



AIC8800 射频测试说明

RF_TEST版本

版本号 v8.5

2021-9-8

爱科微半导体（上海）有限公司



公司	爱科微半导体（上海）有限公司 AIC Semiconductor (Shanghai) CO., Ltd.	
版本信息	日期	Release note
V8.2	2021 年 7 月 5 日	
V8.3	2021 年 7 月 8 日	增加 aic_userconfig.txt
V8.4	2021 年 8 月 6 日	增加 set_papr,
V8.5	2021 年 9 月 8 日	增加 set_txtone, 更新 rdwr_pwridx, rdwr_pwrofst, aic_userconfig
V8.5.1	2021 年 10 月 14 日	修改 BT 测试接口编译说明
V8.5.1	2021 年 11 月 9 日	增加读 drvibit 指令
V8.5.2	2022 年 4 月 8 日	更新测试指令



目录

一. 工具介绍.....	3
二. RF_TEST 测试指令.....	4
2.1 WIFI 部分.....	4
2.1.1 WiFi 测试指令.....	4
2.1.2 晶体频偏校准指令.....	5
2.1.3 读写 mac 地址.....	6
2.1.4 TX power 增益表档位设置.....	7
2.1.5 信道功率补偿.....	8
2.1.6 WiFi2.4G EVM 优化.....	9
2.1.7 userconfig 使用.....	10
2.2 BT 部分 (BT 走 USB 接口).....	11
2.2.1 BT TX 测试指令.....	11
2.2.2 BT RX 测试指令.....	12
三. RF_TEST 编译说明.....	14
四. BT 测试接口编译说明.....	16
五. BT 走 UART 接口测试方法.....	17
5.1. BT TX (非信令).....	18
5.2. BT RX (非信令).....	19
5.3. BLE 测试 (目前只支持 BT 走 UART 接口测试模式).....	20
5.3.1 TX, LE Transmitter Test command.....	20
5.3.2 RX, LE Receiver Test command.....	21



一. 工具介绍

适用于 linux (ubuntu /android)

fmacfw.bin用于正常模式, fmacfw_rf.bin用于测试模式

以下以ubuntu为例, 用户界面输入测试命令: (以下命令均以 wlan0 为例, 实际以 ifconfig 显示为准) 格式
wifi_test if_name command parameters

COMMAND:

enum {

1. SET_TX,
 2. SET_TXSTOP,
 3. SET_RX,
 4. SET_RXSTOP,
 5. GET_RX_RESULT,
 6. SET_XTAL_CAP,
 7. SET_XTAL_CAP_FINE,
 8. SET_FREQ_CAL,
 9. SET_FREQ_CAL_FINE,
 10. GET_FREQ_CAL,
 11. GET_EFUSE_BLOCK,
 12. SET_TXTONE
 13. SET_MAC_ADDR,
 14. GET_MAC_ADDR,
 15. SET_BT_MAC_ADDR,
 16. GET_BT_MAC_ADDR,
 17. RDWR_PWRIDX,
 18. RDWR_PWROFST,
 19. RDWR_EFUSE_PWROFST,
 20. RDWR_DRVIBIT,
 21. RDWR_EFUSE_DRVIBIT,
 22. AIC_USERCONFIG.TXT,
 23. SET_PAPR

 24. BT_RESET,
 25. BT_TXDH,
 26. BT_RXDH,
 27. BT_STOP,
 28. GET_BT_RX_RESULT
- };



二. RF_TEST测试指令

2.1 WIFI部分

2.1.1 WiFi测试指令

1. wifi_test wlan0 set_tx chan bw mode rate length \\ WiFi 发射测试开始

1-1-1: channel

		Chan_num
2.4G	1-13	ch1-ch13
5.8G	36-64	Ch36-ch64
	100-144	Ch100-ch144
	149-165	Ch149-ch165

1-1-2: bandwidth

	bw
0	20M
1	40M

1-1-3: mode 和 rate 对应关系

	mode	rate											
0	NON HT	0	1	2	3	4	5	6	7	8	9	10	11
		1M	2M	5.5M	11M	6M	9M	12M	18M	24M	36M	48M	54M
2	HT MF	0-7											
		mcs0-7											
4	VHT	0-9											
		mcs0-9											
5	HE SU	0-11											
		mcs0-11											

eg: wifi_test wlan0 set_tx 1 0 2 7 1500

\\ 设置信道1,20M带宽,HT MF模式,速率mcs7,长度1500

2. wifi_test wlan0 set_txstop
no parameter

\\ WiFi发射测试停止

3. wifi_test wlan0 set_rx chan_num bw

\\ WiFi接收测试开始

chan_num (见1-1-1 channel)

bw (见1-1-2 bandwidth)

eg: wifi_test wlan0 set_rx 1 0

\\ 设置信道1, 带宽20M

4. wifi_test wlan0 set_rxstop
no parameter

\\ WiFi接收测试停止

5. wifi_test wlan0 get_rx_result
no parameter

\\ WiFi 接收测试收到的包的个数

返回参数: 从 SET_RX 到 SET_RXSTOP 这段时间内接收到总的数据包的个数



2.1.2 晶体频偏校准指令

AIC8800M/D XTAL 电路内部提供了可变负载电容，最大支持负载电容为 7pF 的 crystal unit。若采用晶体负载电容大于 7pF，需要板上预留晶体负载电容。

本校准流程做如下假设：晶体负载电容不大于 7pF；如果晶体负载电容大于 7pF，PCB 上已经刚好补齐所缺部分负载电容。例如，晶体所需负载电容为 10pF，PCB 上给晶体两端都提供了一个 6pF 的片外电容（等效于 3pF 负载电容）。

1. `wifi_test wlan0 set_xtal_cap val` \\晶体频偏粗调，默认值24 (0x18)，范围0-31 (0x00~0x1F)
val: 十进制有符号数

eg: `wifi_test wlan0 set_xtal_cap -2` \\ 负向频偏，降低内部负载电容
2. `wifi_test wlan0 set_xtal_cap_fine val` 晶体频偏细调，默认值31 (0x1F)，范围0-63 (0x00~0x3F)
val: 十进制有符号数

eg: `wifi_test wlan0 set_xtal_cap_fine 10` \\ 正向频偏，提高内部负载电容
3. `wifi_test wlan0 set_freq_cal val` \\ 写晶体频偏校准粗调值到efuse\flash
val 十六进制绝对值

eg: `wifi_test wlan0 set_freq_cal 1a` \\ 写晶体频偏校准粗调值 0x1A 到 efuse\flash
4. `wifi_test wlan0 set_freq_cal_fine val` \\写晶体频偏校准细调值到efuse\flash
val: 十六进制绝对值

eg: `wifi_test wlan0 set_freq_cal_fine 16` \\ 写晶体频偏校准细调值0x16到efuse\flash
5. `wifi_test wlan0 get_freq_cal` \\ 读频偏值
no parameter

粗调校准流程：

- ①判断 frequency offset (Δf) 极性， $\Delta f > 0$ ，setxtalcap 4，反之，setxtalcap -4；
- ②判断 frequency offset (Δf) 极性， $\Delta f > 0$ ，setxtalcap 2，反之，setxtalcap -2；
- ③判断 frequency offset (Δf) 极性， $\Delta f > 0$ ，setxtalcap 1，反之，setxtalcap -1；

细调校准流程：

- ①判断 frequency offset (Δf) 极性， $\Delta f > 0$ ，setxtalcapfine 16，反之，setxtalcapfine -16；
- ②判断 frequency offset (Δf) 极性， $\Delta f > 0$ ，setxtalcapfine 8，反之，setxtalcapfine -8；
- ③判断 frequency offset (Δf) 极性， $\Delta f > 0$ ，setxtalcapfine 4，反之，setxtalcapfine -4；
- ④判断 frequency offset (Δf) 极性， $\Delta f > 0$ ，setxtalcapfine 2，反之，setxtalcapfine -2；
- ⑤判断frequency offset (Δf) 极性， $\Delta f > 0$ ，setxtalcapfine 1，反之，setxtalcapfine -1；

Note: 校准频偏指令对应参数均为十进制相对值，即相对默认值偏移值，输入指令后会返回配置后频偏实际参数，且以十六进制显示。写入efuse或flash的频偏校准值为十六进制绝对值



2.1.3 读写mac地址

1. wifi_test wlan0 set_mac_addr \\写WiFi MAC地址到efuse(2次)或flash(重复)
eg: wifi_test wlan0 set_mac_addr 88 00 11 22 33 44 \\写WiFi MAC地址
2. wifi_test wlan0 get_mac_addr \\读WiFi MAC地址
no parameter
3. wifi_test wlan0 set_bt_mac_addr \\写BT MAC地址到efuse(2次)或flash(重复)
eg: wifi_test wlan0 set_bt_mac_addr 0A 1C 6B C6 96 7E \\写BT MAC地址
4. wifi_test wlan0 get_bt_mac_addr \\读BT MAC地址
no parameter
5. wifi_test wlan0 get_efuse_block val \\读efuse
val:0~15
6. wifi_test wlan0 set_txtone val \\tx单tone
val: 0 关闭
val: 1 val 打开（1后面的参数范围-20 - 19）

0 关闭	no parameter		
1 打开	-20 - -1	0	1-19
	负偏	中心偏点	正偏

eg: wifi_test wlan0 set_txtone 1 1 \\打开正向偏1M
eg: wifi_test wlan0 set_txtone 0 \\关闭



2.1.4 TX power增益表档位设置

1. wifi_test wlan0 rdwr_pwridx band mod idx

\\ 设置功率增益档位

4-1-1: band

	band
2.4G	1
5.8G	2

4-1-2: mod

mod	
OFDM-LowRate	0
OFDM-64QAM	1
OFDM-256QAM	2
1024QAM	3
DSSS	4

Tx_power 增益表

type\idx	0	1	2	3	4	5	6	7	8	9	10	11
OFDM-LowRate	-2	0	2	4	6	8	10	12	14	16	18	22
OFDM-64QAM	-2	0	2	4	6	8	10	12	14	16	18	
OFDM-256QAM	-2	0	2	4	6	8	10	12	14	16	18	
OFDM-1024QAM	-2	0	2	4	6	8	10	12	14	15		
DSSS	1	3	5	7	9	12	14	16	18	20	22	23

Figure2 -1 2.4G 发射增益表

type\idx	0	1	2	3	4	5	6	7	8	9	10	11
OFDM-LowRate	-5	-3	-1	1	3	5	7	9	11	11	15	18
OFDM-64QAM	-5	-3	-1	1	3	5	7	9	11	13	13	
OFDM-256QAM	-5	-3	-1	1	3	5	7	9	11	13	13	
OFDM-1024QAM	-5	-3	-1	1	3	5	7	9	11	11		

Figure 2-2 5.8G 发射增益表

eg. wifi_test wlan0 rdwr_pwridx 1 4 11

\\ 设置 2.4G DSSS 模式发射增益档位为 11

Note: pwridx 后面不带参数可直接显示当前发射功率增益档位配置信息

Note: 档位 11 专门为低速率提供的高功率档，不能在该档位做校准，并且开启该档位后，信道补偿不再有效。

4-1-3: OFDM Rate 分类

	OFDM-LowRate						64QAM			256QAM		1024QAM	
	BPSK 1/2	BPSK 3/4	QPSK 1/2	QPSK 3/4	16QAM 1/2	16QAM 3/4	64QAM 2/3	64QAM 3/4	64QAM 5/6	256QAM 3/4	256QAM 5/6	1024QAM 3/4	1024QAM 5/6
11a/g	6M	9M	12M	18M	24M	36M	48M	54M					
11n	MCS0		MCS1	MCS2	MCS3	MCS4	MCS5	MCS6	MCS7				
11ac	MCS0		MCS1	MCS2	MCS3	MCS4	MCS5	MCS6	MCS7	MCS8	MCS9		
11ax	MCS0		MCS1	MCS2	MCS3	MCS4	MCS5	MCS6	MCS7	MCS8	MCS9	MCS10	MCS11



2.1.5 信道功率补偿

1. `wifi_test wlan0 rdwr_pwrofst band ch ofst` \\ 设置信道补偿

5-1-1: band\ch\ofst 对应关系表

	band		ch	ofst
2.4G	1	CH1~CH4	0	-15~15
		CH5~CH9	1	-15~15
		CH10~CH13	2	-15~15
5.8G	2	CH36~CH64	0	-15~15
		CH100~CH120	1	-15~15
		CH122~CH140	2	-15~15
		CH142~CH165	3	-15~15

eg. `wifi_test wlan0 rdwr_pwrofst 1 1 2` \\设置 CH5~CH9 信道补偿为 2

ofst 为带符号偏移值，步进为 1，对应功率变化 0.375dbm，最大 15，最小-15，可通过调整响应信道补偿值来优化信道功率差异。

Note: pwrofst 后面不带参数可直接显示当前发射功率增益档位配置信息。

2. `wifi_test wlan0 rdwr_efuse_pwrofst band ch ofst` \\ 写信道补偿值到efuse（2次）或flash（重复）

5-2-1: band\ch\ofst 对应关系表

	band		ch	ofst
2.4G	1	CH1~CH4	0	-15~15
		CH5~CH9	1	-15~15
		CH10~CH13	2	-15~15
5.8G	2	CH36~CH64	0	-15~15
		CH100~CH120	1	-15~15
		CH122~CH140	2	-15~15
		CH142~CH165	3	-15~15

eg. `wifi_test wlan0 rdwr_efuse_pwrofst 1 2 5` \\写2.4G CH10~CH13校准值到efuse

Note: efpwrofst 后不加参数能读取 efuse 中信道功率补偿值。



2.1.6 WiFi2.4G EVM优化

1. wifi_test wlan0 rdwr_drvibit 1 val
1:2.4g
val:1~7

\\优化 WIFI 2.4G EVM

eg: wifi_test wlan0 rdwr_drvibit 1 3

\\ drvibit值配置成3

rdwr_drvibit 0

\\读配置的drvibit

Note: rdwr_drvibit后加 0 或不加参数实现读功能

2. wifi_test wlan0 rdwr_efuse_drvibit 1 val

\\ drvibit配置值写入efuse/flash,只可写一次

eg: wifi_test wlan0 rdwr_efuse_drvibit 1 3

rdwr_efuse_drvibit 0

\\ 读写入efuse/flash的drvibit值

Note: 遇到2.4G EVM大于-35db, 不满足msc11 EVM指标的状况, 可以调此配置来优化, 先写4, 仍然不满足指标的话继续按step 1向下调, 直至EVM达标。如果向下调EVM变差, 则反向调节, 直至EVM满足指标, 将值写入efuse/flash。

3. wifi_test wlan0 set_papr val
val:0,1,2

\\配置papr filter, 优化边带抑制

6-3-1:

	0	1	2
Papr	Disable	Normal	Strengthen

eg:papr 1

\\ papr filter 设置成normal

note: papr filter开启后, 切换带宽需要重新配置papr设置, 默认为disable, 该配置会对EVM有一定的程度影响, 需根据实际测试需求来配置



2.1.7 userconfig 使用

1. aic_userconfig.txt 文档使用:

随固件一起 cp 到 /lib/firmware/下，更改文档内参数后掉电重新上电生效

enable = 0 文档不生效，enable = 1 文档生效，默认为1

(参数意义可以详见上述2.1.4、2.1.5)

```
#txpwr_idx
Enable=1
Dsss=9
Ofdmlowrate_2g4=10
Ofdm64qam_2g4=10
Ofdm256qam_2g4=9
Ofdm1024qam_2g4=8
Ofdmlowrate_5g=10
Ofdm64qam_5g=9
Ofdm64qam_5g=9
Ofdm1024qam_5g=8
```

```
Txpwr_ofst
Ofst_enable=1
Ofst_chan_1_4=0
Ofst_chan_5_9=0
Ofst_chan_10_13=0
Ofst_chan_36_64=0
Ofst_chan_100_120=0
Ofst_chan_122_140=0
Ofst_chan_142_165=0
```



2.2 BT部分（BT走USB接口）

*** 输入参数均为 16 进制 ***

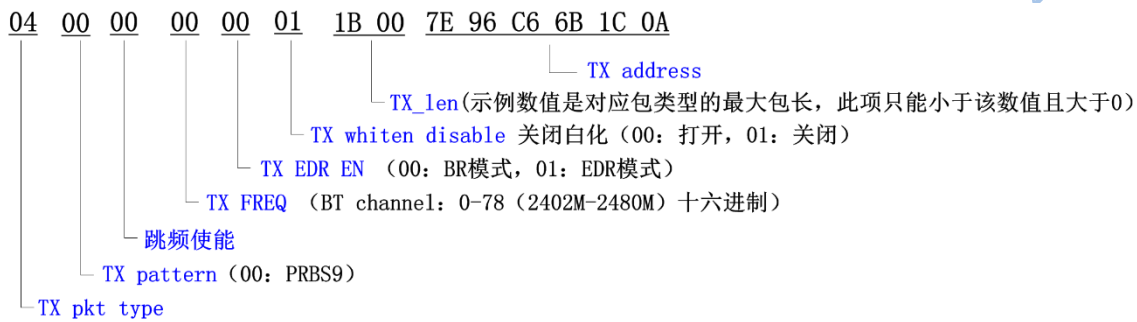
1. wifi_test wlan0 bt_reset \\ BT开始测试
no parameter

2.2.1 BT TX测试指令

1. wifi_test wlan0 bt_txdh \\ BT发射测试

eg: wifi_test wlan0 bt_txdh 04 00 00 00 00 01 1b 00 7E 96 C6 6B 1C 0A

note: tx DH1 包为例:



note: Package type与max len、Pattern、Tx address对应关系参考图2-2-1、2-2-2、2-2-3

bt_tx 非信令测试模式 hci cmd 示例

04 00 00 00 00 01 1B 00 7E 96 C6 6B 1C 0A	DH1
0B 00 00 00 00 01 B7 00 7E 96 C6 6B 1C 0A	DH3
0F 00 00 00 00 01 53 01 7E 96 C6 6B 1C 0A	DH5
04 00 00 00 01 01 36 00 7E 96 C6 6B 1C 0A	2DH1
0A 00 00 00 01 01 6F 01 7E 96 C6 6B 1C 0A	2DH3
0E 00 00 00 01 01 A7 02 7E 96 C6 6B 1C 0A	2DH5
08 00 00 00 01 01 53 00 7E 96 C6 6B 1C 0A	3DH1
0B 00 00 00 01 01 28 02 7E 96 C6 6B 1C 0A	3DH3
0F 00 00 00 01 01 FD 03 7E 96 C6 6B 1C 0A	3DH5

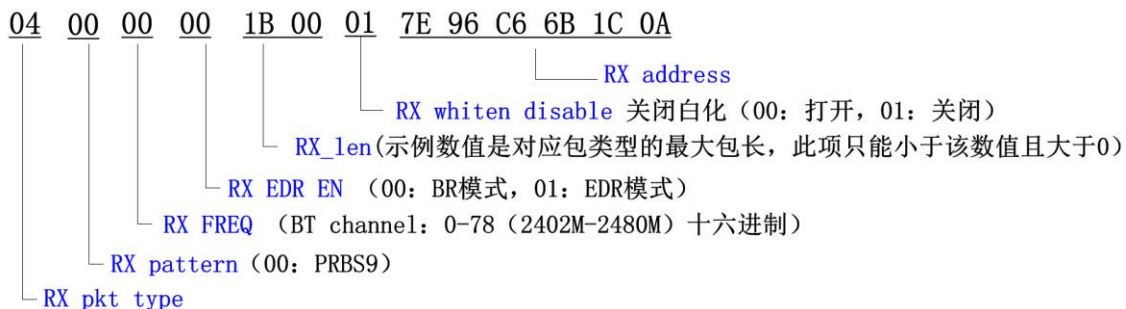


2.2.2 BT RX测试指令

1. wifi_test wlan0 bt_rxdh

eg: wifi_test wlan0 bt_rxdh 04 01 00 00 1B 00 01 7E 96 C6 6B 1C 0A

note: 以 RX DH1 包为例:



note: Package type 与 max len、Pattern、Tx address 对应关系参考图 2-2-1、2-2-2、2-2-3

bt_rx 非信令测试模式 hci cmd 示例

04 00 00 00 1B 00 01 7E 96 C6 6B 1C 0A	DH1
0B 00 00 00 B7 00 01 7E 96 C6 6B 1C 0A	DH3
0F 00 00 00 53 01 01 7E 96 C6 6B 1C 0A	DH5
04 00 00 01 36 00 01 7E 96 C6 6B 1C 0A	2DH1
0A 00 00 01 6F 01 01 7E 96 C6 6B 1C 0A	2DH3
0E 00 00 01 A7 02 01 7E 96 C6 6B 1C 0A	2DH5
08 00 00 01 53 00 01 7E 96 C6 6B 1C 0A	3DH1
0B 00 00 01 28 02 01 7E 96 C6 6B 1C 0A	3DH3
0F 00 00 01 FD 03 01 7E 96 C6 6B 1C 0A	3DH5

2. wifi_test wlan0 get_bt_rx_result no parameter

返回参数: 从 SET_RX 到 SET_RXSTOP 这段时间内接收到总的数据包个数

3. wifi_test wlan0 bt_stop val val:00/01 // 00 代表 stop tx test; 01 代表 stop rx test

对于相同的测试模式下不同参数的切换, stop 命令可发可不发。比如 tx test 模式, 在需要更新参数时, 可以不发送 stop 命令, 只需要发送 TX 相应的命令。rx test 模式同理。对于不同的测试模式之间的切换, 比如 tx test 切到 rx test, 必需先发一次对应的 stop 命令, 然后再开启新的测试。

**2-2-1 Package type 与 max length 的对应关系如下表所示:**

edr en	mode	Package type	idx	Length (max)	idx
00	BR	DH1	04	27	1B 00
		DH3	0B	138	B7 00
		DH5	0F	339	53 01
01	EDR	2DH1	04	54	36 00
		2DH3	0A	367	6F 01
		2DH5	0E	679	A7 02
		3DH1	08	83	53 00
		3DH3	0B	552	28 02
		3DH5	0F	1021	FD 03

2-2-2: Pattern:

pattern	idx
PRBS9	00
11110000	01
10101010	02
PRBS15	03
11111111	04
00000000	05
00001111	06
01010101	07

2-2-3: Rx address (note:指令里面的地址倒着写)

BD Address

BD Address 0A1C hex 6B hex C6967E hex

NAP UAP LAP

Auto Sync ☐



三. RF_TEST编译说明

1. `sudo cp *.bin /lib/firmware/aic8800/`
2. `make` 编译驱动生成 `rwnx_fdrv.ko`
3. 插入 usb 板子, 按下 `pwrkey`
4. 输入 `lsusb`, 在 ubuntu 上能看到 ID 为 `a69c:8800` 的设备
5. `sudo insmod fullmac/aic8800_fdrv.ko testmode=1` (如果要从测试模式切换回正常模式, 请 `rmmod wifi驱动` 后重新上电执行 `sudo insmod fullmac/aic8800_fdrv.ko testmode=0`)
6. 运行 `aicrf_test`

例子 1: 可以连上 cable 测试

`set_tx 1 1 2 7 1500` // chan:1 bw:20m mode:2 rate:mcs7 length:1500byte

```
aic@ubuntu:~/work/0706_aic_rfctest_v8.2_combo/driver/rwnx_drv$ aicrf_test wlan0 set_tx 1 1 2 7 1500
set_tx:
done
aic@ubuntu:~/work/0706_aic_rfctest_v8.2_combo/driver/rwnx_drv$ aicrf_test wlan0 set_txstop
set_txstop:
done
aic@ubuntu:~/work/0706_aic_rfctest_v8.2_combo/driver/rwnx_drv$
```

例子 2: 可以连上 cable 测试

`set_rx 14 1` // chan:14 bw:40m 开始接收

`set_rxstop` //停止接收

`get_rx_result:` // 1 秒内收到722个包,537 个正确

```
aic@ubuntu:~/work/0706_aic_rfctest_v8.2_combo/driver/rwnx_drv$ aicrf_test wlan0 set_rxstop
set_rxstop:
done
aic@ubuntu:~/work/0706_aic_rfctest_v8.2_combo/driver/rwnx_drv$ aicrf_test wlan0 get_rx_result
get_rx_result:
done: getrx fcsok=537, total=722
aic@ubuntu:~/work/0706_aic_rfctest_v8.2_combo/driver/rwnx_drv$
```

例子 3:

设置频偏校准:

`set_xtal_cap 1` 后晶体的寄存器值为 `0x19`, 设置为 `-1` 后晶体的值为 `0x18`, 经过校准后, 最后一次显示的值就是校准完后需要配置的值。

```
aic@ubuntu:~/work/0706_aic_rfctest_v8.2_combo/driver/rwnx_drv$ aicrf_test wlan0 set_xtal_cap 1
set_xtal_cap:
done:xtal cap: 0x19
aic@ubuntu:~/work/0706_aic_rfctest_v8.2_combo/driver/rwnx_drv$ aicrf_test wlan0 set_xtal_cap -1
set_xtal_cap:
done:xtal cap: 0x18
aic@ubuntu:~/work/0706_aic_rfctest_v8.2_combo/driver/rwnx_drv$
```

将校准后的值设置到硬件 efuse 里去:

```
aic@ubuntu:~/work/0706_aic_rfctest_v8.2_combo/driver/rwnx_drv$ aicrf_test wlan0 set_freq_cal 18
set_freq_cal:
done: freq_cal: 0x 18
aic@ubuntu:~/work/0706_aic_rfctest_v8.2_combo/driver/rwnx_drv$
```

例子 4: mac 地址的 efuse 写, 写完后读取一下:

```
aic@ubuntu:~/work/0706_aic_rfctest_v8.2_combo/driver/rwnx_drv$ aicrf_test wlan0 get_mac_addr
get_mac_addr:
done: get macaddr = 0 : 0 : 0 : 0 : 0 : 0
aic@ubuntu:~/work/0706_aic_rfctest_v8.2_combo/driver/rwnx_drv$
```

注 1:

以上是以 usb 平台为例，sdio 平台也类似，需要将 driver/rwnx_drv/fullmac/Makefile 的 CONFIG_USB_SUPPORT=n, CONFIG_SDIO_SUPPORT=y。用户空间的 aicrf_test 在客户平台上运行即可。

注 2:

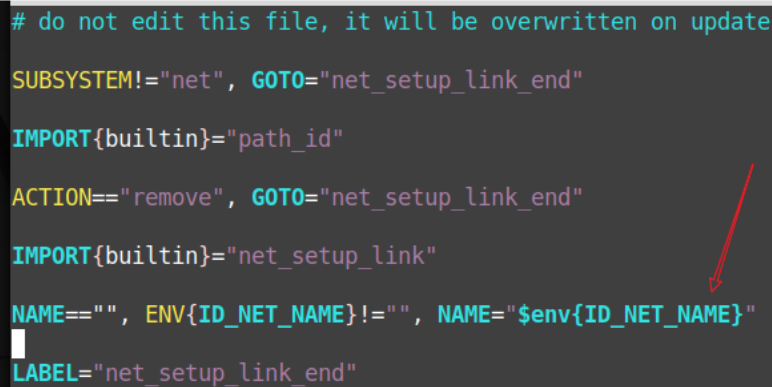
Ubuntu 平台建议做一下网络重命名规则，这样子 lsusb 后 aic8800 的芯片会显示成 wlan0，否则会用 mac 地址进行了重命名。

```
1 | cp /lib/udev/rules.d/80-net-setup-link.rules /etc/udev/rules.d/
```

然后执行如下命令，修改刚才复制过来的80-net-setup-link.rules文件：

```
1 | sudo vim /etc/udev/rules.d/80-net-setup-link.rules
```

如下图所示，将箭头所指的ID_NET_NAME改成ID_NET_SLOT即可。



```
# do not edit this file, it will be overwritten on update
SUBSYSTEM!="net", GOTO="net_setup_link_end"
IMPORT{builtin}="path_id"
ACTION=="remove", GOTO="net_setup_link_end"
IMPORT{builtin}="net_setup_link"
NAME="", ENV{ID_NET_NAME}!="", NAME="$env{ID_NET_NAME}"
LABEL="net_setup_link_end"
```


四. BT测试接口编译说明

xxx/driver/rwnx_drv/ fullmac/Makefile

aic@aic-PC:~/workspace/aic_rftest_release_2021_0926/linux/driver/rwnx_drv\$ vim fullmac/Makefile

CONFIG_BT_SUPPORT ?=y

BT 走 USB 接口时: CONFIG_RFTTEST_USB_BT=y

CONFIG_RFTTEST_UART_BT=n

BT 走 UART 接口时: CONFIG_RFTTEST_USB_BT=n

CONFIG_RFTTEST_UART_BT=y

Enable BFMER support (need FW support)

CONFIG_RWNX_BFMER ?= n

CONFIG_SDIO_SUPPORT =n

CONFIG_USB_SUPPORT =y

CONFIG_RX_REORDER ?=y

CONFIG_ARP_OFFLOAD =y

CONFIG_USE_5G =n

CONFIG_RADAR_OR_IR_DETECT =n

CONFIG_DOWNLOAD_FW =y

CONFIG_LOAD_USERCONFIG ?=y

CONFIG_BT_SUPPORT ?=y

CONFIG_RFTTEST=y

CONFIG_RFTTEST_USB_BT=y

CONFIG_RFTTEST_UART_BT=n

CONFIG_USB_BT =n

xxx/driver/rwnx_drv/ rwnx_platform.h

aic@aic-PC:~/workspace/aic_rftest_release_2021_0926/linux/driver/rwnx_drv\$ vim rwnx_platform.h

:218

BT_ONLY 走 USB 接口时: AICBT_BTMODE_BT_ONLY_TEST

AICBT_BTPORT_MB

BT_ONLY 走 UART 接口时: AICBT_BTMODE_BT_ONLY_TEST

AICBT_BTPORT_UART

BT_WIFI_COMBO 走 USB 接口时: AICBT_BTMODE_BT_WIFI_COMBO_TEST

AICBT_BTPORT_MB

BT_WIFI_COMBO 走 UART 接口时: AICBT_BTMODE_BT_WIFI_COMBO_TEST

AICBT_BTPORT_UART

#define AICBSP_HWINFO_DEFAULT

(-1)

#define AICBSP_CPMODE_DEFAULT

AICBSP_CPMODE_TEST

#define AICBT_BTMODE_DEFAULT

AICBT_BTMODE_BT_ONLY_TEST

#define AICBT_BTPORT_DEFAULT

AICBT_BTPORT_MB

#define AICBT_UART_BAUD_DEFAULT

AICBT_UART_BAUD_1_5M

#define AICBT_UART_FC_DEFAULT

AICBT_UART_FLOWCTRL_DISABLE

USB测试模式:

MB,

UART测试模式:

UART

修改完后需要在 linux/driver/rwnx_drv 重新编译驱动。编译完成后需要卸载驱动，板卡重新上电，再次加载驱动。

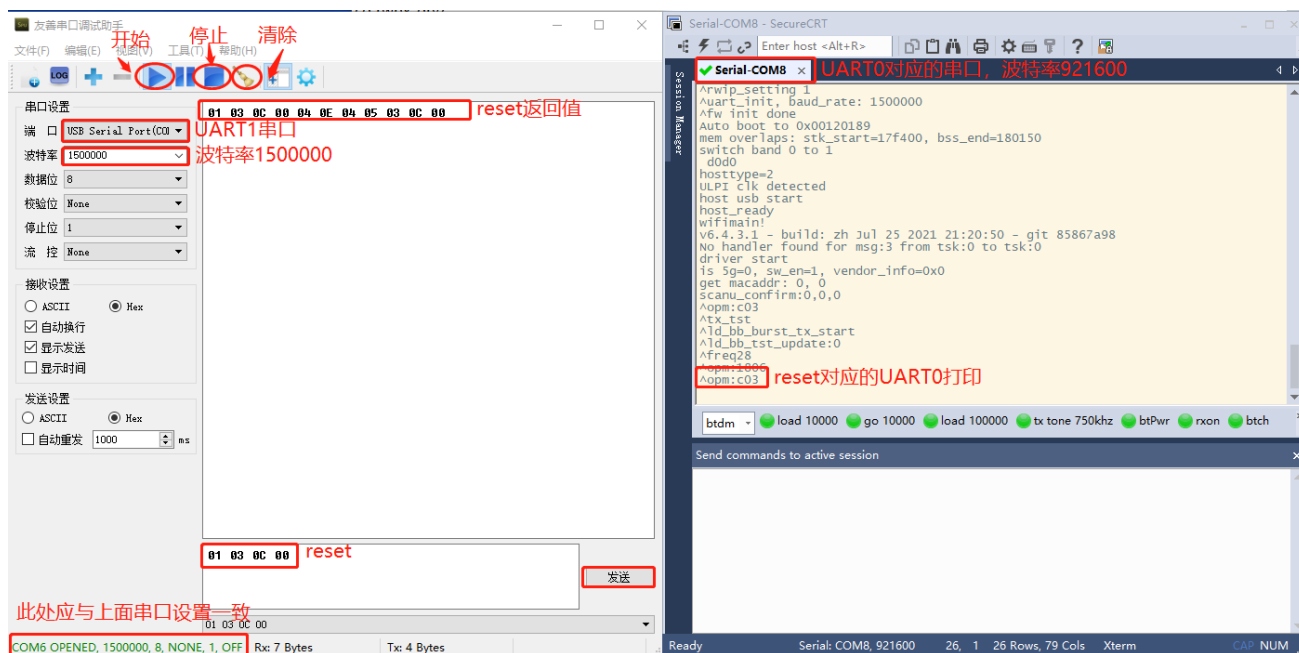
卸载驱动: sudo rmmod aic8800_fdrv

加载驱动: sudo insmod fullmac/aic8800_fdrv.ko testmode=1

五. BT走UART接口测试方法

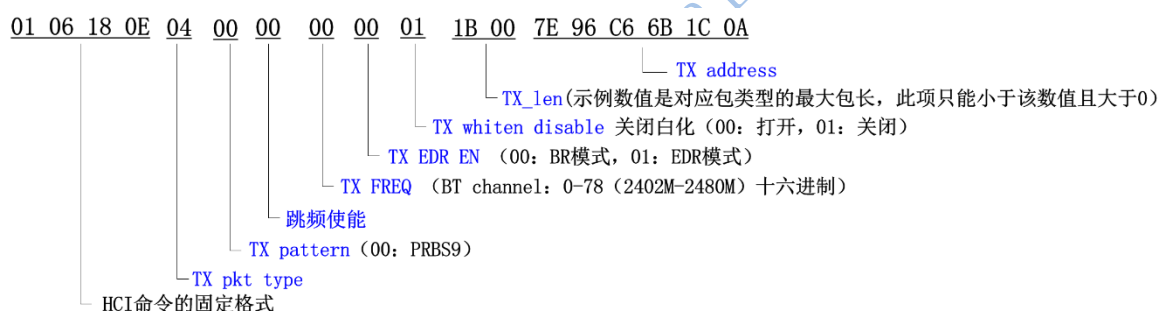
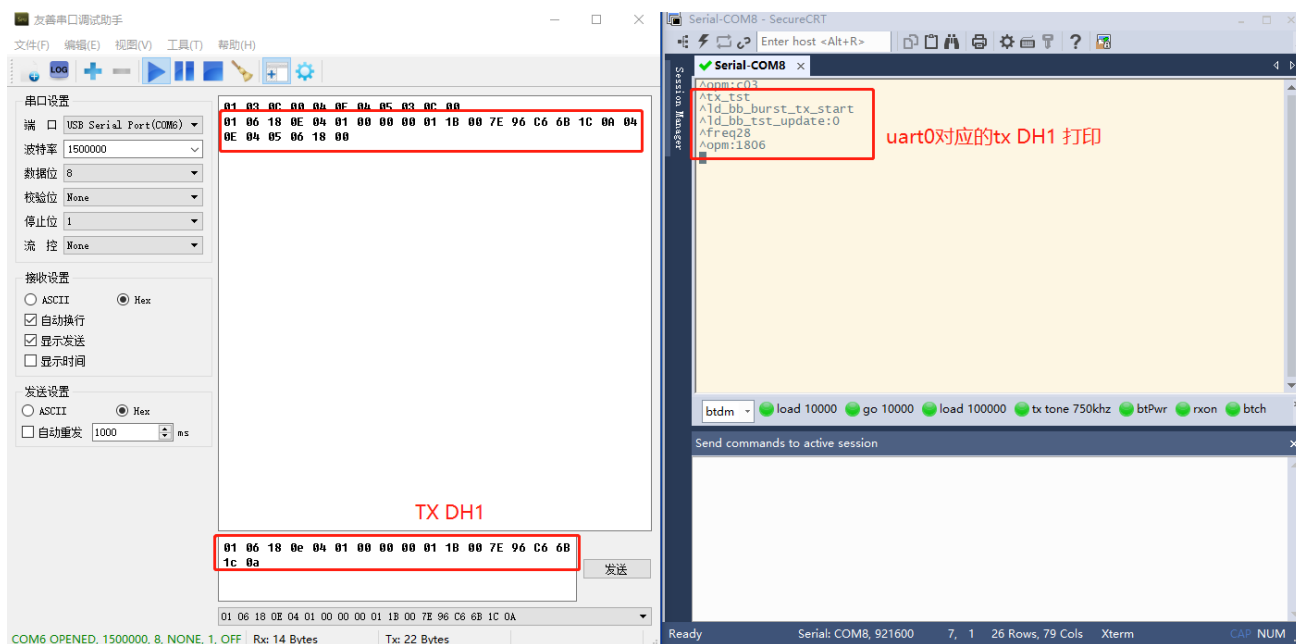
*** 输入参数均为 16 进制 ***

在 linux/driver/rwnx_drv/fullmac/Makefile 里将 BT 测试修改为 UART 测试模式，打开友善串口调试助手，设置串口，波特率为 1500000



5.1. BT TX（非信令）

以 TX DH1 为例：



note: Package type 与 max len、Pattern、Tx address 对应关系参考图 2-2-1、2-2-2、2-2-3

bt_tx 非信令测试模式 hci cmd 示例

01 06 18 0e 04 00 00 00 01 1B 00 7E 96 C6 6B 1C 0A	DH1
01 06 18 0e 0B 00 00 00 01 B7 00 7E 96 C6 6B 1C 0A	DH3
01 06 18 0e 0F 00 00 00 01 53 01 7E 96 C6 6B 1C 0A	DH5
01 06 18 0e 04 00 00 00 01 01 36 00 7E 96 C6 6B 1C 0A	2DH1
01 06 18 0e 0A 00 00 00 01 01 6F 01 7E 96 C6 6B 1C 0A	2DH3
01 06 18 0e 0E 00 00 00 01 01 A7 02 7E 96 C6 6B 1C 0A	2DH5
01 06 18 0e 08 00 00 00 01 01 53 00 7E 96 C6 6B 1C 0A	3DH1
01 06 18 0e 0B 00 00 00 01 01 28 02 7E 96 C6 6B 1C 0A	3DH3
01 06 18 0e 0F 00 00 00 01 01 FD 03 7E 96 C6 6B 1C 0A	3DH5

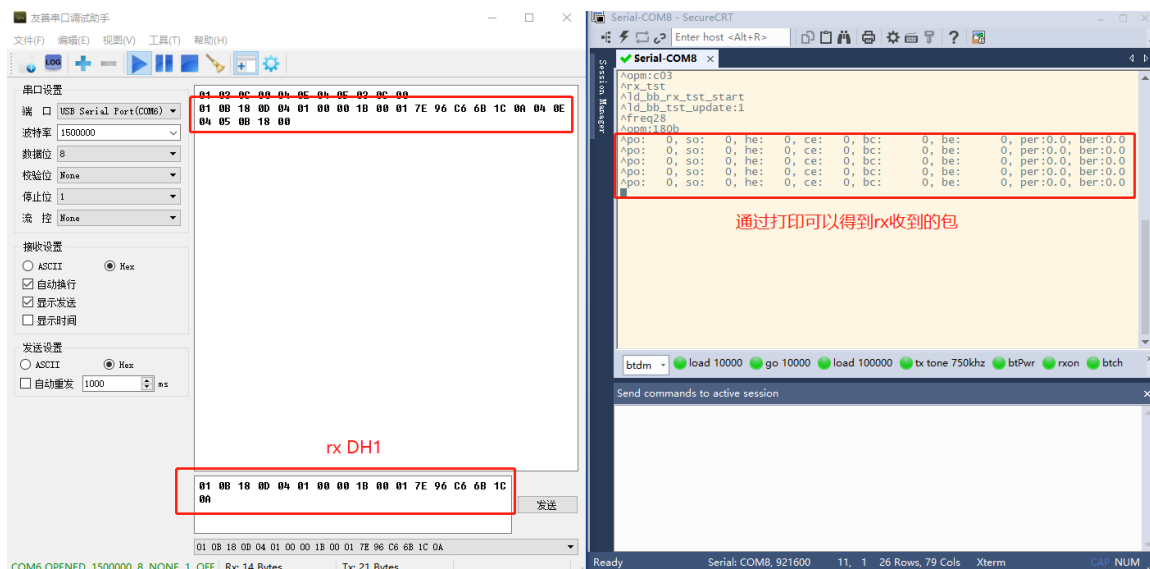
TX stop: 01 0C 18 01 00

前面四个 byte 是 hci 命令的固定格式，最后一个字节，00 代表 stop tx test



5.2. BT RX（非信令）

以 RX DH1 为例：



01 0b 18 0d 04 00 00 00 1b 00 01 7e 96 c6 6b 1c 0a

RX address

RX whitening disable 关闭白化 (00: 打开, 01: 关闭)

RX len (示例数值是对应包类型的最大包长, 此项只能小于该数值且大于0)

RX EDR EN (00: BR模式, 01: EDR模式)

RX FREQ (BT channel: 0-78 (2402M-2480M) 十六进制)

RX pattern (00: PRBS9)

RX pkt type

HCI命令的固定格式

note: Package type 与 max len、Pattern、Tx address 对应关系参考图 2-2-1、2-2-2、2-2-3

bt_rx 非信令测试模式 hci cmd 示例

01 0b 18 0d 04 00 00 00 1b 00 01 7e 96 c6 6b 1c 0a	DH1
01 0b 18 0d 0b 00 00 00 b7 00 01 7e 96 c6 6b 1c 0a	DH3
01 0b 18 0d 0f 00 00 00 53 01 01 7e 96 c6 6b 1c 0a	DH5
01 0b 18 0d 04 00 00 01 36 00 01 7e 96 c6 6b 1c 0a	2DH1
01 0b 18 0d 0a 00 00 01 6f 01 01 7e 96 c6 6b 1c 0a	2DH3
01 0b 18 0d 0e 00 00 01 a7 02 01 7e 96 c6 6b 1c 0a	2DH5
01 0b 18 0d 08 00 00 01 53 00 01 7e 96 c6 6b 1c 0a	3DH1
01 0b 18 0d 0b 00 00 01 28 02 01 7e 96 c6 6b 1c 0a	3DH3
01 0b 18 0d 0f 00 00 01 fd 03 01 7e 96 c6 6b 1c 0a	3DH5

RX stop: 01 0c 18 01 01

前面四个 byte 是 hci 命令的固定格式, 最后一个字节, 01 代表 stop rx test。

对于相同的测试模式下不同参数的切换, stop 命令可发可不发。比如 tx test 模式, 在需要更新参数时, 可以不发送 stop 命令, 只需要发送 TX 相应的命令。rx test 模式同理。

对于不同的测试模式之间的切换, 比如 tx test 切到 rx test, 必需先发一次对应的 stop 命令, 然后再开启新的测试。

5.3. BLE测试（目前只支持BT走UART接口测试模式）

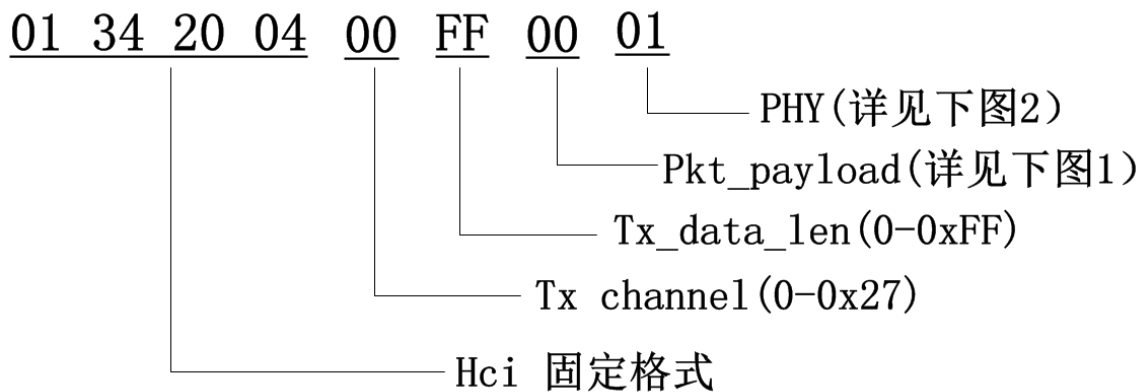
BLE 测试模式为标准指令，示例如下：

BLE TEST CMD

5.3.1 TX, LE Transmitter Test command

5.0 版本的 LE 的 tx test

(1) 带PHY的测试模式示例如下：



Packet_Payload:

Size: 1 octet

Value	Parameter Description
0x00	PRBS9 sequence '1111111100000111101...' (in transmission order) as described in [Vol 6] Part F, Section 4.1.5
0x01	Repeated '11110000' (in transmission order) sequence as described in [Vol 6] Part F, Section 4.1.5
0x02	Repeated '10101010' (in transmission order) sequence as described in [Vol 6] Part F, Section 4.1.5
0x03	PRBS15 sequence as described in [Vol 6] Part F, Section 4.1.5
0x04	Repeated '11111111' (in transmission order) sequence
0x05	Repeated '00000000' (in transmission order) sequence

图1

PHY:

Size: 1 octet

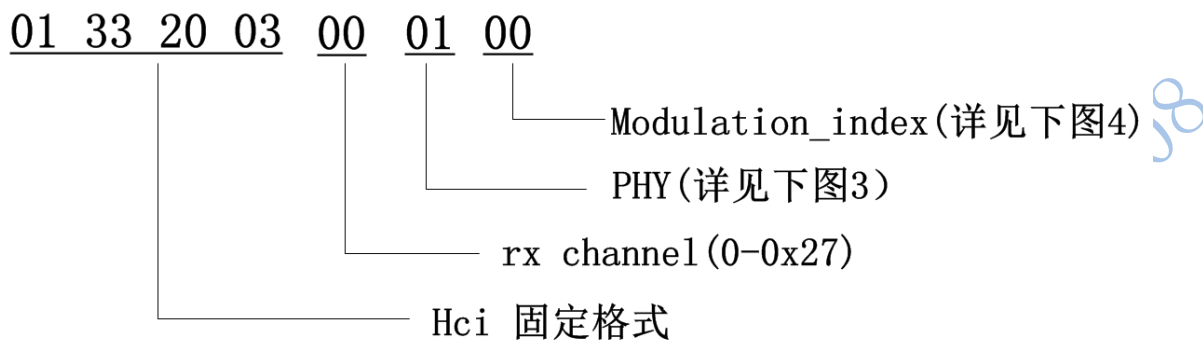
Value	Parameter Description
0x01	Transmitter set to use the LE 1M PHY
0x02	Transmitter set to use the LE 2M PHY
0x03	Transmitter set to use the LE Coded PHY with S=8 data coding
0x04	Transmitter set to use the LE Coded PHY with S=2 data coding
All other values	Reserved for future use

图 2

5.3.2 RX, LE Receiver Test command

5.0 版本的 LE 的 rx test

(1) 第二种带PHY和modulation_index, 示例如下:



PHY:

Size: 1 octet

Value	Parameter Description
0x01	Receiver set to use the LE 1M PHY
0x02	Receiver set to use the LE 2M PHY
0x03	Receiver set to use the LE Coded PHY
All other values	Reserved for future use

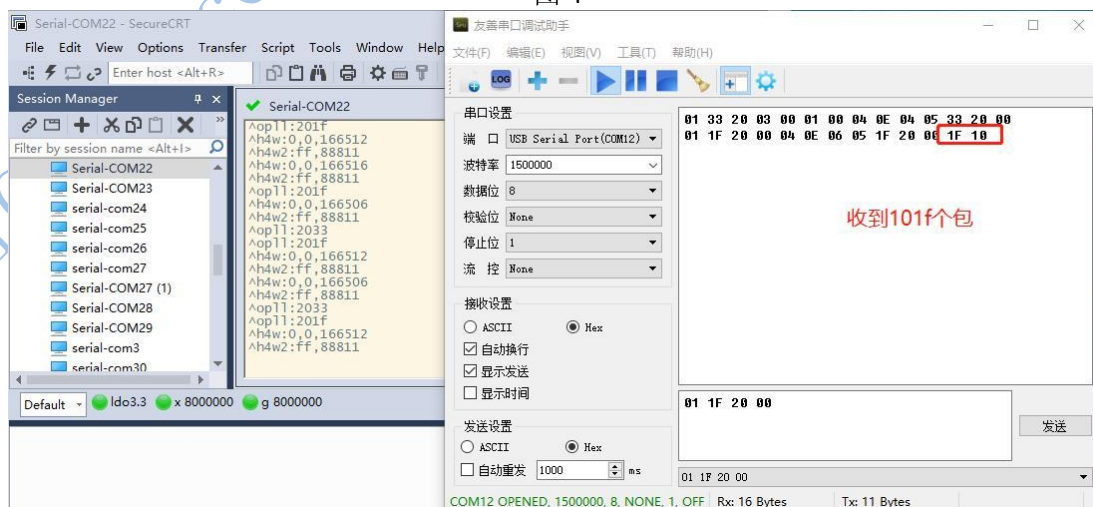
图 3

Modulation_Index:

Size: 1 octet

Value	Parameter Description
0x00	Assume transmitter will have a standard modulation index
0x01	Assume transmitter will have a stable modulation index
All other values	Reserved for future use

图 4



STOP: 01 1F 20 00 //stop the current test mode