



XRADIO BT RFTest 工具 使用指南

版本号：1.2

发布时间：2021-11-16

版本历史

版本	日期	责任人	版本描述
1.2	2021-11-16	AWA 1677	增加 5.4.6 章节，补充 BLE TX 功率调整的说明。
1.1	2021-08-06	AWA 1677	3.1 章节补充工具文件夹的说明。
1.0	2021-05-12	AWA 1677	创建文档。

目录

版本历史.....	i
目录.....	ii
图片目录.....	iv
1 前言.....	1
1.1 文档简介.....	1
1.2 目标读者.....	1
1.3 适用范围.....	1
1.4 文档约定.....	1
1.4.1 标志说明.....	1
1.4.2 地址与数据描述方法约定.....	1
1.4.3 数值单位约定.....	2
2 工具简介.....	3
3 测试平台搭建与配置.....	4
3.1 搭建测试平台.....	4
3.2 配置测试环境（启动蓝牙）.....	5
3.2.1 运行脚本进行配置.....	5
3.2.2 手动进行配置.....	5
3.3 工具正常使用的检查点.....	8
3.3.1 确保相关文件节点存在.....	8
3.3.2 确保相关文件节点存在.....	8
3.3.3 确保 HCI UART driver 被编译.....	8
4 RfTester 测试工具使用说明.....	10
4.1 打开程序.....	10
4.2 打开蓝牙.....	10
4.3 BT Test.....	11
4.4 参数设置.....	11
4.5 BT TX.....	12
4.6 BT RX.....	12
4.7 BLE TX/RX.....	13
5 btetf 测试工具使用说明.....	15
5.1 帮助说明.....	15
5.2 参数说明.....	15

5.3 BT 测试.....	15
5.3.1 BT RX 测试.....	15
5.3.2 BT RX 测试结束.....	17
5.3.3 BT TX 测试.....	17
5.3.4 BT TX 测试结束.....	19
5.4 BLE 测试.....	19
5.4.1 BLE RX 测试.....	19
5.4.2 BLE RX 测试结束.....	20
5.4.3 BLE RX 测试结束后 Read RSSI.....	20
5.4.4 BLE TX 测试.....	20
5.4.5 BLE TX 测试结束.....	21
5.4.6 BLE TX 功率调整.....	21
5.5 BT 单载波测试.....	22
5.6 TX 测试.....	22
5.6.1 BT 单载波示例.....	22
5.6.2 BT TX 测试示例.....	22
5.6.3 BLE TX 测试示例 1.....	23
5.6.4 BLE TX 测试示例 2.....	23
5.7 RX 测试.....	23
5.7.1 BT RX 测试示例.....	23
5.7.2 BLE RX 测试示例.....	23
6 FAQ.....	24
6.1 配置测试环境失败.....	24
6.1.1 不能打开串口.....	24
6.1.2 Fail To Load FW 的排查.....	25
6.2 收发包异常.....	26

图片目录

图 3-1	脚本运行示意图	4
图 3-2	配置成功图	5
图 3-3	手动配置 - 加载蓝牙固件图 (XR829)	6
图 3-4	手动配置 - 加载蓝牙固件图 (XR819S)	7
图 3-5	手动配置 - 启动设备图	7
图 3-6	手动配置 - 检查初始化状态图	8
图 3-7	确认系统存在蓝牙相关的关键节点	8
图 3-8	确认相关内核编译配置	9
图 4-1	RFtester v2.4	10
图 4-2	打开蓝牙界面	11
图 4-3	打开蓝牙成功界面	11
图 4-4	参数设置界面	12
图 4-5	APK 执行 BT TX 测试的界面	12
图 4-6	APK BT RX 测试的界面	13
图 4-7	BLE TX 界面	13
图 4-8	BLE RX 界面	14
图 5-1	“btetf -h” 命令获取工具使用帮助	15
图 5-2	“btetf bt_rx -h” 命令获取 BT RX 测试使用帮助	16
图 5-3	BT RX 测试启动示例	17
图 5-4	BT RX 测试结束	17
图 5-5	“btetf bt_tx -h” 命令获取 BT TX 测试使用帮助	17
图 5-6	“btetf bt_tx -h” 命令获取 BT TX 测试使用帮助	18
图 5-7	BT TX 测试启动示例	19
图 5-8	BT TX 测试结束	19
图 5-9	BLE RX 帮助示例	19
图 5-10	RX 结果图	20
图 5-11	BLE RX 测试结束，显示收到的包个数	20
图 5-12	BLE RX 测试结束 Read RSSI	20
图 5-13	BLE TX 测试启动示例 (XR829)	21
图 5-14	BLE TX 测试启动示例 (XR819S)	21
图 5-15	BLE TX 测试结束	21

图 5-16 “btetf single_tone --help” 命令获取蓝牙单载波测试使用帮助.....	22
图 6-1 运行“init_test.bat”脚本出现“Cant’t open serial port”问题.....	24
图 6-2 查看设备上蓝牙对应的 tty 编号.....	25
图 6-3 Brom 同步失败.....	25
图 6-4 检查蓝牙是否已加载.....	26
图 6-5 检查芯片 core 是否已 reset.....	26
图 6-6 检查芯片是否 wake up.....	26
图 6-7 收发包数量图.....	26
图 6-8 hciconfig 结果图.....	27
图 6-9 hciconfig 结果图.....	27

1 前言

1.1 文档简介

本文档主要介绍了 XR 系列芯片在 Android 或 Linux 环境下进行 RF 测试所用工具的使用方法和常见问题的分析，下面将以 Android10 (A100 B3 + XR829)平台为例介绍 RFtester.apk 与 Tina (R328 + XR829) 平台上 ETF CLI 工具的使用方法，并列举部分测试用例，其他平台以此参考。

1.2 目标读者

从事蓝牙模块开发和测试的相关人员。





1.3 适用范围

Linux 和 Android 平台。

1.4 文档约定

1.4.1 标志说明

本文档采用各种醒目的标志来表示在操作过程中应该特别注意的地方，这些标志的含义如下：

标识	说明
 警告	该标志后的说明应给予格外关注，如果不遵守，可能会导致人员受伤或死亡。
 注意	提醒操作中应注意的事项。不当的操作可能会损坏器件，影响可靠性、降低性能等。
 说明	为准确理解文中指令、正确实施操作而提供的补充或强调信息。
 窍门	一些容易忽视的小功能、技巧。了解这些功能或技巧能帮助解决特定问题或者节省操作时间。

1.4.2 地址与数据描述方法约定

本文档在描述地址、数据时遵循如下约定：

符号	例子	说明
0x	0x0200, 0x79	地址或数据以 16 进制表示。
0b	0b010, 0b00 000 111	数据采用二进制表示(寄存器描述除外)。
X	00X, XX1	数据描述中，X 代表 0 或 1。 例如，00X 代表 000 或 001；XX1 代表 001, 011, 101 或 111。

1.4.3 数值单位约定

本文档在描述数据容量（如 NAND 容量）时，单位词头代表的是 1024 的倍数；描述频率、数据速率等时则代表的是 1000 的倍数。具体如下：

类型	符号	对应数值
数据容量（如 NAND 容量）	1 K	1024
	1 M	1 048 576
	1 G	1 073 741 824
频率，数据速率等	1 k	1000
	1 M	1 000 000
	1 G	1 000 000 000

2 工具简介

蓝牙 RF 测试工具，主要用于 BT/BLE 的 RF 测试。其主要的功能与目的如下：

- 1) RFtester APK:用户可以通过在 APK 里面点击按键下发指令至 XR819S 或者 XR829 的方式,控制 XR819S 或者 XR829 执行 TX、RX 动作。
- 2) 命令行工具:用户不必点击 Android 平台的 APK 界面,就可以通过输入命令来执行测试,这样可以在高低温测试、蓝牙音箱等不方便点击、甚至没有界面的场景下进行蓝牙 RF 测试。

其功能列举如表 2-1 所示：

表 2-1 RF 测试工具功能表

主要功能	功能描述
BLE TX	控制 XR829 或者 XR819S 进行 BLE TX。可以设置信道、比特模式、包长。支持与综测仪连接测试。
BLE RX	控制 XR829 或者 XR819S 进行 BLE RX。可以设置信道、比特模式。停止 RX 后会返回 RX 结果，即收包总数、错误数。支持与综测仪连接测试。
BT TX	控制 XR829 进行 BT TX。可以设置信道、链路类型、包类型、比特模式、功率、包长。
BT RX	控制 XR829 进行 BT RX。可以设置信道、链路类型、包类型、比特模式。停止 RX 后可以返回 RX 结果，即收包总数、HEC 错误数、CRC 错误数、包类型错误数。



注意

XR829 支持 BT/BLE 的测试，XR819S 只支持 BLE 的测试。

3 测试平台搭建与配置

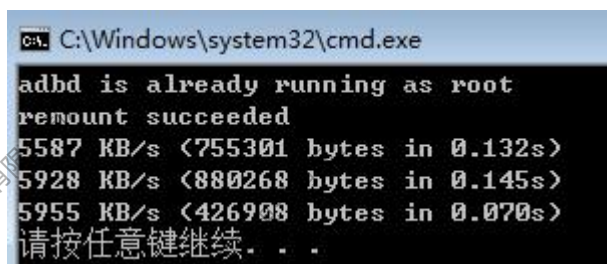
3.1 搭建测试平台

需要的测试工具为：hciattach, hciconfig, btetf。

方式一：脚本安装

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

图 3-1 脚本运行示意图



方式二：手动安装

- 1) 通过 adb 把相关的工具推到指定的目录。默认情况下,Android 平台：**/vendor/bin**, Tina 平台：**/usr/bin**, 此外还可以根据需求下载到其他自定义的目录下。
- 2) 修改权限：xxx/bin/etf 权限修改为 755

/*xxx 为本地存放 RFtester_v2.4.apk、hciattach、hciconfig、btetf 的具体目录, Android 平台*/

```
adb install xxx/RFtester_v2.4.apk
adb push xxx/hciattach /vendor/bin
adb push xxx/hciconfig /vendor/bin
adb push xxx/btetf /vendor/bin

adb shell chmod 755 /vendor/bin/hciconfig
adb shell chmod 755 /vendor/bin/hciattach
adb shell chmod 755 /vendor/bin/btetf
```

/*xxx 为本地存放 btetf、haciaatch、hciconfig 的具体目录, Tina 平台*/

```
adb push xxx/btetf /usr/bin
adb push xxx/hciattach /usr/bin
adb push xxx/hciconfig /usr/bin

adb shell chmod 755 /usr/bin/hciattach
adb shell chmod 755 /usr/bin/btetf
adb shell chmod 755 /usr/bin/hciconfig
```

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。
- (3) 工具文件夹中，hciattach_xr829 适用于 xr829，hciattach_xr819s 适用于 xr819s。
- (4) 如果运行 “init_tools_linux.bat” 或 “init_tools_android.bat” 脚本失败，请确认脚本内工具名字和工具文件夹的名字是否对得上。

3.2 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

3.2.1 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：

图 3-2 配置成功图

```

[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
[rxradio_initl] send reset cmd...
writing
01 03 0c 00
received 7
04 0e 04 05 03 0c 00
[rxradio_initl] update hci baudrate...
writing
01 18 fc 04 60 e3 16 00
received 7
04 0e 04 05 18 fc 00
Done setting baudrate
[rxradio_initl] 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
[rxradio_initl] bring up hci...
Done setting line discipline
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 HU1 HU2 HU3
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!
    
```

3.2.2 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 或 XR819S 蓝牙固件（请使用实际蓝牙对应的 tty 编号替换黄色部分）

```
hciattach -n ttyS1 xradio &
```



注意

默认使用 ttyS1，个别平台使用 ttyAS1。

手动配置 - 加载蓝牙固件图 (XR829)

```
root@TinaLinux:/# hciattach -n ttyS1 xradio &
hciattach -n ttyS1 xradio &
root@TinaLinux:/# 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.
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 87 21 2c c8 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 discipline
Device setup complete
```

XR819S 启动示意图与上类似，唯一区别如下图。

图 3-4 手动配置 - 加载蓝牙固件图 (XR819S)

```
hciattach -n ttyS1 xradio &
root@TinaLinux:/# xradio_init
set LPM mode:disabled[serial_sync] uart sync count: 1.
[serial_sync] read buf: 00 00.
[serial_sync] uart sync count: 2.
[serial_sync] read buf: 00 00.
[serial_sync] uart sync count: 3.
[serial_sync] read buf: 4f 4b.
[serial_sync] Receive OK, uart sync done.
Set uart mode done
[serial_sync] uart sync count: 1.
[serial_sync] read buf: 00 00.
[serial_sync] uart sync count: 2.
[serial_sync] read buf: 4f 4b.
[serial_sync] Receive OK, uart sync done.
[load_btfirmware] start loading firmware...
[load_btfirmware] open firmware file success.
load firmware done.
jump:
set pc 201101, val 1112000
Now the system will jump to 00201101
Set HW FlowControl On
[serial_vendor_set_hw_fctrl] set hw flowcontrol on
[rxradio_init] send reset cmd...
writing
01 03 0c 00
received 7
04 0e 04 01 03 0c 00
[rxradio_init] set bdaddr...
writing
01 0a fc 09 02 00 06 89 6a 22 c7 22 22
received 7
04 0e 04 01 0a fc 00
writing
01 03 0c 00
received 7
04 0e 04 01 03 0c 00
[rxradio_init] bring up hci...
Done setting line discipline
Device setup complete
```

2) 启动设备

```
hciconfig hci0 up
```

图 3-5 手动配置 - 启动设备图

```
root@TinaLinux:/# hciconfig hci0 up
hciconfig hci0 up
root@TinaLinux:/#
```

3) 检查状态

```
hciconfig -a
```

如果检查状态如下图所示，即代表初始化成功:

图 3-6 手动配置 - 检查初始化状态图

```
root@TinaLinux:/# hciconfig -a
hciconfig -a
hci0: Type: Primary Bus: UART
      BD Address: 22:22:C8:2C:21:87 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: 0xc19
      LMP Version: 4.1 (0x7) Subversion: 0xc19
      Manufacturer: not assigned (1597)
```

3.3 工具正常使用的检查点

3.3.1 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/proc/bluetooth/sleep 目录下存在以下文件节点：

图 3-7 确认系统存在蓝牙相关的关键节点

```
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
```

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。XR819S 也同理。

3.3.2 确保相关文件节点存在

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

```
vendor/etc/firmware/*.bin
```

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

```
/lib/firmware/*.bin
```

*.bin 为芯片对应的固件名字，例如 fw_xr829_bt.bin 或者 fw_xr819s.bin。请确认 bt 固件的路径是否为上述路径之一。

3.3.3 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:

图 3-8 确认相关内核编译配置

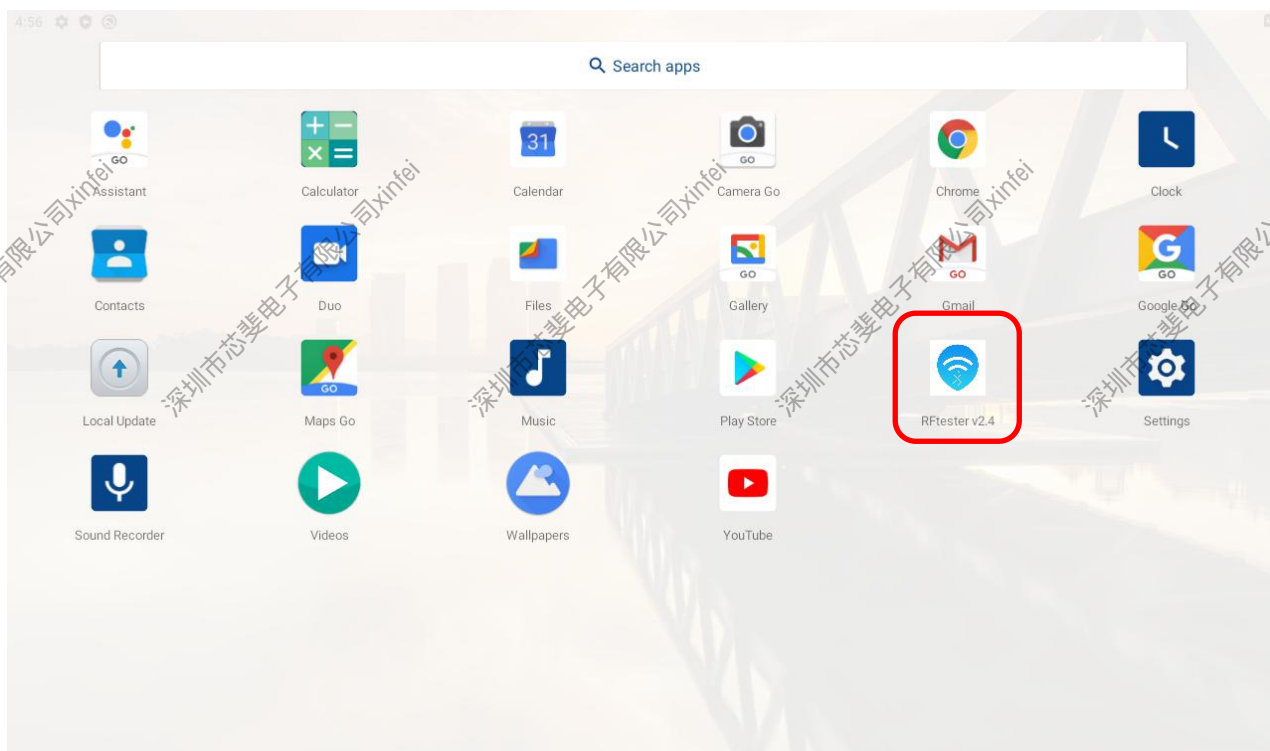
```
< > HCI USB driver
< > HCI SDIO driver
< * > HCI UART driver
[ * ] UART (H4) protocol support
[ ] BCSP protocol support (NEW)
[ ] Atheros AR300x serial support (NEW)
[ ] HCILL protocol support (NEW)
[ ] Three-wire UART (H5) protocol support (NEW)
< > HCI BCM203x USB driver
< > HCI BPA10x USB driver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

4 RFtester 测试工具使用说明

4.1 打开程序

找到 RFtester v2.4，打开，如图 4-1 所示。在不方便点击界面的地方，可以运行 etfcmd 文件夹下的 run_etf.bat 脚本来打开它。

图 4-1 RFtester v2.4



4.2 打开蓝牙

使用 USB 线将设备连入电脑，输入 `adb shell setenforce 0` 命令或通过串口直接输入 `setenforce 0` 命令，临时关闭 SELinux 防火墙。（每次重启机器后，都需要进行此操作）

在 BT SETTING 界面先选择正确的串口，Android11 一般使用 `ttyAS11`，其他一般使用 `ttyS1`。然后点击“OPEN BT”按钮，会显示 processing，需要等待几秒钟后方可正常打开 BT，此时该按钮变成绿色。

注意，部分平台的系统 IMG 由于应用权限限制与蓝牙电源控制文件节点的原因，不能够直接点击“OPEN BT”按钮。在打开蓝牙之前，需要先运行 etfcmd 文件夹下的 `bt_on.bat`，再点击“OPEN BT”按钮。与此类似，点击“OPEN OFF”按钮之后，需要运行 `bt_off.bat`。

图 4-2 打开蓝牙界面



图 4-3 打开蓝牙成功界面



4.3 BT Test

RFtester v2.4 暂不支持 BT 进入测试模式的功能。

4.4 参数设置

在 APK 界面，可以点击按钮来设置测试参数。

图 4-4 参数设置界面

4.5 BT TX

在 APK 界面可以进行 BT TX 测试。
界面上提示“status: HCI_SUCCESS!”即指令下发成功。
开始 BT TX 测试后，点击 STOP TX，可以停止 TX。

图 4-5 APK 执行 BT TX 测试的界面

4.6 BT RX

在 APK 界面可以进行 BT RX 测试。与 BT TX 类似，界面上提示“status: HCI_SUCCESS!”即指令下发成功。
开始 BT RX 测试后，点击 STOP RX，可以停止 RX。停止 RX 后，会显示收包统计结果。

图 4-6 APK BT RX 测试的界面



4.7 BLE TX/RX

在 APK 界面可以进行 BLE TX 或者 BLE RX 测试。

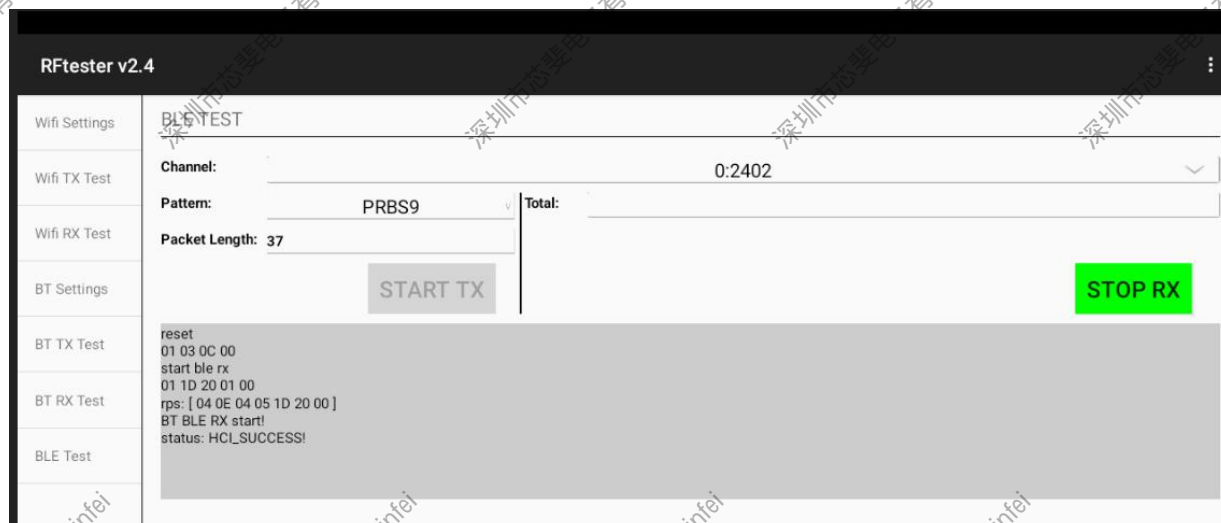
界面上提示“status: HCI_SUCCESS!”即指令下发成功。

开始 BLE TX 或 BLE RX 测试后，点击 STOP TX 或 STOP RX，可以停止测试。如图 4-7 和 4-8 所示。其中，停止 RX 后，会显示收包统计结果。

图 4-7 BLE TX 界面



图 4-8 BLE RX 界面



5 btetf 测试工具使用说明

5.1 帮助说明

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

使用“btetf -h”命令获取工具的使用帮助信息：

```
btetf -h
```

图 5-1 “btetf -h”命令获取工具使用帮助

```
root@TinaLinux:/# btetf -h
btetf -h
BT-ETF Tool Version 1.1.0
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_rssi       Ble Rx Rssi Result
    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
    trace_console  Trace Console Mode
    single_tone    Single Tone Test
    freq_cali      Freq Offset setup
    pwr_cali       Power calibration
```

5.2 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。

5.3 BT 测试

5.3.1 BT RX 测试

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

图 5-2“btetf bt_rx -h” 命令获取 BT RX 测试使用帮助

```

root@TinaLinux:/# btetf bt_rx -h
btetf bt_rx -h
bt_rx: invalid 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:
    btetf bt_rx --bdaddr 11:22:33:44:22:22 --channel_num 8 --link_type 1 --packet_type 3
    
```

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）：

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

图 5-3 BT RX 测试启动示例

```
root@TinaLinux:/# btetf -i hci0 -d bt_rx --bdaddr 11:22:33:44:22:22 --channel_num 1 --link_type 0 --packet_type 4
btetf -i hci0 -d bt_rx --bdaddr 11:22:33:44:22:22 --channel_n
um 1 --link_type 0 --packet_type 4
bt_rx: channel_num(1) link_type/packet_type(0-4) lpt:(0x 4)
< HCI Command: opcode:(0xfc50) ogf-ocf:(0x3f-0x0050) plen:(13)
  10 11 22 33 44 22 22 00 00 00 01 00 04
> HCI Event: 0x0f plen 4
  01 01 50 FC
```

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）：

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

5.3.2 BT RX 测试结束

以下命令用于关闭 BT RX 测试：

```
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：

图 5-4 BT RX 测试结束

```
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 00
  00
event: 04 0E 15 05 50 FC 00 F0 5A 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
```

5.3.3 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：

图 5-5 “btetf bt_tx -h”命令获取 BT TX 测试使用帮助

```
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)
```

图 5-6 “btetf bt_tx -h” 命令获取 BT TX 测试使用帮助

```
[--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
packet_type 3
```

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：

```
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
```


图 5-7 BT TX 测试启动示例

```
root@TinaLinux:/# 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
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
bt_tx: pattern(6)-packet_len(27)-channel_num(1)-power_level(6)-link_type/packet_type(0-4) lpt:(0x 4) hopping_mode(0)
< HCI Command: opcode:(0xfc50) ogf-ocf:(0x3f-0x0050) plen:(13)
00 11 22 33 44 22 22 06 1B 00 01 06 04
> HCI Event: 0x0e plen 17
05 50 FC 00 00 11 22 33 44 22 22 06 1B 00 01 06 04
```

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:

```
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
```

5.3.4 BT TX 测试结束

以下命令用于关闭 BT TX 测试:

```
btetf -i hci0 -d bt_close_tx
```

图 5-8 BT TX 测试结束

```
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
```

5.4 BLE 测试

5.4.1 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:

```
btetf ble_rx -h
```

图 5-9 BLE RX 帮助示例

```
root@TinaLinux:/# btetf ble_rx -h
btetf ble_rx -h
ble_rx: invalid 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
```

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
auto	1	自动测试模式

如：指定在信道 5 监听 BLE 数据包命令如下所示。运行结果如图 5-10 所示。

```
btetf -d -i hci0 ble_rx --rx_channel 5
```

图 5-10 RX 结果图

```
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
```

5.4.2 BLE RX 测试结束

以下命令用于关闭 BLE RX 测试：

```
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：

图 5-11 BLE RX 测试结束，显示收到的包个数

```
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
```

5.4.3 BLE RX 测试结束后 Read RSSI

```
btetf -i hci0 -d ble_rssi
```

图 5-12 BLE RX 测试结束 Read 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 DF
status:0x 0
ble_rssi_result:-32
root@inaLinux:/#
```

5.4.4 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：

```
btetf ble_tx -h
```

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
rate	1M	速率（BLE5.0 以上版本才支持）

目前 XR829 不支持 rate 参数，XR819S 支持 rate 参数。

例子（XR829）：指定在信道 1 发送 BLE 数据包，包长度 37byte，数据为随机填充。

```
btetf -d -i hci0 ble_tx --tx_channel 1 --len 37 --payload 0
```

图 5-13 BLE TX 测试启动示例 (XR829)

```
root@TinaLinux:/# btetf -d -i hci0 ble_tx --tx_channel 1 --len 37 --payload 0
btetf -d -i hci0 ble_tx --tx_channel 1 --len 37 --payload 0
BtFwv:7.12.25
chip is 1
ble_tx: tx_channel: 1, tx_data_len: 37, packet_payload: 0
< HCI Command: opcode:(0x201e) ogf-ocf:(0x08-0x001e) plen:(3)
  01 25 00
> HCI Event: 0x0e plen 4
  05 1E 20 00
```

例子 (XR819S)：指定在信道 1 发送 BLE 数据包，包长度 37byte，数据为随机填充，速率为 1M。

```
btetf -d -i hci0 ble_tx --tx_channel 1 --len 37 --payload 0 --r 1M
```

图 5-14 BLE TX 测试启动示例 (XR819S)

```
root@TinaLinux:/# btetf -d -i hci0 ble_tx --tx_channel 1 --len 37 --payload 0 --r 1M
btetf -d -i hci0 ble_tx --tx_channel 1 --len 37 --payload 0 --
r 1M
BtFwv:9.1.11
chip is 2
ble_tx: tx_channel: 1, tx_data_len: 37, packet_payload: 0, rate: 1
< HCI Command: opcode:(0x2034) ogf-ocf:(0x08-0x0034) plen:(4)
  01 25 00 01
> HCI Event: 0x0e plen 4
  01 34 20 00
```

5.4.5 BLE TX 测试结束

```
btetf -d -i hci0 ble_close
```

图 5-15 BLE TX 测试结束

```
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:/#
```

5.4.6 BLE TX 功率调整

可以使用 “btetf pwr_cali -h” 命令获取功率调整的使用帮助信息。

```
btetf pwr_cali -h
```

BLE TX 测试需要指定以下参数：

默认参数列表	示例值	备注
set	1	TX 功率大小

对于 XR829，工具配置功率值范围是 0~112；对于 XR819S，工具配置功率值范围是 0~255。

例子 (XR829)：指定 BLE TX 功率为最大值。

```
btetf -d -i hci0 pwr_cali --set 112
```

例子 (XR819S)：指定 BLE TX 功率为最大值。

```
btetf -d -i hci0 pwr_cali --set 255
```

使用频谱仪或相关仪器，观察 BLE TX 功率是否有变化。

5.5 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：

```
btetf single_tone --help
```

图 5-16 “btetf single_tone --help” 命令获取蓝牙单载波测试使用帮助

```
root@linux:/# btetf single_tone --help
btetf single_tone --help
Usage:
    single_carrier [option]
Options:
    [--open channel_num]
    [--power_level level]
    [--close]
Example:
    btetf single_tone --open 0 --power_level 6
```

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：

```
btetf -d -i hci0 single_tone --open 1
```

例子：关闭单载波测试：

```
btetf -d -i hci0 single_tone --close
```

5.6 TX 测试



注意

在进行 TX 测试之前，请确保蓝牙固件已经加载。

5.6.1 BT 单载波示例

指定在信道 1 进行单载波测试，power 使用等级 1。

```
btetf -d -i hci0 single_tone --open 1
btetf -d -i hci0 single_tone --close
```

5.6.2 BT TX 测试示例

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。

```
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
btetf -i hci0 -d bt_close_tx
```

5.6.3 BLE TX 测试示例 1

XR829: 指定在信道 1 发送 BLE 数据包，包长度 37byte，数据为随机填充。

```
btetf -d -i hci0 ble_tx --tx_channel 1 --len 37 --payload 0
btetf -d -i hci0 ble_close
```

5.6.4 BLE TX 测试示例 2

XR819S: 指定在信道 1 发送 BLE 数据包，包长度 37byte，数据为随机填充。

```
btetf -d -i hci0 ble_tx --tx_channel 1 --len 37 --payload 0 --r 1M
btetf -d -i hci0 ble_close
```

5.7 RX 测试



注意

在进行 TX 测试之前，请确保蓝牙固件已经加载。

5.7.1 BT RX 测试示例

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）。

```
btetf -i hci0 -d bt_rx --bdaddr 11:22:33:44:22:22 --channel_num 1 --link_type 2 --packet_type 15
btetf -i hci0 -d bt_close_rx
```

5.7.2 BLE RX 测试示例

指定在信道 5 监听 BLE 数据包。

```
btetf -d -i hci0 ble_rx --rx_channel 5
btetf -d -i hci0 ble_close
```


6 FAQ

6.1 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。

6.1.1 不能打开串口

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

图 6-1 运行“init_test.bat”脚本出现“Can't open serial port”问题



```
adb 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
```

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

```
@echo off
adb root
adb remount

::kill_hciattach
echo kill_hciattach
adb shell "ps -A | grep hciattach | busybox awk '{print $2}'" >xrTtemp.txt
for /f %i in (xrTtemp.txt) do (
    set pid=%i
)
echo adb shell kill -9 %pid%
del xrTtemp.txt
if "%pid%"==" " (goto start_bt_fw_download)
adb shell kill -9 %pid%

:start_bt_fw_download
start "" /b cmd /c "adb shell hciattach -n ttyS1 xradio &"
ping 1.1.1.1 -n 15 > nul
echo brom_doneadb root
adb remount
```

其中，蓝牙对应的 tty 编号可以通过查看 /dev/ 目录获得：

图 6-2 查看设备上蓝牙对应的 tty 编号

```
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
crwxrwxrwx 1 media    mediadm  250, 0 2019-04-28 10:48 tee0
crwxrwxrwx 1 media    mediadm  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 #
```

6.1.2 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）：

图 6-3 Brom 同步失败

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

可能原因如下：

1) 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 hciattach 进程，若存在则表明已进行相关配置，无需重复加载固件。

```
ps
kill -9 PID
```

图 6-4 检查蓝牙是否已加载

```
1438 root      716 S      /usr/sbin/MtpDaemon -D
1441 root      576 S      /sbin/swupdate-progress -w
1453 root     1068 S<     /usr/sbin/ntpd -n -N -S /usr/sbin/ntpd-hotplug -p nt
1469 root     2740 S      tt
1528 root      720 S      hciattach -n ttyS1 xradio
1542 root     1384 S      wpa_supplicant -iwlan0 -Dnl80211 -c/etc/wifi/wpa_sup
1547 root          0 SW<     [kworker/u5:0]
1548 root          0 SW<     [hci0]
1549 root          0 SW<     [hci0]
1552 root          0 SW<     [kworker/u5:1]
1578 root     1068 R      ps
root@TinaLinux:/# kill -9 1528
kill -9 1528
```

2) 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：

图 6-5 检查芯片 core 是否已 reset

```
root@TinaLinux:/# cat /sys/class/rfkill/rfkill0/state
cat /sys/class/rfkill/rfkill0/state
1
```

3) 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

图 6-6 检查芯片是否 wake up

```
root@TinaLinux:/# cat /proc/bluetooth/sleep/btwake
cat /proc/bluetooth/sleep/btwake
bt wake state:1
```

6.2 收发包异常

图 6-7 收发包数量图

```
btetf -d -i hci0 ble_close
< HCI Command: opcode:<0x201f> ogf-ocf:<0x08-0x001f> plen:<0>
> HCI Event: 0x0e plen 6
04 1f 20 00 01 00
event: 04 0E 06 04 1F 20 00 01 00
status:0x 0
send_or_receive_packet:1
root@none):/mnt/app# btetf -d -i hci0 ble_rx --rx_channel 12
btetf -d -i hci0 ble_rx --rx_channel 12
ble_rx:rx_channel:12
< HCI Command: opcode:<0x201d> ogf-ocf:<0x08-0x001d> plen:<1>
0C
> HCI Event: 0x0e plen 4
04 1D 20 00
```

这种情况可能是因为其他业务打断了该业务，可通过以下命令查看

```
hcidconfig -a
```

从图 6-8 中可看出此时 BT 正在执行 scan 业务。

图 6-8 hciconfig 结果图

```
root@TinaLinux:/# hciconfig -a
hciconfig -a
hci0:  Type: Primary  Bus: UART
      BD Address: 22:22:C8:2C:21:87  ACL MTU: 1021:8  SCO MTU: 255:4
      UP RUNNING PSCAN ISCAN
      RX bytes:1274 acl:0 sco:0 events:70 errors:0
      TX bytes:1804 acl:0 sco:0 commands:70 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: 'aw-2C21-bt-test'
      Class: 0x240000
      Service Classes: Rendering, Audio
      Device Class: Miscellaneous,
      HCI Version: 4.1 (0x7)  Revision: 0xc19
      LMP Version: 4.1 (0x7)  Subversion: 0xc19
      Manufacturer: not assigned (1597)
```

使用以下命令进行关闭 BT SCAN 业务。

```
hciconfig hci0 noscan
```

再次通过以下命令查看，发现已经关掉了

```
hciconfig -a
```

图 6-9 hciconfig 结果图

```
root@TinaLinux:/# hciconfig -a
hciconfig -a
hci0:  Type: Primary  Bus: UART
      BD Address: 22:22:C8:2C:21:87  ACL MTU: 1021:8  SCO MTU: 255:4
      UP RUNNING
      RX bytes:1561 acl:0 sco:0 events:74 errors:0
      TX bytes:1821 acl:0 sco:0 commands:74 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: 'aw-2C21-bt-test'
      Class: 0x240000
      Service Classes: Rendering, Audio
      Device Class: Miscellaneous,
      HCI Version: 4.1 (0x7)  Revision: 0xc19
      LMP Version: 4.1 (0x7)  Subversion: 0xc19
      Manufacturer: not assigned (1597)
```

著作权声明

版权所有©2020 广州芯之联科技有限公司。保留一切权利。

本文档及内容受著作权法保护，其著作权由广州芯之联科技有限公司（“芯之联”）拥有并保留一切权利。

本文档是芯之联的原创作品和版权财产，未经芯之联书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本文档内容的部分或全部，且不得以任何形式传播。

商标声明



（不完全列举）均为广州芯之联科技有限公司的商标或者注册商标。在本文档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

免责声明

您购买的产品、服务或特性应受您与广州芯之联科技有限公司（“芯之联”）之间签署的商业合同和条款的约束。本文档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本文档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，芯之联概不负责。

本文档作为使用指导仅供参考。由于产品版本升级或其他原因，本文档内容有可能修改，如有变更，恕不另行通知。芯之联尽全力在本文档中提供准确的信息，但并不确保内容完全没有错误，因使用本文档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，芯之联概不负责。本文档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本文档未以明示或暗示或其他方式授予芯之联的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。芯之联不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。芯之联不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。