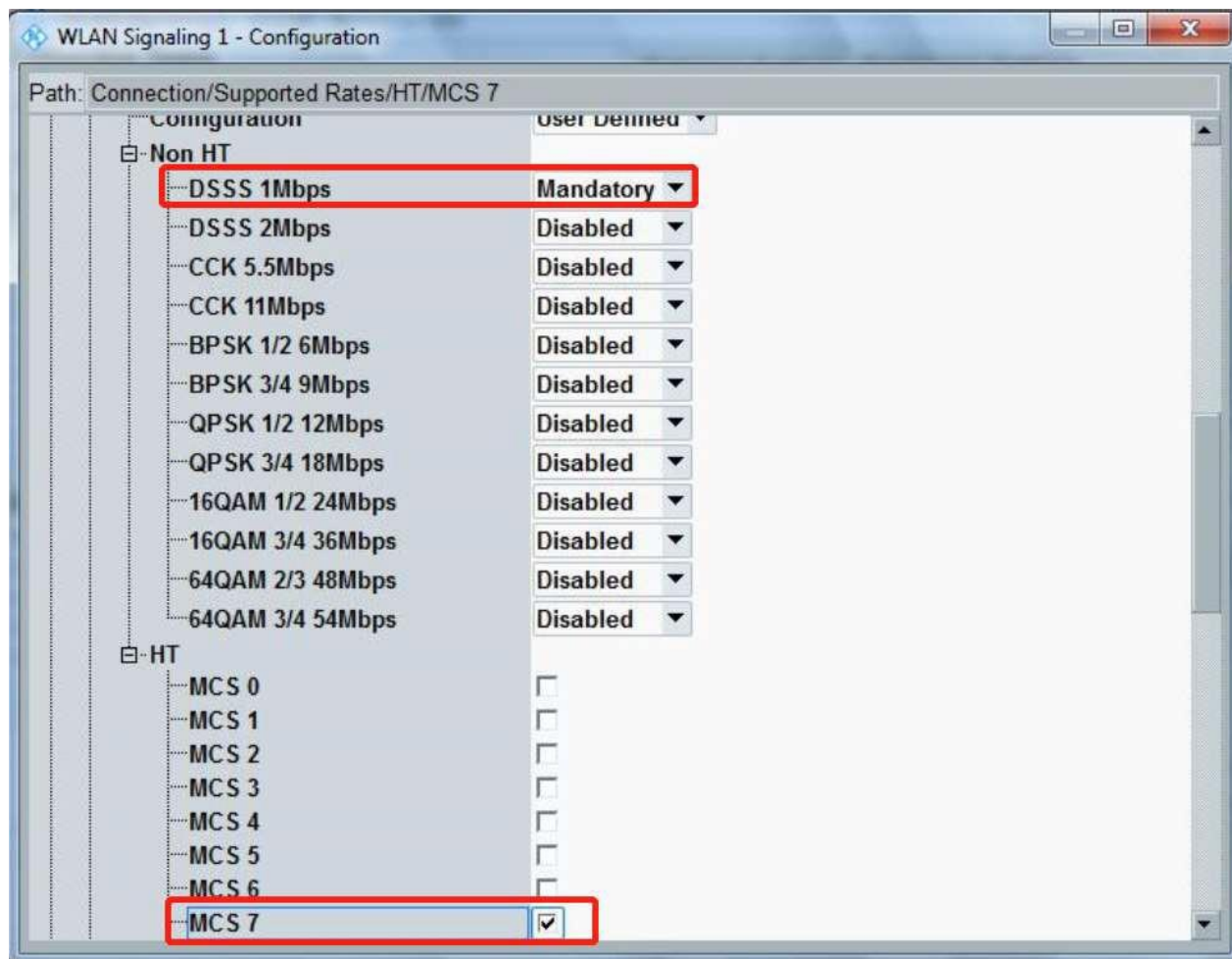
- (1) 进行 Wi-Fi 配网
- (2) 关闭 BT，输入如下指令

```
btrf bt_mp_start
```

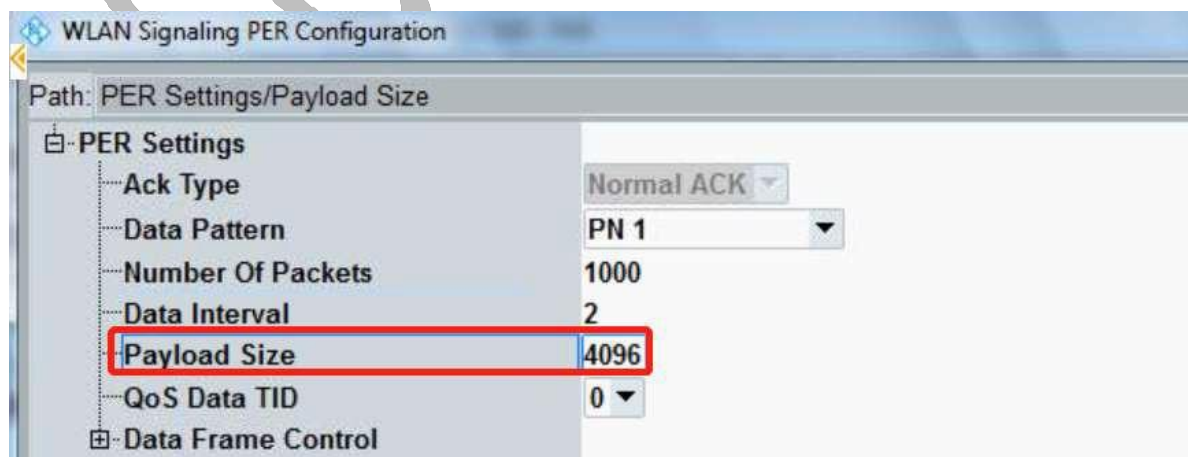
```
btrf bt_hci_reset
```

```
btrf bt_scan off
```

- (3) CMW 固定 Rate 注意 11n 下 support rate 打开所需测试 Rate 的同时要打开 CCK1，OTA 测试打开 Basic Rate。如下 MCS7 同时打开 MCS7 和 CCK1



- (4) Rx 测试注意关闭 PG,同时 Payload size 参照协议设定，11b：1024，11g：1000；11n：4096



## 4. BT 信令测试

### 4.1 BDR/EDR 信令测试

besphy start

btrf bt\_mp\_start

btrf bt\_enter\_signal\_mode

先进行 BT 连接，连接成功后再输入如下指令关闭扫描和广播，关掉扫描和广播后会搜索不到 BT

btrf bt\_scan Off

### 4.2 BLE RS232 UART 非信令测试

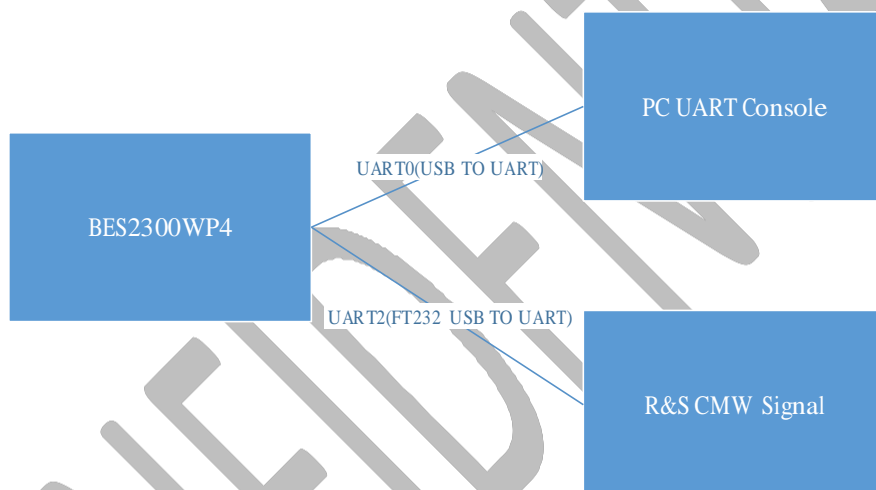
besphy start

btrf bt\_mp\_start

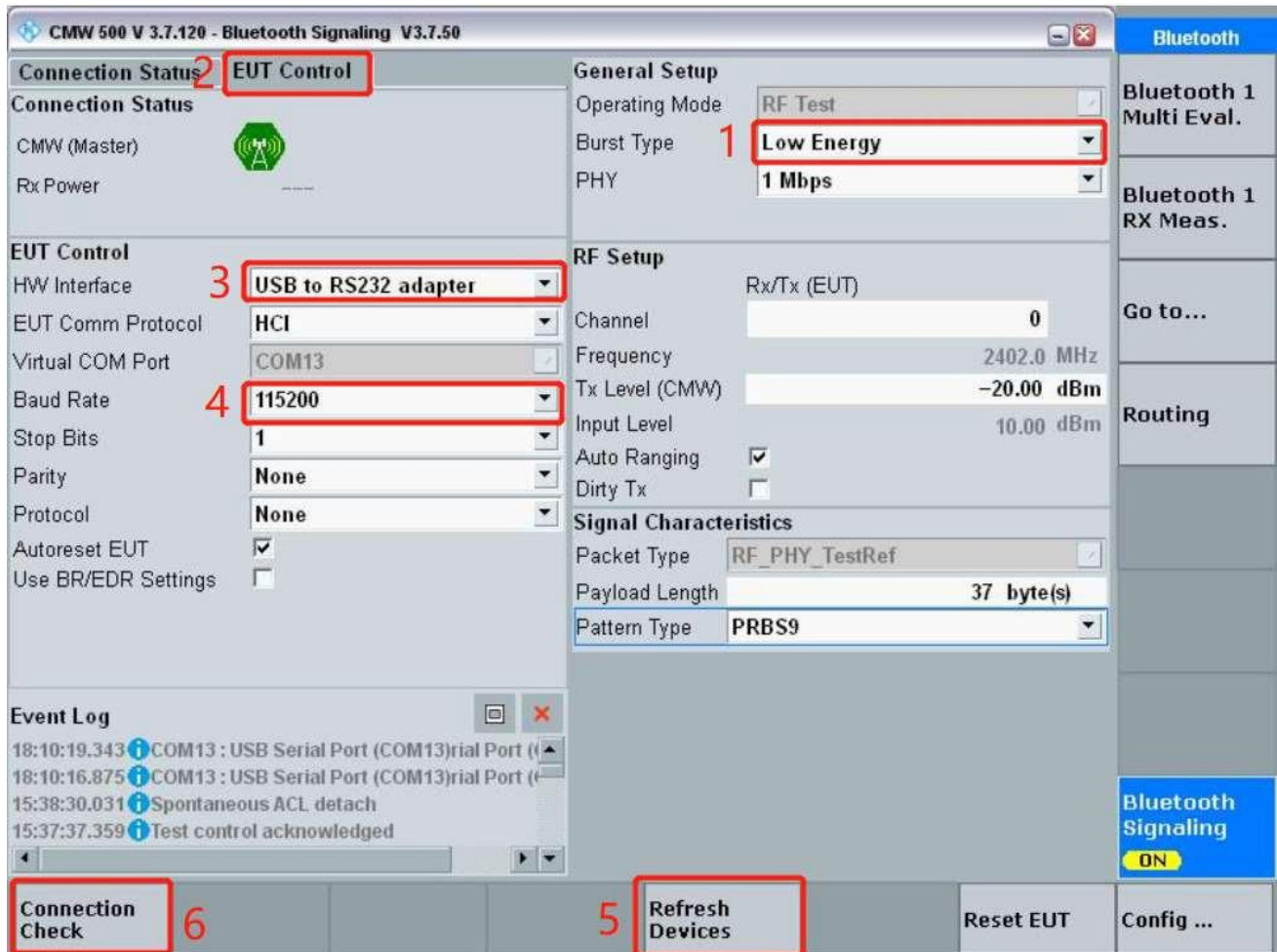
btrf bt\_hci\_reset

btrf ble\_nosignal\_start

BLE 测试上述指令从 BES2300WP4 UART0 输入。BLE 测试需要接 UART2 (这里也有可能是 UART1，需要根据软件定义接口来连接)



进入 BLE 测试模式后设置仪器如下



## 5. Wi-Fi & BT 单载波

### 5.1 Wi-Fi 单载波 ( DC tone )

besphy start

besphy tx

besphy channel 7

besphy tx\_dc\_tone

### 5.2 BT 单载波

上电后输入如下指令，不要输入 besphy start，等待启动完成，出现<I>WebSocket.cpp<GnLwsSt>

[mainLwsLoop:859]mCurStatus: 4, getWi-FiStatus(): 2, thiz->mContextPointer->getAuthStatus(): 0 这个打印后再发下面的指令：

btrf bt\_mp\_start

besphy write 0xd0220c00 0xa0000 //0xa0000 是 channel 0, 0xa0027 是 channel 39,

0xa004e 是 channel 78

besphy write 0x20d1 0x00