XR829 Bluetooth RFTest CLI Tool User Guide

Android or Linux系统

Revision 1.0

November 13, 2018

Declaration

THE INFORMATION FURNISHED BY XRADIO IS BELIEVED TO BE ACCURATE AND RELIABLE. XRADIO RESERVES THE RIGHT TO MAKE CHANGES IN CIRCUIT DESIGN AND/OR SPECIFICATIONS AT ANY TIME WITHOUT NOTICE. XRADIO DOES NOT ASSUME ANY RESPONSIBILITY AND LIABILITY FOR ITS USE. NOR FOR ANY INFRINGEMENTS OF PATENTS OR OTHER RIGHTS OF THE THIRD PARTIES WHICH MAY RESULT FROM ITS USE. NO LICENSE IS GRANTED BY IMPLICATION OR OTHERWISE UNDER ANY PATENT OR PATENT RIGHTS OF XRADIO. THIS DATASHEET NEITHER STATES NOR IMPLIES WARRANTY OF ANY KIND, INCLUDING FITNESS FOR ANY PARTICULAR APPLICATION.

THIRD PARTY LICENCES MAY BE REQUIRED TO IMPLEMENT THE SOLUTION/PRODUCT. CUSTOMERS SHALL BE SOLELY RESPONSIBLE TO OBTAIN ALL APPROPRIATELY REQUIRED THIRD PARTY LICENCES. XRADIO SHALL NOT BE LIABLE FOR ANY LICENCE FEE OR ROYALTY DUE IN RESPECT OF ANY REQUIRED THIRD PARTY LICENCE. XRADIO SHALL HAVE NO WARRANTY, INDEMNITY OR OTHER OBLIGATIONS WITH RESPECT TO MATTERS COVERED UNDER ANY REQUIRED THIRD PARTY LICENCE.

Revision History

Version	Data	Summary of Changes
1.0.0	2018-11-13	Initial Version
1.0.1	2018-11-14	增加 BLE RSSI 测试项
1.0.2	2019-04-18	补充测试环境配置成功说明，确认关键节点存在等内容
1.0.3	2019-04-28	补充工具正常使用的检查点等
1.0.4	2019-07-10	修改 BT TX 测试的参数（添加 hopping_mode）
1.0.5	2019-11-27	增加了 3.2 参数说明，修改各项测试的示例
1.0.6	2020-01-19	增加了蓝牙单载波测试 single_tone 测试项
1.0.7	2020-03-11	添加了不同平台 hciattach 工具的说明

表 0-1 Revision History

目 录

图&表

1 概述

1.1 编写目的

介绍 XR829 蓝牙 RF 测试命令行工具（btetf）的使用。

1.2 使用范围

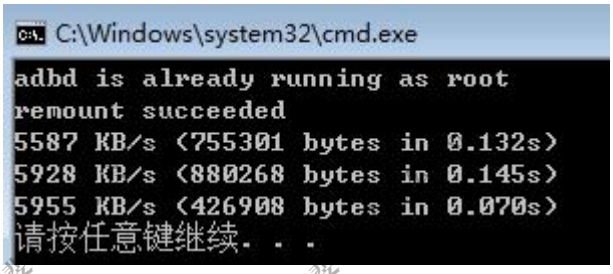
软件环境：Linux & Android

2 测试平台搭建与配置

2.1 搭建测试平台

需要的测试工具为：hciattach，hciconfig，btetf。

运行脚本“ **init_tools_linux.bat** ”或“init_tools_android.bat”进行工具安装：



说明：

（1）在使用安立 MT8850A 综测仪的时候，测试要把 TEST PAUSE 功能设置成关闭状态方可进行测试。

（2）使用 btetf 工具进行测试时依赖于 hciattach 工具加载蓝牙固件，linux 平台自带有 hciattach，因此不需要推该文件，工具包中的 hciattach 工具提供给 Android 使用。

2.2 配置测试环境（启动蓝牙）

配置测试环境有两种方法：1）运行脚本 2）手动输入相关命令

2.2.1 运行脚本进行配置

配置测试环境的命令集合已集成在脚本“ **init_test.bat** ”中。双击运行该脚本，出现“**init success!**”表示测试环境已配置成功：

```
[userial_sync] read buf: 00 00.
[userial_sync] uart sync count: 2.
[userial_sync] read buf: 00 00.
[userial_sync] uart sync count: 3.
[userial_sync] read buf: 4f 4b.
[userial_sync] Receive OK, uart sync done.
[load_btfirmware] start loading firmware...
[load_btfirmware] open firmware file success. loading...
load firmware done.
jump:
set pc 0, val 0
Now the system will jump to 00000000
Set HW FlowControl On
userial_vendor_set_hw_fctrl set hw flowcontrol on
[xradio_init] send reset cmd...
writing
01 03 0c 00
received 7
04 0e 04 05 03 0c 00
[xradio_init] update hci baudrate...
writing
01 18 fc 04 60 e3 16 00
received 7
04 0e 04 05 18 fc 00
Done setting baudrate
[xradio_init] set bdaddr...
generating random bdaddr...
writing
01 0a fc 09 02 00 06 0b b9 f9 fe 22 22
received 7
04 0e 04 05 0a fc 00
writing
01 03 0c 00
received 7
04 0e 04 05 03 0c 00
[xradio_init] bring up hci...
Done setting line discpline
Device setup complete
brom_done
hci0:   Type: Primary  Bus: UART
        BD Address: 22:22:FE:F9:B9:0B  ACL MTU: 1021:8  SCO MTU: 255:4
        UP RUNNING
        RX bytes:1168 acl:0 sco:0 events:56 errors:0
        TX bytes:752 acl:0 sco:0 commands:56 errors:0
        Features: 0xbf 0xfe 0xcd 0xfe 0xdb 0xfd 0x7b 0x87
        Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
        Link policy: RSWITCH SNIFF
        Link mode: SLAVE ACCEPT
        Name: 'XR829_BT'
        Class: 0x000000
        Service Classes: Unspecified
        Device Class: Miscellaneous,
        HCI Version: 4.1 (0x7)  Revision: 0xa68
        LMP Version: 4.1 (0x7)  Subversion: 0xa68
        Manufacturer: not assigned (1597)

init success!
```

图 2-1 运行脚本，配置测试环境

2.2.2 手动进行配置

如果选择手动配置测试环境，请依次按照以下步骤进行操作：

1) 加载 XR829 蓝牙固件（请使用实际蓝牙对应的 tty 编号替换黄色部分）

```
venus-a3:/# hciattach -n ttyS1 xradio &
```

```
venus-a3:/ # hciattach -n ttyS1 xradio &
hciattach -n ttyS1 xradio &
[1] 4182
venus-a3:/ # xradio_init
set LPM mode:disabled[userial_sync] uart sync count: 1.
[userial_sync] read buf: 00 00.
[userial_sync] uart sync count: 2.
[userial_sync] read buf: 4f 4b.
[userial_sync] Receive OK, uart sync done.
Set uart mode done
[userial_sync] uart sync count: 1.
[userial_sync] read buf: 00 00.
[userial_sync] uart sync count: 2.
[userial_sync] read buf: 4f 4b.
[userial_sync] Receive OK, uart sync done.
[load_btfirmware] start loading firmware...
[load_btfirmware] open firmware file success. loading...
load firmware done.
jump:
set pc 0, val 0
Now the system will jump to 00000000
Set HW FlowControl On
userial_vendor_set_hw_fctrl set hw flowcontrol on
[xradio_init] send reset cmd...
writing
01 03 0c 00
received 7
04 0e 04 05 03 0c 00
[xradio_init] update hci baudrate...
writing
01 18 fc 04 60 e3 16 00
received 7
04 0e 04 05 18 fc 00
Done setting baudrate
[xradio_init] set bdaddr...
writing
01 0a fc 09 02 00 06 6b 7b eb 13 22 22
received 7
04 0e 04 05 0a fc 00
writing
01 03 0c 00
received 7
04 0e 04 05 03 0c 00
[xradio_init] bring up hci...
Done setting line discpline
Device setup complete
```

图 2-2 手动配置 - 加载蓝牙固件

2) 启动设备

```
venus-a3:/# hciconfig hci0 up
```

```
venus-a3:/ # hciconfig hci0 up
hciconfig hci0 up
venus-a3:/ #
```

图 2-3 手动配置 - 启动设备

3) 检查状态

```
venus-a3:/# hciconfig -a
```

如果检查状态如下图所示，即代表初始化成功:

```
venus-a3:/ # hciconfig -a
hciconfig -a
hci0:   Type: Primary  Bus: UART
        BD Address: 22:22:13:EB:7B:6B  ACL MTU: 1021:8  SCO MTU: 255:4
        UP RUNNING
        RX bytes:1168 acl:0 sco:0 events:56 errors:0
        TX bytes:752 acl:0 sco:0 commands:56 errors:0
        Features: 0xbf 0xfe 0xcd 0xfe 0xdb 0xfd 0x7b 0x87
        Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
        Link policy: RSWITCH SNIFF
        Link mode: SLAVE ACCEPT
        Name: 'XR829_BT'
        Class: 0x000000
        Service Classes: Unspecified
        Device Class: Miscellaneous,
        HCI Version: 4.1 (0x7)  Revision: 0xa64
        LMP Version: 4.1 (0x7)  Subversion: 0xa64
        Manufacturer: not assigned (1597)
```

图 2-4 手动配置 - 检查初始化状态

2.3 工具正常使用的检查点

2.3.1 确保相关文件节点存在

配置蓝牙测试环境时，会操作一些关键文件节点；请确认系统存在以下文件节点：

```
venus-a3:/proc/bluetooth/sleep # ls -l
ls -l
total 0
--w--w---- 1 bluetooth net_bt_admin 0 2019-04-18 09:52 btwake
--w--w---- 1 bluetooth net_bt_admin 0 2019-04-18 09:52 btwrite
--w--w---- 1 bluetooth net_bt_admin 0 2019-04-18 09:52 lpm
venus-a3:/proc/bluetooth/sleep #
```

图 2-5 确认系统存在蓝牙相关的关键节点

如不存在，则表明 XR829 蓝牙休眠唤醒模块没有移植好；请确认是否已完成 《XR829 Bluetooth Porting Guide(Android8.1).pdf》 中“2.1 添加休眠唤醒与 FDI 模块” 相关步骤。

2.3.2 确保 bt 固件路径

Android 端的 hciattach 工具会在以下路径寻找 bt 固件：

/system/etc/firmware/fw_xr829_bt.bin 或者 /system/vendor/etc/firmware/fw_xr829_bt.bin

Linux 端默认的 hciattach 工具会在以下路径寻找 bt 固件：

/lib/firmware/fw_xr829_bt.bin

请确认 bt 固件的路径是否为上述路径之一。

2.3.3 确保 HCI UART driver 被编译

btetf 命令行工具依赖于内核自带的蓝牙驱动模块，请确认以下内核编译选项选上：

make ARCH=arm menuconfig --> Networking support --> Bluetooth subsystem support --> Bluetooth device drivers：

```
< > HCI USB driver
< > HCI SDIO driver
<*> HCI UART driver
[*]   UART (H4) protocol support
[ ]   BCSP protocol support (NEW)
[ ]   Atheros AR300x serial support (NEW)
[ ]   HCILL protocol support (NEW)
[ ]   Three-wire UART (H5) protocol support (NEW)
< > HCI BCM203x USB driver
< > HCI BPA10x USB driver
< > HCI BlueFRITZ! USB driver
< > HCI VHCI (Virtual HCI device) driver
< > Marvell Bluetooth driver support
```

图 2-6 确认相关内核编译配置

3 软件介绍

3.1 帮助文档

btetf 工具主要功能是进行 BT TX/RX、BLE TX/RX 测试。

可以使用“btetf -h”命令获取工具的使用帮助信息：

```
venus-a3:/# btetf -h
```

```
venus-a3:/system/bin # btetf -h
btetf -h
BT-ETF Tool Version 1.0.0
Usage:
        btetf [options] <command> [command parameters]
Options:
        --help  Display help
        --debug Dump Debug Info
        -i dev  HCI device
Commands:
        hci_reset       Hci Reset
        get_fwv         Hci Read Local Version Info
        ble_tx          Ble Tx Mode Start
        ble_rx          Ble Rx Mode Start
        ble_close       Ble Test Mode End
        bt_test_mode    Bt Test Mode Open
        bt_tx           Bt Tx Mode Start
        bt_close_tx     Bt Tx Mode End
        bt_rx           Bt Rx Mode Start
        bt_close_rx     Bt Rx Mode End
        auto_test       Auto Test Mode

For more information on the usage of each command use:
        btetf <command> --help
```

图 3-1 “btetf -h”命令获取工具使用帮助

3.2 参数说明

在某些测试项中，可能需要对链路类型和包类型进行设置。数据包类型与使用它们的逻辑传输链路有关，定义了四种不同的链路类型：ACL/SCO(Basic Rate)、eSCO(Basic Rate)、ACL(EDR)和 eSCO(EDR)。

不同传输类型有其相应的数据包类型，对包类型的具体描述见《Bluetooth Core Specification Core5.1》Vol 2, Part B, 6.5 Package type。

4 工具使用说明

4.1 BT 测试

4.1.1 BT RX 测试

可以使用“ btetf bt_rx -h ”命令获取 BT RX 测试的使用帮助信息：

```
venus-a3:/ # btetf bt_rx -h
btetf bt_rx -h
bt_rx: unrecognized option: h
Usage:
        bt_rx [option] [parameters]
Options:
       [--bdaddr=N]
       [--channel_num=N]  Range:0~79 default=1
       [--link_type=N] Range:0~3 default=0
                     0 ACL/SCO (Basic Rate)
                     1 eSCO (Basic Rate)
                     2 ACL (EDR)
                     3 eSCO (EDR)
       [--packet_type=N] Range:0~15 default=3
              ACL/SCO (Basic Rate):
                     0 NULL
                     1 POLL
                     2 FHS
                     3 DM1
                     4 DH1
                     5 HV1
                     6 HV2
                     7 HV3
                     8 DV
                     9 AUX1
                     10 DM3
                     11 DH3
                     14 DM5
                     15 DH5
              eSCO (Basic Rate):
                     0 NULL
                     1 POLL
                     7 EV3
                     12 EV4
                     13 EV5
              ACL (EDR):
                     0 NULL
                     1 POLL
                     2 FHS
                     3 DM1
                     4 2-DH1
                     5 HV1
                     6 HV2
                     7 HV3
                     8 3-DH1
                     9 AUX1
                     10 2-DH3
                     11 3-DH3
                     14 2-DH5
                     15 3-DH5
              eSCO (EDR):
                     0 NULL
                     1 POLL
                     6 2-EV3
                     7 3-EV3
                     12 2-EV5
                     13 3-EV5
Example:
tetf bt_rx --bdaddr 11:22:33:44:22:22 --channel_num 8 --link_type 1 --packet_type 3
```

图 4-1 “btetf bt_rx -h” 命令获取 BT RX 测试使用帮助

BT RX 测试需要指定以下参数：

默认参数列表	示例值	备注
bdaddr	11:22:33:44:22:22	Mac Address
channel_num	1	Channel
link_type	0	Link Type
packet_type	4	Packet Type

例子: BT RX 测试指定接收 1 信道 BR DH1 的包（注：此时应有辅助设备向测试设备发包）：

```
venus-a3:/# btetf -i hci0 -d    bt_rx --bdaddr 11:22:33:44:22:22 --channel_num 1 --link_type 0 --packet_type 4
```

```
venus-a3:/ # btetf -i hci0 -d  bt_rx --bdaddr 11:22:33:44:22:22 --channel_num 8 --link_type 0 --packet
_type 3
2:33:44:22:22 --channel_num 8 --link_type 0 --packet_type 3                    <
bt_rx:channel_num(8) link_type/packet_type(0-3) lpt:(0x 3)
< HCI Command: opcode:(0xfc50) ogf-ocf:(0x3f-0x0050) plen:(13)
   10 11 22 33 44 22 22 00 00 00 08 00 03
> HCI Event: 0x0e plen 17
   05 50 FC 00 10 11 22 33 44 22 22 00 00 00 08 00 03
venus-a3:/ #
```

图 4-2 BT RX 测试启动示例

例子: BT RX 测试指定接收 1 信道 EDR 3-DH5 的包（注：此时应有辅助设备向测试设备发包）：

```
venus-a3:/# btetf -i hci0 -d    bt_rx --bdaddr 11:22:33:44:22:22 --channel_num 1 --link_type 2 --packet_type 15
```

4.1.2 BT RX 测试结束

以下命令用于关闭 BT RX 测试：

```
venus-a3:/# btetf -i hci0 -d    bt_close_rx
```

关闭 BT RX 测试后会显示测试结果。如下图，红框显示本次 RX 测试接收到 90 个包：

```
venus-a3:/ # btetf -i hci0 -d  bt_close_rx
btetf -i hci0 -d  bt_close_rx
< HCI Command: opcode:(0xfc50) ogf-ocf:(0x3f-0x0050) plen:(1)
   F0
> HCI Event: 0x0e plen 21
   05 50 FC 00 F0 5A 00 00 00 00 00 00 00 00 00 00 00 00
   00
event: 04 0E 15 05 50 FC 00 F0 5A 00 00 00 00 00 00 00 00 00 00 00
event: 00 00 00 00
status:0x 0, mode_status:0xf0
receive_packet:90, hec_err_packet:0
crc_err_packet:0, type_err_packet:0
venus-a3:/ #
```

图 4-3 BT RX 测试结束

4.1.3 BT TX 测试

可以使用“ btetf bt_tx -h ”命令获取 BT TX 测试的使用帮助信息：

```
root@TinaLinux:/# btetf  bt_tx  -h
btetf  bt_tx  -h
bt_tx: unrecognized option: h
Usage:
        bt_tx [option] [parameters]
Options:
       [--bdaddr=N]
       [--pattern=N] Range:0~7 Data pattern
                     0 Transmitter test - 0 pattern
                     1 Transmitter test - 1 pattern
                     2 Transmitter test - 1010 pattern
                     3 Transmitter test - 0101 pattern
                     4 Transmitter test 1111 0000 pattern
                     5 Transmitter test-0000 1111 pattern
                     6 Pseudorandom 9 bit sequence
                     7 Pseudorandom 15 bit sequence
                     8-254 reserved
       [--packet_len=N]   Range:0~65535 default=10
       [--channel_num=N]  Range:0~79 default=1
       [--power_level=N]  Range:0~6 default=1
       [--link_type=N] Range:0~3 default=0
                     0 ACL/SCO (Basic Rate)
                     1 eSCO (Basic Rate)
                     2 ACL (EDR)
                     3 eSCO (EDR)
```

```
       [--packet_type=N] Range:0~15 default=3
              ACL/SCO (Basic Rate):
                     0 NULL
                     1 POLL
                     2 FHS  (0-17)
                     3 DM1  (0-17)
                     4 DH1  (0-27)
                     5 HV1  (10)
                     6 HV2  (20)
                     7 HV3  (30)
                     8 DV   (10)
                     9 AUX1 (0-29)
                     10 DM3 (0-121)
                     11 DH3 (0-183)
                     14 DM5 (0-224)
                     15 DH5 (0-339)
              eSCO (Basic Rate):
                     0 NULL
                     1 POLL
                     7 EV3  (30)
                     12 EV4 (120)
                     13 EV5 (180)
              ACL (EDR):
                     0 NULL
                     1 POLL
                     2 FHS      (0-17)
                     3 DM1      (0-17)
                     4 2-DH1    (0-54)
                     8 3-DH1    (0-83)
                     9 AUX1     (0-29)
                     10 2-DH3   (0-367)
                     11 3-DH3   (0-552)
                     14 2-DH5   (0-679)
                     15 3-DH5   (0-1021)
              eSCO (EDR):
                     0 NULL
                     1 POLL
                     6 2-EV3    (60)
                     7 3-EV3    (90)
                     12 2-EV5   (360)
                     13 3-EV5   (540)
Example:
tetf bt_tx --bdaddr 01:02:03:04:05:06 --packet_len 2 --channel_num 8 --power_level 6 --link_type 1 --p
acket type 3
```

图 4-4 “btetf bt_tx -h” 命令获取 BT TX 测试使用帮助

BT TX 测试需要指定以下参数（其中信道、链路类型和包类型要与 RX 测试时的配置一致）：

默认参数列表	示例值	备注
bdaddr	11:22:33:44:22:22	Mac Address
pattern	6	Data Type
packet_len	37	Packet Length
channel_num	1	Channel
power_level	6	Power
link_type	0	Link Type
packet_type	4	Packet Type
hopping_mode	0	关闭跳频模式

例子: BT TX 测试，指定在 1 信道发送 DH1 包，包长度为 37byte，数据为 9bit 随机填充，不打开跳频模式：

```
venus-a3:/# btetf -i hci0 -d    bt_tx --bdaddr 11:22:33:44:22:22 --pattern 6 --packet_len 37 --channel_num 1
--power_level 6 --link_type 0 --packet_type 4 --hopping_mode 0
```

```
venus-a3:/ # btetf -i hci0 -d  bt_tx --bdaddr 11:22:33:44:22:22 --packet_len 10 --channel_num 8 --powe
r_level 6 --link_type 0 --packet_type 3
n 10 --channel_num 8 --power_level 6 --link_type 0 --packet_type 3            <
bt_tx: pattern(0)-packet_len(10)-channel_num(8)-power_level(6) link_type/packet_type(0-3) lpt:(0x 3)
< HCI Command: opcode:(0xfc50) ogf-ocf:(0x3f-0x0050) plen:(13)
   00 11 22 33 44 22 22 00 0A 00 08 06 03
> HCI Event: 0x0e plen 17
   05 50 FC 00 00 11 22 33 44 22 22 00 0A 00 08 06 03
```

图 4-5 BT TX 测试启动示例

例子: BT TX 测试，指定在 1 信道发送 3-DH5 包，包长度为 37byte，数据为 9bit 随机填充，不打开跳频模式：

```
venus-a3:/# btetf -i hci0 -d    bt_tx --bdaddr 11:22:33:44:22:22 --pattern 6 --packet_len 37 --channel_num 1
--power_level 6 --link_type 2 --packet_type 15 --hopping_mode 0
```

4.1.4 BT TX 测试结束

以下命令用于关闭 BT TX 测试：

```
venus-a3:/# btetf -i hci0 -d bt_close_tx
```

```
venus-a3:/ # btetf -i hci0 -d  bt_close_tx
btetf -i hci0 -d  bt_close_tx
< HCI Command: opcode:(0xfc50) ogf-ocf:(0x3f-0x0050) plen:(1)
   F0
> HCI Event: 0x0e plen 5
   05 50 FC 00 F0
```

图 4-6 BT TX 测试结束

4.2 BLE 测试

4.2.1 BLE RX 测试

可以使用 “ btetf ble_rx -h ” 命令获取 BLE RX 测试的使用帮助信息：

```
venus-a3:/# btetf ble_rx -h
```

```
venus-a3:/ # btetf ble_rx -h
btetf ble_rx -h
ble_rx: unrecognized option: h
Usage:
        ble_rx [option] [parameters]
Options:
       [--rx_channel=N] Range:0~27 Frequency Range:2402 MHz to 2480 MHz
       [--auto] Auto test mode, must connect measuring device first
venus-a3:/ #
```

图 4-7 “btetf ble_rx -h” 命令获取 BLE RX 测试使用帮助

BLE RX 测试需要指定以下参数：

默认参数列表	示例值	备注
channel	1	Channel

例子: 指定在信道 1 监听 BLE 数据包：

```
venus-a3:/# btetf -d -i hci0 ble_rx --rx_channel 1
```

```
venus-a3:/ # btetf -d -i hci0 ble_rx --rx_channel 5
btetf -d -i hci0 ble_rx --rx_channel 5
ble_rx:rx_channel:5
< HCI Command: opcode:(0x201d) ogf-ocf:(0x08-0x001d) plen:(1)
   05
> HCI Event: 0x0e plen 4
   05 1D 20 00
```

图 4-8 BLE RX 测试启动示例

4.2.2 BLE RX 测试结束

以下命令用于关闭 BLE RX 测试：

```
venus-a3:/# btetf -d -i hci0 ble_close
```

关闭 BLE RX 测试后会显示测试结果。如下图，红框显示本次 BLE RX 测试接收到 BLE 数据包 3744 个：

```
venus-a3:/ # btetf -d -i hci0 ble_close
btetf -d -i hci0 ble_close
< HCI Command: opcode:(0x201f) ogf-ocf:(0x08-0x001f) plen:(0)
> HCI Event: 0x0e plen 6
   05 1F 20 00 A0 0E
event: 04 0E 06 05 1F 20 00 A0 0E
status:0x 0
send_or_receive_packet:3744
venus-a3:/ #
```

图 4-9 BLE RX 测试结束，显示收到的包个数

4.2.3 BLE RX 测试结束后 Read RSSI

```
venus-a3:/# btetf -i hci0 -d    ble_rssi
```

```
root@TinaLinux:/# btetf -d -i hci0 ble_rssi
btetf -d -i hci0 ble_rssi
< HCI Command: opcode:(0xfc15) ogf-ocf:(0x3f-0x0015) plen:(1)
   00
> HCI Event: 0x0e plen 8
   05 15 FC 00 00 00 00 DF
event: 04 0E 08 05 15 FC 00 00 00 00 DF
status:0x 0
ble_rssi_result:-32
root@TinaLinux:/#
```

图 4-10 BLE RX 测试结束 Read RSSI

4.2.4 BLE TX 测试

可以使用“ btetf ble_tx -h ”命令获取 BLE TX 测试的使用帮助信息：

```
venus-a3:/# btetf ble_tx -h
```

```
venus-a3:/ # btetf ble_tx -h
btetf ble_tx -h
ble_tx: unrecognized option: h
Usage:
        ble_tx [option] [parameters]
Options:
       [--tx_channel=N] Range:0~39 Frequency Range:2402 MHz to 2480 MHz
       [--len=N] Range:0~37 Length in bytes of payload data in each packet
       [--payload=N] Range:0~7 Type of payload
                     0x00 Pseudo-Random bit sequence 9
                     0x01 Pattern of alternating bits '11110000'
                     0x02 Pattern of alternating bits '10101010'
                     0x03 Pseudo-Random bit sequence 15
                     0x04 Pattern of All '1' bits
                     0x05 Pattern of All '0' bits
                     0x06 Pattern of alternating bits '00001111'
                     0x07 Pattern of alternating bits '0101'
       [--auto] Auto test mode, must connect measuring device first
venus-a3:/ #
```

图 4-11 “btetf ble_tx -h” 命令获取 BLE TX 测试使用帮助

BLE TX 测试需要指定以下参数：

默认参数列表	示例值	备注

channel	1	Channel
len	37	Packet length
payload	0	Data Type

例子: 指定在信道 1 发送 BLE 数据包，包长度 37byte，数据为随机填充。

```
venus-a3:/# btetf -d -i hci0 ble_tx --tx_channel 1 --len 37 --payload 0
```

```
root@TinaLinux:/# btetf -d -i hci0 ble_tx --tx_channel 5 --len 10 --payload 0
btetf -d -i hci0 ble_tx --tx_channel 5 --len 10 --payload 0
ble_tx:tx_channel:5, tx_data_len:10, packet_payload:0
< HCI Command: opcode:(0x201e) ogf-ocf:(0x08-0x001e) plen:(3)
   05 0A 00
> HCI Event: 0x0e plen 4
   05 1E 20 00
```

图 4-12 BLE TX 测试启动示例

4.2.5 BLE TX 测试结束

```
venus-a3:/# btetf -d -i hci0 ble_close
```

```
root@TinaLinux:/# btetf -d -i hci0 ble_close
btetf -d -i hci0 ble_close
< HCI Command: opcode:(0x201f) ogf-ocf:(0x08-0x001f) plen:(0)
> HCI Event: 0x0e plen 6
   05 1F 20 00 00 00
event: 04 0E 06 05 1F 20 00 00 00
status:0x 0
send_or_receive_packet:0
root@TinaLinux:/#
```

图 4-13 BLE TX 测试结束

4.3 BT 单载波测试

可以使用 “ btetf single_tone --help” 命令获取 BLE RX 测试的使用帮助信息：

```
venus-a3:/# btetf single_tone --help
```

```
venus-a3:/ # btetf single_tone --help
btetf single_tone --help
Usage:
        single_carrier [option]
Options:
        [--open channel_num]
        [--power_level level]
        [--close]
Example:
tetf single_tone --opne 0 --power_level 6
```

图 4-14 “btetf single_tone --help” 命令获取蓝牙单载波测试使用帮助

蓝牙单载波测试需要指定以下参数：

默认参数列表	示例值	备注
open	1	channel
power_level	6	Power
close	NULL	close

例子: 指定在信道 1 进行单载波测试，power 使用默认值：

```
venus-a3:/# btetf -d -i hci0 single_tone --open 1
```

例子: 关闭单载波测试：

```
venus-a3:/# btetf -d -i hci0 single_tone --close
```

5 FAQ

5.1 配置测试环境失败

如果配置测试环境失败，请首先按照本文档“2.3 工具正常使用的检查点”进行检查；然后按照以下步骤进行排查。

5.1.1 不能打开串口

双击脚本“ init_test.bat ”进行测试环境配置，出现“Cant't open serial port”问题：

```
adbd is already running as root
remount succeeded
kill_hciattach
adb shell kill -9
Can't open serial port: No such file or directory
Can't initialize device: No such file or directory
```

图 5-1 运行“init_test.bat”脚本出现“Cant't open serial port”问题

请确认“ init_test.bat ”脚本中的 tty 编号是否为蓝牙对应的 tty 编号：

init_test.bat

```
1  @echo off
2  adb root
3  adb remount
4
5  ::kill_hciattach
6  echo kill_hciattach
7      adb shell "ps -A| grep hciattach| busybox awk '{print $2}'" >xrTxtemp.txt
8      for /f  %%i in (xrTxtemp.txt) do (
9      set pid=%%i
10     )
11     echo adb shell kill -9 %pid%
12 del xrTxtemp.txt
13 if "%pid%"=="" (goto start_bt_fw_download)
14 adb shell kill -9 %pid%
15
16 :start_bt_fw_download
17 start "" /b  cmd /c "adb shell hciattach -n ttyS1 xradio &"
18 ping 1.1.1.1 -n 15  > nul
19 echo brom_done
```

图 5-2 确认“init_test.bat”脚本中的 tty 编号

其中，蓝牙对应的 tty 编号可以通过查看 /dev/ 目录获得：

```
crw-------  1 root      root            10,  59 2019-04-28 10:48 sunxi-wlan
crw-------  1 root      root            10,  61 2019-04-28 10:48 sunxi_soc_info
crwxrwxrw-  1 media     mediadrm       250,   0 2019-04-28 10:48 tee0
crwxrwxrw-  1 media     mediadrm       250,  16 2019-04-28 10:48 teepriv0
crw-rw-rw-  1 root      root             5,   0 2019-04-28 10:48 tty
crw-------  1 root      root           247,   0 2019-04-28 10:48 ttyS0
crw-rw----  1 bluetooth net_bt_admin   247,   1 2019-04-28 10:50 ttyS1
crw-rw----  1 system    vpn             10, 200 2019-04-28 10:48 tun
crw-rw----  1 uhid      uhid            10, 239 2019-04-28 10:48 uhid
crw-rw----  1 system    bluetooth       10, 223 2019-04-28 10:48 uinput
crw-rw-rw-  1 root      root             1,   9 2019-04-28 10:48 urandom
drwxrwx---  3 shell     shell                60 2019-04-28 10:48 usb-ffs
crw-rw----  1 root      usb             10,  46 2019-04-28 10:48 usb_accessory
crwxrwx---  1 camera    camera          81,   1 2019-04-28 10:48 v4l-subdev0
crwxrwx---  1 camera    camera          81,   2 2019-04-28 10:48 v4l-subdev1
crwxrwx---  1 camera    camera          81,   3 2019-04-28 10:48 v4l-subdev2
crwxrwx---  1 camera    camera          81,   4 2019-04-28 10:48 v4l-subdev3
crwxrwx---  1 camera    camera          81,   5 2019-04-28 10:48 v4l-subdev4
crwxrwx---  1 camera    camera          81,   6 2019-04-28 10:48 v4l-subdev5
crwxrwx---  1 camera    camera          81,   7 2019-04-28 10:48 v4l-subdev6
crwxrwx---  1 camera    camera          81,   8 2019-04-28 10:48 v4l-subdev7
crwxrwx---  1 camera    camera          81,   9 2019-04-28 10:48 v4l-subdev8
crwxrwx---  1 camera    camera          81,   0 2019-04-28 10:48 video0
crw-rw-rw-  1 root      root            10,  53 2019-04-28 10:48 vndbinder
crw-r--r--  1 root      root            10,  52 2019-04-28 10:48 xt_qtaguid
crw-rw-rw-  1 root      root             1,   5 2019-04-28 10:48 zero
venus-a3:/dev #
```

图 5-3 查看设备上蓝牙对应的 tty 编号

5.1.2 Fail To Load FW 的排查

蓝牙启动过程中首先会进行 Brom sync 波特率同步，如果同步失败，则无法正常加载固件。log 如下（一直在同步）：

```
venus-a3:/ # hciattach -n ttyS1 xradio &
hciattach -n ttyS1 xradio &
[1] 11395
venus-a3:/ # xradio_init
set LPM mode:disabled[userial_sync] uart sync count: 1.
[userial_sync] read buf: 00 00.
[userial_sync] uart sync count: 2.
[userial_sync] read buf: 00 00.
[userial_sync] uart sync count: 3.
[userial_sync] read buf: 00 00.
[userial_sync] uart sync count: 4.
[userial_sync] read buf: 00 00.
[userial_sync] uart sync count: 5.
[userial_sync] read buf: 00 00.
```

图 5-4 Brom 同步失败

可能原因如下：

- 蓝牙固件已加载

首先确认 Android 界面蓝牙处于关闭状态（或确认 Linux 系统蓝牙功能已关闭）；然后检查是否有 hciattch

进程，若存在则表明已进行相关配置，无需重复加载固件。

```
venus-a3:/ # ps -A | grep hciattach

venus-a3:/ # kill -9 PID
```

```
venus-a3:/ # ps -A | grep hciattach
ps -A | grep hciattach
root          3398 27197     900    644 poll_schedule_timeout 0 S hciattach
venus-a3:/ #

venus-a3:/ #

venus-a3:/ # kill -9 3398
kill -9 3398
[1] + Killed                 \hciattach -n ttyS1 xradio
venus-a3:/ #
```

图 5-5 检查蓝牙是否已加载

- 芯片 core 没有 reset

通过“cat /sys/class/rfkill/rfkill0/state”命令查看 BT_RST 电平是否为 1，如图所示：

```
venus-a3:/sys/class/rfkill/rfkill0 # cat state
cat state
1
```

图 5-6 检查芯片 core 是否已 reset

- 芯片没有 wake up

通过“echo 1 > /proc/bluetooth/sleep/btwake”命令进行测试，测试前确认 lpm 处于 enable 状态

```
venus-a3:/proc/bluetooth/sleep # cat btwake
cat btwake
bt wake state:1
```

图 5-7 检查芯片是否 wake up