



# XRADIO WLAN RFTTest 工具 使用指南

版本号：1.3

发布时间：2021-05-29

# 版本历史

版本	日期	责任人	版本描述
1.3	2021-05-29	AWA 1605	添加 SDD 文件读写注意事项，修改 etf 工具版本说明（附录 A）
1.2	2021-05-07	AWA 1605	更新使用说明，添加 efuse 读写说明
1.1	2020-11-20	AWA 1605	添加读写 SDD 操作命令。
1.0	2020-11-03	AWA 1605	创建文档。



# 目录

版本历史.....	i
目录.....	ii
图片目录.....	iv
表格目录.....	v
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 使用说明.....	5
3.1.1 Rftester 安装.....	5
3.1.2 ETF CLI 安装.....	5
3.2 注意事项.....	6
3.2.1 确认设备存在 fw、sdd 等文件.....	6
3.2.2 测试前确认 WLAN 关闭.....	6
3.2.2.1 Android 平台（单个 ko 平台均可参考）.....	6
3.2.2.2 Tina 平台（三个 ko 平台均可参考）.....	7
4 WLAN Rftester 使用说明.....	8
4.1 常规说明.....	8
4.2 TX 测试.....	9
4.3 RX 测试.....	10
5 WLAN ETF CLI 使用说明.....	12
5.1 常规说明.....	12
5.1.1 帮助手册.....	12
5.1.2 启动 etf 模式.....	12
5.1.3 使能 PHY.....	12
5.1.4 MAC 地址读写.....	13

5.1.5 设置频段.....	13
5.1.6 设置信道.....	13
5.1.7 设置速率.....	14
5.1.8 设置 TX 功率.....	14
5.1.9 频偏读写.....	15
5.1.10 设置带宽（仅适用于支持 HT40 的芯片/设备）.....	15
5.1.11 设置子信道（仅适用于支持 HT40 的芯片/设备）.....	15
5.2 SDD 文件读写.....	16
5.2.1 获取 SDD 文件.....	16
5.2.2 获取 sdd 频偏值.....	16
5.2.3 设置 sdd 频偏值.....	16
5.2.4 获取 sdd 功率值.....	17
5.2.5 设置 sdd 功率值.....	17
5.3 efuse 读写.....	17
5.3.1 读取 CHIPID.....	17
5.3.2 功率校准.....	18
5.3.3 频偏校准.....	18
5.3.4 注意事项.....	19
5.4 TX 测试.....	19
5.4.1 TX 测试示例 1.....	20
5.4.2 TX 测试示例 2.....	20
5.4.3 TX 测试示例 3.....	20
5.4.4 TX 测试示例 4.....	21
5.4.5 TX 测试示例 5.....	21
5.5 RX 测试.....	21
5.5.1 RX 测试示例 1.....	22
5.5.2 RX 测试示例 2.....	22
6 常见问题与解决.....	23
附录 A： ETF CLI 版本说明.....	25

# 图片目录

图 3-1 PC、被测设备与仪器的连接关系.....5

图 3-2 确认设备存在 fw、sdd 等文件.....6

图 3-3 单 ko 形式测试前确认模块是否存在.....6

图 3-4 三个 ko 形式测试前确认 xradio\_wlan 模块是否存在.....7

图 4-1 寻找 RFtester v2.4 图标.....8

图 4-2 RFtester 中 Wi-Fi Settings 界面.....8

图 4-3 RFtester 中 TURN-ON-WIFI 后的 Wi-Fi Settings 界面.....9

图 4-4 RFtester 中 Start TX Test 设置界面.....9

图 4-5 RFtester 中“START RX TEST”设置界面.....10

图 5-1 etf help 命令示例.....12

图 5-2 etf connect 命令示例.....12

图 5-3 etf enable\_phy 命令示例.....13

图 5-4 etf get\_mac/etf set\_mac 命令示例.....13

图 5-5 etf phy\_mode 命令示例.....13

图 5-6 etf channel 命令示例.....14

图 5-7 etf rate 命令示例.....14

图 5-8 etf power\_level 命令示例.....14

图 5-9 etf get\_freq\_offset/etf set\_freq\_offset 命令示例.....15

图 5-10 etf bandwidth/etf subchannel 命令示例.....15

图 5-11 etf subchannel 命令示例.....15

图 5-12 40M LOWER 和 UPPER 概念.....15

图 5-13 etf get\_sdd\_file 命令示例.....16

图 5-14 get\_sdd\_freq\_offset 命令示例.....16

图 5-15 set\_sdd\_freq\_offset 命令示例.....16

图 5-16 get\_sdd\_power 命令示例.....17

图 5-17 get\_sdd\_power 命令示例.....17

# 表格目录

表 2-1	ETF CLI 测试工具主要功能： .....	3
表 5-1	ETF CLI 速率配置参数： .....	14
表 6-1	etf 无法使用可能原因与解决方法： .....	23
表 6-2	etf 无法收发帧可能原因与解决方法： .....	24
表 A-1	ETF CLI 版本更新内容以及依赖关系.....	25



# 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与解决。下面将以 A100+XR829 (Android10) 平台为例介绍 RfTester.apk 安装、使用方法, 以 R328+XR829 (Tina Linux-4.9) 为例介绍 ETF CLI 工具的安装、使用方法。列举部分测试用例, 其他平台、模组以此作为参考。

## 1.2 目标读者

从事 Wi-Fi 模块开发和测试的相关人员。





## 1.3 适用范围

适用于 Android 或 Linux 环境下 XR 系列 Wi-Fi 芯片 RF 测试。

## 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
 <b>警告</b>	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。
 <b>注意</b>	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。
 <b>说明</b>	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。
 <b>窍门</b>	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

为满足 WLAN RF 性能、硬件布线等方面测试需求，Xradio 提供了两种 WLAN RF 测试工具，分别是 RFtester（Android APK 工具）以及 ETF CLI（Linux Command Line）。这两款工具实现类似的功能，RFtester 工具主要用于带有屏显的 Android 系统，ETF 工具则用于无屏的 Android 或者 Linux 系统，以及自动化测试场景。

本文以 Android10（单 ko：xr829.ko）和 Tina（Linux-4.9，三个 ko：xradio\_wlan.ko、xradio\_core.ko、xradio\_mac.ko）平台为例，介绍这两款工具的安装和使用方法，其他平台、模组（XR819、XR819S）工具的安装和使用方法请参考例子。WLAN ETF 大致功能如下：

**表 2-1 ETF CLI 测试工具主要功能：**

类别	测试支持	描述	备注
基本配置	频段选择	测试的频段可配置	目前只支持 2.4G
	信道选择	测试信道可配置	可配置 1 至 14 信道
	带宽选择	可配置带宽为 20MHz 或 40MHz	
	MAC 地址配置	修改发送帧的 MAC 地址	可配置 A1, A2, A3
TX	速率选择	可选择 11b, 11g, 11n 的速率	11b 22Mbps 33Mbps 除外
	连续发送模式	连续发送模式下不断发送帧，直到进行停止发送操作	
	帧数发送模式	发送一定数目的帧后停止发送	
	单载波发送模式	可发送单载波，幅度可调整	CLI 支持频偏可调整
	帧数配置	对发送帧的数量进行配置，仅适用于帧数发送模式。	
	帧长度配置	调整发送的帧长度	大于 MAC 头部，小于 4096
	功率调整	发送功率可以按等级调整，单位不是 dbm	每个速率有对应的默认功率，一般情况下不用调整
	11n tx mode 选择	11n 模式下，选择 Premble 和 GuardInterval 的模式。对于 Premble，可选择 Mixed/GreenField；对于 GuardInterval，可选择 Long/Short GI。	注意，GreenField 只能和 Long GI 搭配使用。

类别	测试支持	描述	备注
	11b 长短前导选择	11b 模式下，可选择 Long/Short Preamble	
RX	连续接收	停止接收后显示接收帧总数、错误帧数	
	模式配置	可以配置 11b only、11g/n 或者 11b/g/n	APK 暂不支持 11b only 和 11g/n 模式


**注意**

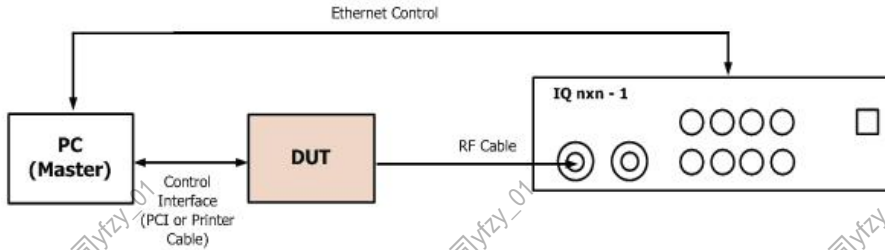
XR819、XR819S 不支持 HT40，不需要对它们进行 40M 带宽的相关测试。



### 3 使用说明

使用工具对芯片的 RF 参数进行测试，通常还需要 PC、测试仪器来辅助测试，测试仪器一般为 IQ 或频谱仪。他们的连接关系如图 3-1 所示：

图 3-1 PC、被测设备与仪器的连接关系



#### 3.1.1 Rftester 安装

- A) 请确认测试系统版本在 Android10 及以上；
- B) 通过 adb 工具安装 Rftester\_v2.4.apk。

```
adb root
adb remount
adb install /XXX/Rftester_v2.4.apk //“xxx”为具体的目录
```

#### 3.1.2 ETF CLI 安装

方法一：脚本安装

双击 etf\_cli 目录下 load\_etf\_cli-\*\*\*.bat 脚本，“\*\*\*”为 Tina 或 Android

方法二：手动安装

- A) 将带有版本号的 etf 命令行工具（例如“etf\_version\_2\_0\_5”）重命名为 etf。
- B) 通过 adb 工具把 etf 工具下载到指定目录。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。
- C) 修改权限：XXX/bin/etf 权限改为 755。

```
/* xxx 为本地存放 etf 工具的具体目录，Android 平台 */
adb push xxx/etf /vendor/bin/etf
adb shell chmod 755 /vendor/bin/etf
```

```
/* xxx 为本地存放 etf 工具的具体目录，Tina 平台 */
adb push xxx/etf /usr/bin/etf
adb shell chmod 755 /usr/bin/etf
```

## 3.2 注意事项

### 3.2.1 确认设备存在 fw、sdd 等文件

Android 平台 fw、sdd 等文件存放的目录一般为 `/vendor/etc/firmware`，Tina 平台存放的目录为 `/lib/firmware`，到对应目录下使用 `ls -l` 命令查看确认，如图 3-2，Android10 XR829 平台 `/vendor/etc/firmware` 目录文件：

图 3-2 确认设备存在 fw、sdd 等文件

```
ceres-b3:/vendor/etc/firmware # ls -l
total 565
-rw-r--r-- 1 root root 2184 2020-07-13 09:44 boot_xr829.bin
-rw-r--r-- 1 root root 192 2020-07-13 09:44 bt_sdd.bin
-rw-r--r-- 1 root root 77136 2020-12-26 17:05 etf_xr829.bin
-rw-r--r-- 1 root root 174624 2020-07-13 09:44 fw_xr829.bin
-rw-r--r-- 1 root root 311368 2020-09-09 13:22 fw_xr829_bt.bin
-rw-r--r-- 1 root root 744 2021-04-26 17:46 sdd_xr829.bin
```

其他模组参考上面的方法，在对应的目录下找到 `boot_xr***.bin`、`etf_xr***.bin`、`sdd_xr***.bin` 和 `fw_xr***.bin` 四个 bin 文件（\*\*\*为芯片型号，如 819,819s）。

### 3.2.2 测试前确认 WLAN 关闭

由于 WLAN 与 RF 测试共用一个驱动，但下载的固件不一样，因此两者互斥。在 RF 测试之前请确认 WLAN 处于关闭状态，否则 RF 测试模式会开启失败。如果 Android 系统打开了随时扫描、无线位置等功能会导致 WLAN 没有真正关闭，可以通过是否加载了对应的驱动来确认，如果加载了驱动，需将驱动先卸载掉。

#### 3.2.2.1 Android平台（单个ko平台均可参考）

图 3-3 单 ko 形式测试前确认模块是否存在

```
ceres-b3:/vendor/etc/firmware # lsmod
Module              Size Used by
gs1X680new          1118208 0
xr829                925696 0
pursrvkm            1921024 51
xradio_bt_lpm       28672 0
vin_v412             237568 0
gc0310_mipi         24576 0
gc2355_mipi         20480 0
gc030a_mipi         20480 0
gc2385_mipi         20480 0
vin_io               73728 5 vin_v412,gc0310_mipi,gc2355_mipi,gc030a_mipi,gc2385_mipi
videobuf2_v412       28672 1 vin_v412
videobuf2_dma_contig 24576 1 vin_v412
videobuf2_memops    16384 1 videobuf2_dma_contig
videobuf2_core       49152 2 vin_v412,videobuf2_v412
ceres-b3:/vendor/etc/firmware #
```

如图 3-3，此时已经加载了 XR829 驱动，需要将其卸载。在 Wi-Fi 设置界面点击关闭 Wi-Fi 按钮或使用命令将驱动卸载，卸载成功后再以 etf 的模式加载驱动。

```
rmmod xr829
insmod /vendor/modules/xr829.ko etf_enable=1 //Android11 路径为"/vendor/lib/modules/"
```

重新加载 etf 模式驱动后才能使用 RF 测试工具，RF 测试完成后，恢复 Wi-Fi 正常的功能执行以下操作：

```
rmmod xr829
insmod /vendor/modules/xr829.ko
```



注意

XR819 或 XR819S 用的是同一个驱动：**xr819.ko**，因此 XR819 或 XR819S 在以上操作都只需将 xr829 替换成 xr819，且保证存在对应芯片型号四个 bin 文件即可。

### 3.2.2.2 Tina平台（三个ko平台均可参考）

图 3-4 三个 ko 形式测试前确认 xradio\_wlan 模块是否存在

```
root@TinaLinux:/lib/modules/4.9.191# lsmod
fuse                73446  0
snd_mixer_oss       12405  1 snd_pcm_oss
snd_pcm_oss         33238  0
snd_seq_device      2999   0
xradio_core         428562  1 xradio_wlan
xradio_mac          217941  1 xradio_core
xradio_wlan         690    0
root@TinaLinux:/lib/modules/4.9.191#
```

如图 3-4，此时 xradio\_mac.ko、xradio\_core.ko 和 xradio\_wlan.ko 都已经加载，需要卸载 xradio\_wlan.ko 且保留 xradio\_mac.ko 和 xradio\_core.ko 才能使用 RF 测试工具：

```
rmmod xradio_wlan //必须确保已经加载"xradio_mac"和"xradio_core"
```



注意

三个 ko 形式运行 etf 前必须确保已经加载"xradio\_mac"和"xradio\_core"

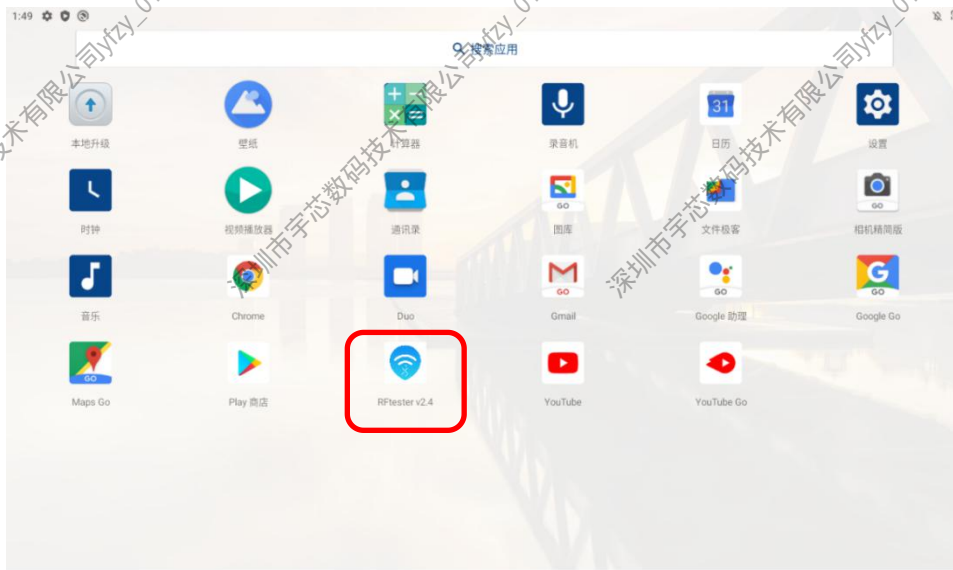
测试完成后，加载 xradio\_wlan.ko 即可正常使用 Wi-Fi。

# 4 WLAN RFtester 使用说明

## 4.1 常规说明

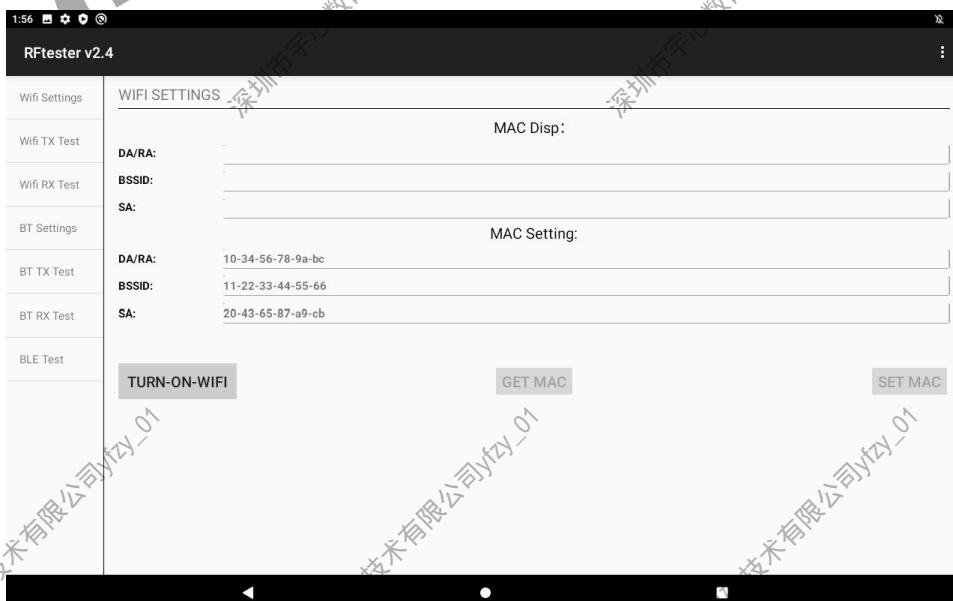
- 1) 工具安装请参考 3.1.1 小节。
- 2) 使用 USB 线将设备连入电脑，输入“adb shell setenforce 0”命令或通过串口直接输入“setenforce 0”命令，临时关闭 SELinux 防火墙。（每次重启机器后，都需要进行此操作）
- 3) 找到 RFtester v2.4 图标

图 4-1 寻找 RFtester v2.4 图标



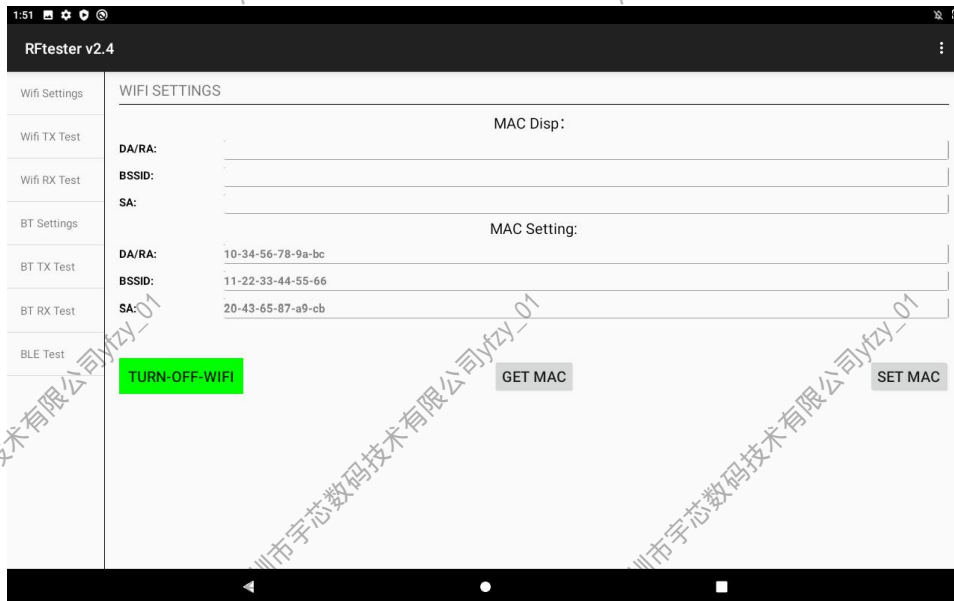
- 4) 点击图标打开软件。
- 5) 打开软件后，会默认显示 Wi-Fi Settings 界面，如图 4-2：

图 4-2 RFtester 中 Wi-Fi Settings 界面



6) TURN-ON-WIFI 按钮用于开启 PHY。点击后，按钮显示为 TURN-OFF-WIFI。此后才可进行相关的配置和 TX/RX 测试。再次点击此按钮，则关闭 PHY。在 START TX 或 RX 前，必须在 Wi-Fi Settings 界面中成功开启 PHY。

图 4-3 RfTester 中 TURN-ON-WIFI 后的 Wi-Fi Settings 界面



7) 在 TURN-ON-WIFI 之后，可以 Get MAC 和 Set MAC。前者会获取当前 PHY 配置的 MAC 信息，包括待发送帧的目的 MAC 地址、BSSID 地址和源 MAC 地址。后者会将 MAC Setting 中的地址信息写入到 PHY 中。在 MAC Setting 中按照提示格式 XX-XX-XX-XX-XX 输入 MAC 地址信息，点击 Set MAC 地址按钮，将键入的 MAC 地址设置配置到 PHY 中。

## 4.2 TX 测试

点击侧控制区中的 Wifi TX Test，出现 START TX TEST 界面，如图 4-4

图 4-4 RfTester 中 Start TX Test 设置界面





**注意**  
在结束测试后，需把测试关闭，否则设备一直处于工作状态，可能会导致电池耗光。

“Wifi TX Test”中支持多种配置选择，具体的配置选项如下：

名称	说明
Band	仅支持 2.4G
Bandwidth	仅支持 20MHz
Channel	20MHz 模式下频段的选择配置
Data Rate	支持 11b/g/n 共 20 种速率发送。在选择 11b 速率时，可以在 radioButton 中选择 L/S Preamble 发送；在选择 11n 速率时，可以在 radioButton 中选择 Mix/Green 和 Long/Short GI 发送
TX Mode	可以选择 PACKETS/CONTINUOUS/SINGLE TONE 方式。
PACKETS	需要在 Packet Counts 和 Packet Length 中指定待发送的包数目和包长度(包长度限定在 1~4095)
CONTINUOUS	按照指定的包长度发包，持续到按下 STOP TX 按钮
SINGLE TONE	持续发送单载波，持续到按下 Stop TX 按钮
TX power Level	调整发送功率。(power_level - 2)/4 等于对应的 dB 值

### 4.3 RX 测试

点击左侧控制区中的 Wifi RX Test，出现“START RX TEST”界面，如图 4-5：

图 4-5 Rftester 中“START RX TEST”设置界面



- 1) 在 RX 配置区域可以对带宽、信道进行选择；可以选择是否产生 Ack 包，并指定产生 Ack 包的 delay 时间，范围为 0~1024us。



2) RX 结果显示区显示 RX 的包数目和 CRC 错误的包数目



## 5 WLAN ETF CLI 使用说明

ETF CLI 工具可用于手动测试，也可以被其他程序调用进行自动化测试。环境搭建参考 3.1.2 小节。



**注意**

若 ETF CLI 工具版本不匹配时请根据提示信息更换正确版本，或参考附录 A。

### 5.1 常规说明

#### 5.1.1 帮助手册

ETF 命令行工具的基本格式如下（可以通过 `etf help` 获取 ETF 工具详细的帮助信息）

```
etf cmd [param0] [param1] [param2] [param3]
```

图 5-1 etf help 命令示例

```
root@TinaLinux:~# etf help
ETF version: 2.0.5 Apr 26 2021 14:14:26

connect          before sending any other cmds,send this cmd first to
disconnect       send this cmd after all of other cmds!
enable_phy       default is ENABLE,include set mode,channel,pwr leve
help             [-options]          Show this help msg, add options for details
tx               [-options]          [param]          for detail: help tx
tx_stop          stop continuous tx
rx               to recieve packets
rx_stop          stop recieving packets
tone             [-options]          [param]          single tone functions,for detail: help tone
tone_stop        stop single tone operation
phy_mode         [OFDM_2GHZ/DSSS_2GHZ/2GHZ] default is 2GHZ, and 5GHZ is not support yet!
channel          [OFDM_2GHZ/DSSS_2GHZ/2GHZ] [channelNum] here phy mode param is only
rate             [-options]          [param]          for detail: help rate
get_mac          to get the mac addresses right now
set_mac          [-options]          [param]          set mac address,for detail: help set_mac
power_level      [2~120]              set the tx pwr level
get_antifuse     get the antifuse bits right now
bandwidth        [40M/20M]           auto set to 20M if invalid input detected
subchannel       [LOWER/UPPER]       auto set to UPPER if invalid input detected
read_efuse       [CHIPID/DCXO_TRIM/POUT_CAL/MAC/USER_AREA] [param]          When reading user_a
write_efuse      [CHIPID/DCXO_TRIM/POUT_CAL/MAC/USER_AREA] [param]          Only support writti

Input [help] for help message.Add options for more details!
root@TinaLinux:~#
```

#### 5.1.2 启动 etf 模式

启动 RF 测试模式，使设备处于运行状态（其他测试命令只能在该命令完成以后才能进行）

```
etf connect
```

图 5-2 etf connect 命令示例

```
root@TinaLinux:~# etf connect
sh: pkill: not found
ETF fw is running!!

Sending CMD OK!
root@TinaLinux:~#
```

#### 5.1.3 使能 PHY

在进行 RF 测试相关操作之前如 TX 或 RX 必须先使能 PHY。

```
etf enable_phy
```

图 5-3 etf enable\_phy 命令示例

```
root@TinaLinux:/# etf enable_phy
power level:66
Transmit gain register value:
digital:      346
analogue:     5
index:        0

Phy mode is OFDM_DSSS_2GHZ!
Channel is 7!

Calibrations:
lo_start:     -41
im_start:     -21
lo_stop:      -47
im_stop:      -40
gain:         -120
phase:        -351
i_ofs:        -6
q_ofs:        -9

Sending CMD OK!
root@TinaLinux:/#
```

### 5.1.4 MAC 地址读写

获取和配置 MAC 地址。其中-d 为目的地址 (A1) , -s 为源地址 (A2) , -t 为 BSSID (A3)

```
etf get_mac
```

```
etf set_mac -d XX:XX:XX:XX:XX:XX -s XX:XX:XX:XX:XX:XX -t XX:XX:XX:XX:XX:XX
```

图 5-4 etf get\_mac/etf set\_mac 命令示例

```
root@TinaLinux:/# etf get_mac
Source mac address: 12:34:56:78:AA:CC
Dest mac address:  11:22:33:44:55:BB
BSSID:             11:22:33:44:55:66
Sending CMD OK!
root@TinaLinux:/# etf set_mac -d 11:22:33:44:55:BB -s 12:34:56:78:AA:CC -t 8c:45:00:12:bc:58
Dest mac address:  11:22:33:44:55:BB
Source mac address: 12:34:56:78:AA:CC
BSSID:             8C:45:00:12:BC:58
Sending CMD OK!
root@TinaLinux:/#
```

### 5.1.5 设置频段

设置 PHY 模式。其中 mode 可为 DSSS\_2GHZ, OFDM\_2GHZ, 2GHZ, 默认为 2GHZ

```
etf phy_mode [mode]
```

图 5-5 etf phy\_mode 命令示例

```
root@TinaLinux:/# etf phy_mode 2GHZ
Phy mode is OFDM_DSSS_2GHZ!
Regulatory class is 0!
Channel is 7!
Sending CMD OK!
root@TinaLinux:/#
```

### 5.1.6 设置信道

设置特定频段模式下的信道, mode 如上, channel\_num 为 1~14

```
etf channel [mode] [cahnnel_num]
```

图 5-6 etf channel 命令示例

```
root@TinaLinux:~# etf channel 2GHZ 3
channel 3 ok
band = 2GHZ
Here phy mode param is only for channel check!
Sending CMD OK!
root@TinaLinux:~#
```

### 5.1.7 设置速率

当前支持速率有 20 种，配置对应的模式后，只能选择该模式下支持的速率，具体请参考表 5-1。

```
etf rate -m[x] -r[y] //模式 x 和速率 y 的值参考表 5-1
```

表 5-1 ETF CLI 速率配置参数：

模式 x	定义	对应速率 y
0	11b short preamble	2, 5.5, 11
1	11b long preamble	1, 2, 5.5, 11
2	11g	6, 9, 12, 18, 24, 36, 48, 54
4	11n Greenfield	
5	11n Mixed	6.5, 13, 19.5, 26, 39, 52, 58.5, 65

图 5-7 etf rate 命令示例

```
root@TinaLinux:~# etf rate -m 5 -r 65
mode is _802_11_N_OFDM_LONG_MIXED_
rate is _65_Mbps_64QAM_Code_5_6_
power level:58
Transmit gain register value:
digital:      276
analogue:    5
index:       0
Sending CMD OK!
root@TinaLinux:~#
```

### 5.1.8 设置 TX 功率

对应的速率有默认功率，如需自定义功率 TX 请在“etf tx”的上一步执行该指令

```
etf power_level [num] //num 可设置 2~120
```

图 5-8 etf power\_level 命令示例

```
root@TinaLinux:~# etf power_level 72
Power set to 72 !
Transmit gain register value:
digital:      310
analogue:    12
index:       0
Sending CMD OK!
root@TinaLinux:~#
```

其中 num 的范围：2~120，每个速率有对应的默认功率和最大功率，速率配置后自动使用默认功率进行发送；当功率调整超过最大功率时，会配置为最大功率。

### 5.1.9 频偏读写

```
etf get_freq_offset //获取当前频偏值
etf set_freq_offset [num] //设置频偏值，参考范围：0~127 退出 etf 之后恢复至 sdd 文件设置值
```

图 5-9 etf get\_freq\_offset/etf set\_freq\_offset 命令示例

```
root@TinaLinux:~# etf get_freq_offset
Cur freq_offset = 55.
Sending CMD OK!
root@TinaLinux:~# etf set_freq_offset 100
Cur freq_offset = 100.
Sending CMD OK!
root@TinaLinux:~#
```

### 5.1.10 设置带宽（仅适用于支持 HT40 的芯片/设备）

```
etf bandwidth [bw] //bw 可为 20M、40M
```

图 5-10 etf bandwidth/etf subchannel 命令示例

```
root@TinaLinux:~# etf bandwidth 40M
Band width set to 40MHz!
Please set channel after setting bandwidth!
Sending CMD OK!
```

此后需要设置子信道，即通过 etf subchannel 命令去设置。

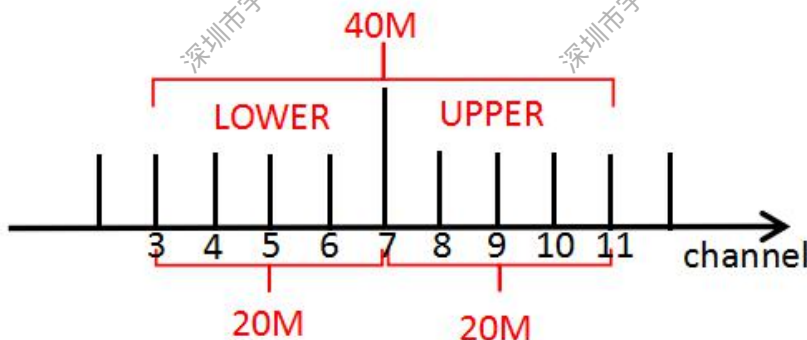
### 5.1.11 设置子信道（仅适用于支持 HT40 的芯片/设备）

```
etf subchannel [sub_ch] //sub_ch 可为 LOWER、UPPER
```

图 5-11 etf subchannel 命令示例

```
root@TinaLinux:~# etf subchannel LOWER
Set sub channel lower!
Please set channel after setting subchannel!
Sending CMD OK!
```

图 5-12 40M LOWER 和 UPPER 概念



HT40 是由两个相邻的 20M 带宽信道组成一个 40M 带宽的信道，其中一个作为主信道，另一个作为子信道，40M 带宽信道的中心频点在主信道和子信道的中间。子信道可设置为 LOWER 和 UPPER，当子信道的中心频率大于 40M 带宽的中心频率时，应将子信道设置为 UPPER，反之设为 LOWER。

图 5-12 中主信道是 5 而子信道为 UPPER，和主信道是 9 而子信道为 LOWER，两者组成 40M 带宽的中心频率都在 7 信道（2442MHz）。40M 中心频率的计算方法如下：

etf bandwidth 40M etf subchannel UPPER etf channel 2GHZ 5 //主信道为 5 (2432MHz) $F_0 = 2432 + 10 = 2442(\text{MHz})$ 即 7 信道
etf bandwidth 40M etf subchannel LOWER etf channel 2GHZ 9 //主信道为 9 (2452MHz) $F_0 = 2452 - 10 = 2442(\text{MHz})$ 即 7 信道

## 5.2 SDD 文件读写



**注意**  
Linux-5.4 内核对文件的读写有更高的要求，因此该功能不适用于 Linux-5.4 内核的系统。

### 5.2.1 获取 SDD 文件

```
etf get_sdd_file
```

图 5-13 etf get\_sdd\_file 命令示例

```
root@TinaLinux:~# etf get_sdd_file
sdd_path = /lib/firmware/sdd_xr829.bin
sddLength = 744
00000000h: c0 02 02 03 c1 06 00 01 00 00 4c 00 c4 0a 00 00
00000010h: 00 00 00 00 00 00 00 00 c5 02 c0 5d c7 02 05 00
```

### 5.2.2 获取 sdd 频偏值

```
etf get_sdd_freq_offset
```

图 5-14 get\_sdd\_freq\_offset 命令示例

```
root@TinaLinux:~# etf get_sdd_freq_offset
sdd_path = /lib/firmware/sdd_xr829.bin
sdd_freq_offset = 55
Sending CMD OK!
root@TinaLinux:~#
```

### 5.2.3 设置 sdd 频偏值

```
etf set_sdd_freq_offset
```

图 5-15 set\_sdd\_freq\_offset 命令示例

```
root@TinaLinux:~# etf set_sdd_freq_offset 56
sdd_path = /lib/firmware/sdd_xr829.bin
Sending CMD OK!
root@TinaLinux:~#
```

### 5.2.4 获取 sdd 功率值

```
etf get_sdd_power //显示 11b, 11gn 以及 MCS7 速率的功率
etf get_sdd_power all //查看所有速率对应的功率
```

图 5-16 get\_sdd\_power 命令示例

```
root@TinaLinux:/# etf get_sdd_power
sdd_path = /lib/firmware/sdd_xr829.bin
sdd user power:
  11b: 66
  11gn: 62
  MCS7: 58
Sending CMD OK!
root@TinaLinux:/# etf get_sdd_power all
sdd_path = /lib/firmware/sdd_xr829.bin
sdd user power:
  modulation  rate      power
  DSSS        1.2       : 66
  CCK         5.5,11    : 66
  BPSK_1_2    6.6,5     : 62
  BPSK_3_4    9         : 62
  QPSK_1_2    12, 13    : 62
  QPSK_3_4    18, 19.5  : 62
  16QAM_1_2   24,26     : 62
  16QAM_3_4   36,39     : 62
  64QAM_2_3   48,52     : 62
  64QAM_3_4   54, 58.5  : 62
  64QAM_5_6   65        : 58
  null        null     : 0
Sending CMD OK!
```

### 5.2.5 设置 sdd 功率值

```
etf set_sdd_power 11b [pwr] //设置 11b 功率，此处还可以设置 11gn、mcs7 和对应调制方式的功率
etf set_sdd_power all [11b_pwr] [11gn_pwr] [mcs7_pwr] //设置 11b、11gn 和 mcs7 速率的功率
```

图 5-17 get\_sdd\_power 命令示例

```
root@TinaLinux:/# etf set_sdd_power 11gn 60
sdd_path = /lib/firmware/sdd_xr829.bin
Sending CMD OK!
root@TinaLinux:/# etf set_sdd_power all 66 62 58
sdd_path = /lib/firmware/sdd_xr829.bin
Sending CMD OK!
root@TinaLinux:/#
```



注意

SDD 文件不可轻易改动，否则可能会引发性能问题。

写操作一般会有权限限制，需要先执行 adb root; adb remount

## 5.3 efuse 读写

该功能目前仅在 XR829 上实现，XR819 和 XR819S 暂不支持。efuse 实时读写即通过 ETF CLI 工具对芯片的 efuse 区域进行读写操作，下面将列举两个常用功能。

### 5.3.1 读取 CHIPID

通过读取 chip\_id 来识别芯片的编码信息，不同芯片会有所差异。

```
etf read_efuse CHIPID
```

### 5.3.2 功率校准

```
etf read_efuse POUT_CAL //读取当前 efuse 中功率校准值
```

```
etf write_efuse POUT_CAL [low] [mid] [high] //设置功率校准值低中高信道参考范围[0x7f : 0x3f] (第 7 位为符号位)
```

#### 1) 读操作

通过“etf read\_efuse POUT\_CAL”命令读取当前 efuse 区域有效的功率校准值，总共三个数据，分别为低中高信道校准值。Low：代表 1 信道（2412MHz）的功率校准值；mid：代表 7 信道（2442MHz）的功率校准值；high：代表 14 信道（2484MHz）的功率校准值。

#### 2) 写操作

通过 etf write\_efuse POUT\_CAL [low] [mid] [high] 命令可以分别设置 1、7、14 信道的功率校准值，这三个值的大小由三个连续的 7 位决定（u32 的低 21 位）。第七位为符号位，低六位的值为有效数字，即可以设置的范围：-63~63。

计算方法： $FEC = (P_{set} - 2 - P_{ins} \times 4) \times 2$ ， $xdB = FEC / 8$ ，x 为正数，说明当前 TX 功率偏小，需要补偿功率；反之 TX 功率偏大，需要削减功率。

示例：

以“etf power\_level 58”为例：TX 时利用仪器测得 1 信道功率为 13dB，即  $P_{ins}=13dB$ ，而设置的功率  $P_{set}=58$ ，代入公式算得 FEC 的值为 8，即 1 信道需要补偿 1dB；在 7 信道利用仪器测得  $P_{ins}=14dB$ ，在 14 信道利用仪器所测得功率  $P_{ins}=15dB$ 。因此我们在三个信道上分别要设置的值为 8、0、-8。换算成 16 进制并且第七位为符号位，则 low、mid、high 的值为分别为 0x08、0x00、0x48。输入命令：“etf write\_efuse POUT\_CAL 0x08 0x00 0x48”进行校正。

### 5.3.3 频偏校准

```
etf read_efuse DCXO_TRIM //获取当前生效频偏值
```

```
etf write_efuse DCXO_TRIM [num] //设置频偏值，参考范围：[0x00 : 0x7f] (无符号位)
```

频偏校准与前面的功率校准有所区别，功率校准是将偏差作为校准值写入 efuse，而频偏校准则是直接将频偏值写入 efuse。频偏校准写入 efuse 一般需要遵循一些步骤：

- 1) 通过命令“etf read\_efuse DCXO\_TRIM”确认当前是否已经写过 efuse 以及有剩余可写次数。
- 2) 通过命令“etf set\_freq\_offset [num]” (num 为 10 进制)调节频偏，找个一个合适的值，此时的频偏值仅临时生效，退出 etf 模式后将恢复成 sdd 文件的设置的值。
- 3) 将上面通过命令调出来合适的频偏值，换算成 16 进制，通过命令“etf write\_efuse DCXO\_TRIM [hex\_num]”写入频偏值。

示例：



```

etf read_efuse DCXO_TRIM //确认当前剩余可写次数大于等于 1

etf set_freq_offset 50 //从频偏值等于 50 开始调
...
etf set_freq_offset 60 //调到 60，发现此时频偏接近 0

etf write_efuse DCXO_TRIM 0x3C //60 对应的十六进制为 0x3C,将其写入 efuse
    
```

### 5.3.4 注意事项

- 1) 每块芯片允许功率校准值以及频偏值写入对应 efuse 区域各三次，可通过读命令来确认当前剩余可写次数以及生效值。
- 2) 当前提供了两种校准方式，分别为读写 `sdd` 和读写 `efuse`。两者都有各自的优缺点，需要用户自己权衡利弊选择合适的方式。

**SDD:**

优点：读写操作方便，且没有写操作次数限制，容错率高；

缺点：刷机或 OTA 在线升级可能会导致 `sdd` 文件被 `reset`。

**efuse:**

优点：只对当前写入的芯片有效，刷机、在线升级不会被 `reset`，保持了 `sdd` 文件的统一性；

缺点：有写入次数限制，具体请根据命令提示（读/写对应区域都有剩余次数提示）。容错率低，一旦使用写 `efuse` 的方式，`fw` 将优先使用 `efuse` 的值进行校准，导致 `sdd` 文件中的频偏值不再生效，频偏将严格按照烧写在 `efuse` 中的频偏值来设置；若写了功率校准区域则低中高信道会产生固定功率偏差。

- 3) 校准后需要重启机器。

## 5.4 TX 测试



注意

1. 在进行 TX 测试之前，请确保已经输入 `etf connect`、`etf enable_phy` 命令。
2. 示例是一次完整的流程，如果在同一次测试过程中需要切换信道、速率等参数只需 `tx_stop` 后操作相关的参数设置命令，无需 `etf disconnect`、`etf connect` 和 `etf enable_phy` 操作

TX 测试基本格式如下（可通过 `etf help tx` 获取 TX 的帮助信息）。其中 `continuous` 为 1 时表示连续发送，为 0 时表示帧数发送，默认为 1；当 `continuous` 为 0 时，`num` 表示要发送的帧数；`length` 表示发送帧的长度，默认为 500；`-g` 参数适用于 11n 模式，`guard_interval` 值为 0 时代表 Long GI，为 1 时代表 Short GI，默认为 0。

```

etf tx -c [continuous] -n [num] -l [length] -g [guard_interval]
etf tx_stop
    
```

单载波发送基本格式如下。其中 amplitude 表示单载波幅度，默认为 0dbm；freq 为频偏，默认为 5MHz；mode 为载波模式，默认为 Single Tone Quad。

```
etf tone -a [amplitude] -f [freq] -m [mode]
etf tone_stop
```



**注意**  
如果在配置速率时选择了 GreenField 模式，则 guard\_interval 只能配置为 Long GI。

### 5.4.1 TX 测试示例 1

在 1 信道，使用 11n Mixed 模式 MCS7 LongGI 速率、帧长为 4095 进行连续发送

```
etf connect
etf enable_phy
etf channel 2GHZ 1
etf rate -m 5 -r 54
etf tx -c 1 -l 4095 -g 0 //11N 模式下，默认参数-l = 4095 -c=1 -g=0，即这里与"etf tx"是等效的
etf tx_stop
etf disconnect
```

### 5.4.2 TX 测试示例 2

在 6 信道，使用 11g 模式 54Mbps 速率、功率等级为 50 进行 1000 帧发送。

```
etf connect
etf enable_phy
etf channel 2GHZ 6
etf rate -m 2 -r 54
etf power_level 50
etf tx -c 0 -n 1000 //-c 0 表示根据-n 参数决定发完指定数量帧数后则停止
etf disconnect
```

固定帧数发送不需要 tx\_stop。

### 5.4.3 TX 测试示例 3

在 11 信道，进行单载波连续发送。

```
etf connect
etf enable_phy
etf channel 2GHZ 11
etf tx
etf tone
etf tone_stop
etf tx_stop
etf disconnect
```

#### 5.4.4 TX 测试示例 4

40M 模式下，在 5 信道 UPPER，使用 11n Mixed 模式 MCS7 LongGI 速率、帧长为 4095 进行连续发送。该 40M 带宽信道的中心频率为 2422MHz，有关 40M 带宽说明参考 5.1.10 和 5.1.11 小节。

```

etf connect
etf enable_phy
etf bandwidth 40M
etf subchannel UPPER

etf channel 2GHZ 5 //设置完带宽、子信道后，需要设置主信道以配置中心频率。
etf rate -m 5 -r 65
etf tx -l 4095 -g 0
etf tx_stop
etf disconnect
    
```

#### 5.4.5 TX 测试示例 5

在 7 信道，使用 11n Mixed 模式 MCS7 LongGI 速率，高占空比发包

```

etf connect
etf enable_phy
etf channel 2GHZ 1
etf rate -m 5 -r 65

etf tx -f 1 //由示例 1 知道，对应模式、速率参数有默认值，这里不用刻意设置
etf tx_stop
etf disconnect
    
```

#### 窍门

其他模式或速率提高占空比发包只需要在 etf tx 命令加“-f 1”参数即可。

### 5.5 RX 测试

RX 测试基本格式如下。RX 测试无参数，停止后会返回统计数据：

```

etf rx //启动 RX
etf rx_stop //停止 RX
    
```



#### 注意

在进行 RX 测试之前，请确保已经输入 etf connect、etf enable\_phy 命令。

### 5.5.1 RX 测试示例 1

在 7 信道进行 rx

```
etf connect
etf enable_phy
etf channel 2GHZ 7
etf rx
etf rx_stop
etf disconnect
```

### 5.5.2 RX 测试示例 2

40M 模式下，在 5 信道 LOWER 进行连续 RX。该 40M 带宽信道的中心频率为 2422MHz，有关 40M 带宽说明参考 5.1.10 和 5.1.11 小节。

```
etf connect
etf enable_phy
etf bandwidth 40M
etf subchannel LOWER
etf channel 2GHZ 5 //设置完带宽子信道后，需要设置主信道，以配置中心频率。
etf rx
etf rx_stop
etf disconnect
```

执行“etf rx\_stop”后返回数据如下：

```
Rx mode is:      DSSS_LONG_PREAMBLE
CFO:            -5.371094
SNR:            7.250000
RSSI:           -83.000000
EVM:            0.000000
RCPI:           -82.500000

Total:          194
AbortError:    29
CRCError:      26

Sending CMD OK!
```

## 6 常见问题与解决

### 1) RfTester 进行“WIFI-ON”操作后，提示“Set ON Failed, Can't connect to Driver!”

机器在启动或重启后，在测试之前没有关闭 SELinux 防火墙。请在测试前输入“adb shell setenforce 0”命令。

xradio\_core、xradio\_mac 模块尚未加载，通过 lsmod 命令查看是否已经加载这两个模块。如果没有加载，请依次输入一下命令(Tina 平台)：

```
insmod /lib/modules/*/xradio_mac.ko //根据模块实际目录进行加载,参考 3.2.2 小节
insmod /lib/modules/*/xradio_core.ko
```

### 2) RfTester 点击“WIFI-ON”按钮后，提示“Set ON Timeout, Please make sure wifi is off!”

没有卸载 xradio\_wlan.ko 模块 (Tina 平台)，单 ko 形式没有卸载再重新加载驱动 etf\_enable=1 (Android 平台)，请参考 3.3.2 节。

### 3) ETF\_API\_FW\_NOT\_RUNNING!

没有执行“etf connect”，或 connect 失败。

### 4) 能读 sdd 数据但无法执行写操作

没有权限，需要先执行“adb root; adb remount”获取权限后再进行写操作。

### 5) 无法高占空比发包或者发包功率异常

确认 BT/BLE 已完全关闭：

确保相关进程已经被 kill 掉：ps 命令查看相关是否存在 bt 相关进程，若有请用“killall -9 bt\_XXX” (bt\_XXX 为进程名称，根据实际情况来 kill 掉进程)

若还有问题则执行命令：`echo 0 > /sys/class/rfkill/rfkill0/state`

### 6) ETF 命令行工具无法使用

表 6-1 etf 无法使用可能原因与解决方法：

可能原因	措施
驱动尚未加载	通过 lsmod 命令查看是否存在 xradio_mac 与 xradio_core 模块。(单 ko 则先卸载，再以 etf_enable=1 的后缀加载驱动)
驱动不支持 ETF 模式	通过 ps -A   grep etf 查看是否存在 xradio_etf 线程，如不支持请配置重新编译
设备不能正常工作	WLAN 是否能正常工作，如不能请参考 porting guide 进行排查
缺少 etf 测试固件	查看 etc/firmware 目录是否存在 etf 相关 bin 文件，见“2.3 注意事项”
进行测试之前没有关闭 SELinux 防火墙	运行“RFTTest 工具\WLAN\init_test.bat”脚本临时关闭 SELinux 防火墙，见“2.3 注意事项”

可能原因	措施
没有权限	内核 log 中存在 netlink_socket permissive=0 的打印,请在 root 权限下执行 etf cli
版本问题	升级最新驱动、固件

### 7) ETF 测试无法收发帧

表 6-2 etf 无法收发帧可能原因与解决方法:

可能原因	措施
PHY 没有使能	请在测试之前执行 etf enable_phy
模式和信道配置错误	如 APK 能正常使用,请参考 ETF CLI 使用说明进行正确配置
ETF 命令使用错误	如 APK 能正常使用,请参考 ETF CLI 使用说明进行正确配置
SDD 文件时钟配置错误	请使用 sddedit.exe 工具确认或修改 sdd 文件中的时钟配置,sdd 文件时钟要与硬件保持一致。
芯片或硬件问题	更换模组或者机器进行测试

# 附录 A： ETF CLI 版本说明

ETF CLI 工具发布版本较多，大多数更新都是新增某些功能，依赖驱动和固件的支持，建议使用 SDK 最新驱动和 firmware。以下表格给出相应的依赖关系，注意红色加粗部分。

表 A-1 ETF CLI 版本更新内容以及依赖关系

版本号	备注
1.3.6	1. 修复了单载波无法再 1 信道发包的问题，无驱动、fw 依赖
1.3.7	1. 添加读写频偏接口，可以临时修改频偏，无驱动、fw 依赖
1.3.9 (xr829)	1. 支持 20M 带宽高占空比发包 依赖驱动 XR_V02.16.91_HT40_01.35 <b>注意：使用 XR_V02.16.91_HT40_01.35 至***HT40_01.40 版本驱动</b>
1.4.1 (xr829)	1. 支持 4M 带宽高占空比发包 2. 支持读写 efuse (功率校准、频偏校准) 依赖驱动 XR_V02.16.91_HT40_01.35 新增功能依赖固件 ETF_FW_A09.01.0105 <b>注意：使用 XR_V02.16.91_HT40_01.35 至***HT40_01.40 版本驱动，使用 ETF_FW_A09.01.0105 版本 ETF 固件</b>
2.0.6	1. 适配 XR829、XR819 和 XR819S 模组 2. 支持 sdd 读写 (写操作需要 remount) 3. 优化功率配置 依赖 XR_V02.16.95_HT40_01.41 版本驱动 (当前 adapter 版本为 <b>1.1.2</b> ，后续将以此版本作为参考，版本不支持会有相应的提示) <b>注意：</b> <b>XR829 (Linux-4.9)：使用 XR_V02.16.95_HT40_01.41 及以上版本驱动 (adapter_ver:1.1.2)，使用 ETF_FW_A09.01.0105 版本 ETF 固件</b> <b>XR819/XR819S：使用 XR_V02.25.80 版本驱动 (adapter_ver:1.1.2)</b>

## 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



HRAD TECH、芯之联（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。