



FCS851U&FCS950U Linux

蓝牙用户指导

短距离模块系列

版本: 1.0

日期: 2024-05-22

状态: 受控文件



上海移远通信技术股份有限公司（以下简称“移远通信”）始终以为客户提供最及时、最全面的服务为宗旨。如需任何帮助，请随时联系我司上海总部，联系方式如下：

上海移远通信技术股份有限公司
上海市闵行区田林路 1016 号科技绿洲 3 期（B 区）5 号楼 邮编：200233
电话：+86 21 5108 6236 邮箱：info@quectel.com

或联系我司当地办事处，详情请登录：<http://www.quectel.com/cn/support/sales.htm>。

如需技术支持或反馈我司技术文档中的问题，请随时登陆网址：
<http://www.quectel.com/cn/support/technical.htm> 或发送邮件至：support@quectel.com。

前言

移远通信提供该文档内容以支持客户的产品设计。客户须按照文档中提供的规范、参数来设计产品。同时，您理解并同意，移远通信提供的参考设计仅作为示例。您同意在设计您目标产品时使用您独立的分析、评估和判断。在使用本文档所指导的任何硬软件或服务之前，请仔细阅读本声明。您在此承认并同意，尽管移远通信采取了商业范围内的合理努力来提供尽可能好的体验，但本文档和其所涉及服务是在“可用”基础上提供给您的。移远通信可在未事先通知的情况下，自行决定随时增加、修改或重述本文档。

使用和披露限制

许可协议

除非移远通信特别授权，否则我司所提供硬软件、材料和文档的接收方须对接收的内容保密，不得将其用于除本项目的实施与开展以外的任何其他目的。

版权声明

移远通信产品和本协议项下的第三方产品可能包含受移远通信或第三方材料、硬软件和文档版权保护的相关资料。除非事先得到书面同意，否则您不得获取、使用、向第三方披露我司所提供的文档和信息，或对此类受版权保护的资料进行复制、转载、抄袭、出版、展示、翻译、分发、合并、修改，或创造其衍生作品。移远通信或第三方对受版权保护的资料拥有专有权，不授予或转让任何专利、版权、商标或服务商标权的许可。为避免歧义，除了正常的非独家、免版税的产品使用许可，任何形式的购买都不可被视为授予许可。对于任何违反保密义务、未经授权使用或以其他非法形式恶意使用所述文档和信息的违法侵权行为，移远通信有权追究法律责任。

商标

除另行规定，本文档中的任何内容均不授予在广告、宣传或其他方面使用移远通信或第三方的任何商标、商号及名称，或其缩略语，或其仿冒品的权利。

第三方权利

您理解本文档可能涉及一个或多个属于第三方的硬软件和文档（“第三方材料”）。您对此类第三方材料的使用应受本文档的所有限制和义务约束。

移远通信针对第三方材料不做任何明示或暗示的保证或陈述，包括但不限于任何暗示或法定的适销性或特定用途的适用性、平静受益权、系统集成、信息准确性以及与许可技术或被许可人使用许可技术相关的不侵犯任何第三方知识产权的保证。本协议中的任何内容都不构成移远通信对任何移远通信产品或任何其他硬软件、设备、工具、信息或产品的开发、增强、修改、分销、营销、销售、提供销售或以其他方式维持生产的陈述或保证。此外，移远通信免除因交易过程、使用或贸易而产生的任何和所有保证。

隐私声明

为实现移远通信产品功能，特定设备数据将会上传至移远通信或第三方服务器（包括运营商、芯片供应商或您指定的服务器）。移远通信严格遵守相关法律法规，仅为实现产品功能之目的或在适用法律允许的情况下保留、使用、披露或以其他方式处理相关数据。当您与第三方进行数据交互前，请自行了解其隐私保护和数据安全政策。

免责声明

- 1) 移远通信不承担任何因未能遵守有关操作或设计规范而造成损害的责任。
- 2) 移远通信不承担因本文档中的任何因不准确、遗漏、或使用本文档中的信息而产生的任何责任。
- 3) 移远通信尽力确保开发中功能的完整性、准确性、及时性，但不排除上述功能错误或遗漏的可能。除非另有协议规定，否则移远通信对开发中功能的使用不做任何暗示或法定的保证。在适用法律允许的最大范围内，移远通信不对任何因使用开发中功能而遭受的损害承担责任，无论此类损害是否可以预见。
- 4) 移远通信对第三方网站及第三方资源的信息、内容、广告、商业报价、产品、服务和材料的可访问性、安全性、准确性、可用性、合法性和完整性不承担任何法律责任。

版权所有 ©上海移远通信技术股份有限公司 2024，保留一切权利。

Copyright © Quectel Wireless Solutions Co., Ltd. 2024.

文档历史

修订记录

版本	日期	作者	变更表述
-	2023-10-27	Noah YANG	文档创建
1.0	2024-05-22	Noah YANG	受控版本

目录

文档历史	3
目录	4
表格索引	6
图片索引	7
1 引言	8
2 环境准备	9
2.1. 硬件环境	9
2.2. 软件环境	11
2.3. 蓝牙驱动包	11
3 集成与编译	13
3.1. 集成代码	13
3.1.1. 获取蓝牙驱动源码	13
3.1.2. 拷贝蓝牙固件	13
3.2. 编译代码	14
3.3. 导入相关文件	15
4 功能验证	17
4.1. 验证蓝牙基本功能	17
4.1.1. 蓝牙模块上电	17
4.1.2. 加载蓝牙内核驱动	17
4.1.3. 初始化蓝牙模块	18
4.1.4. 加载蓝牙守护进程	18
4.2. 验证音频播放 (Sink)	19
4.2.1. 加载 PulseAudio 音频服务	19
4.2.2. 搜索&连接设备	19
4.2.3. 播放音频	19
4.3. 验证蓝牙音频 (Source)	20
4.3.1. 加载 PulseAudio 音频服务	20
4.3.2. 搜索&连接设备	20
4.3.3. 播放音频	21
4.4. 验证蓝牙低功耗协议 (Server)	22
4.4.1. 创建 Server 并广播	22
4.4.2. 读写测试	24
4.5. 验证蓝牙低功耗协议 (Client)	25
4.5.1. 开启手机端 Server 并广播	25
4.5.2. 搜索&连接设备	25
4.5.3. 读写测试	25
4.5.3.1. 读测试	25
4.5.3.2. 写测试	26
5 工具使用	错误!未定义书签。

6 附录 术语缩写.....	28
----------------	----

表格索引

表 1: 硬件环境	9
表 2: 软件环境	11
表 3: 蓝牙文件	11
表 4: 蓝牙驱动目录结构	12
表 5: 获取蓝牙驱动源码的命令	13
表 6: 术语缩写	28

图片索引

图 1: RK3568-WF EVB 正面图	10
图 2: RK3568-WF EVB 背面图	10

1 引言

移远通信 FCS851U 和 FCS950U 模块支持在 Linux 平台上实现蓝牙功能。本文档以安装 FCS851U 和 FCS950U 模块的移远通信 RK3568-WF EVB 为例，介绍如何在 Linux 平台上启用 FCS851U 和 FCS950U 模块的蓝牙基本功能并进行功能验证。

2 环境准备

2.1. 硬件环境

表 1：硬件环境

硬件名称	数量
移远通信 RK3568-WF EVB	1
移远通信蓝牙模块	1
天线	1
USB Type-C 线	1
电源线	1

备注

1. 开发板型号仅供参考。如使用其他型号开发板，请向对应厂商咨询详细信息。
2. 请用户根据开发板实际情况确定与模块的各接口/GPIO 的连接是否正确（SDIO、CTS/RTS/TX/RX、PCM、BT_EN、Wakeup 引脚），RK3568-WF EVB 通过 M.2 接口与模块连接。
3. 必须确保蓝牙天线连接正常（不同模块型号对应的天线接口位置不同，具体请参考相应模块的 DataSheets）。

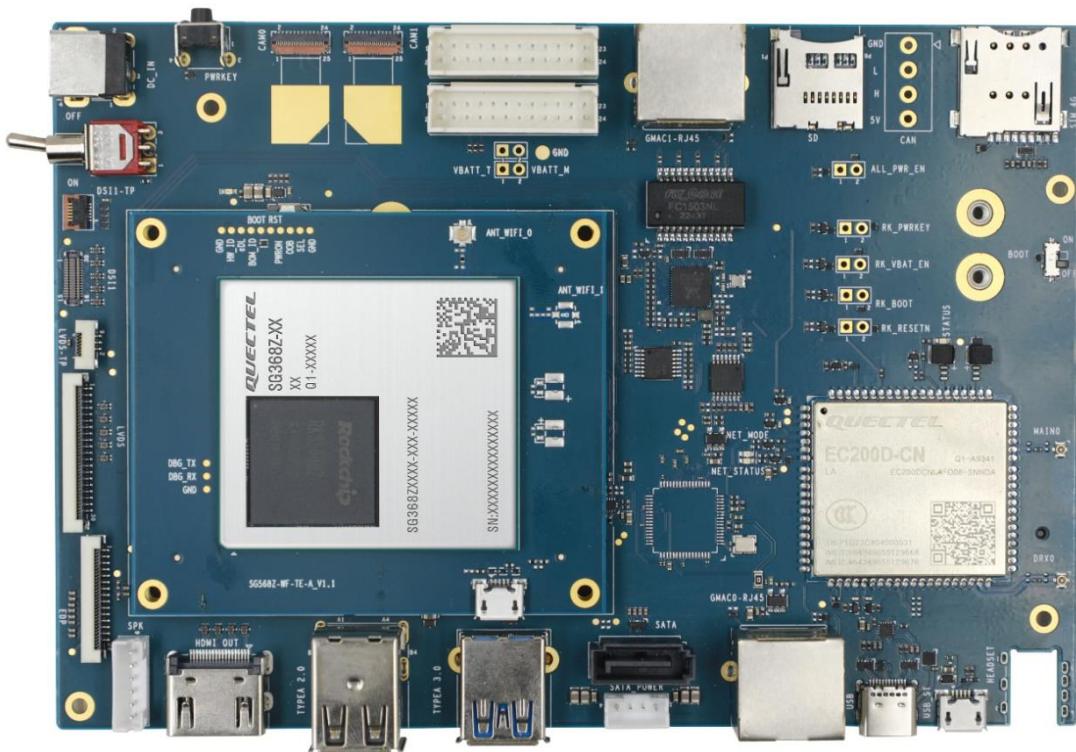


图 1：RK3568-WF EVB 正面图

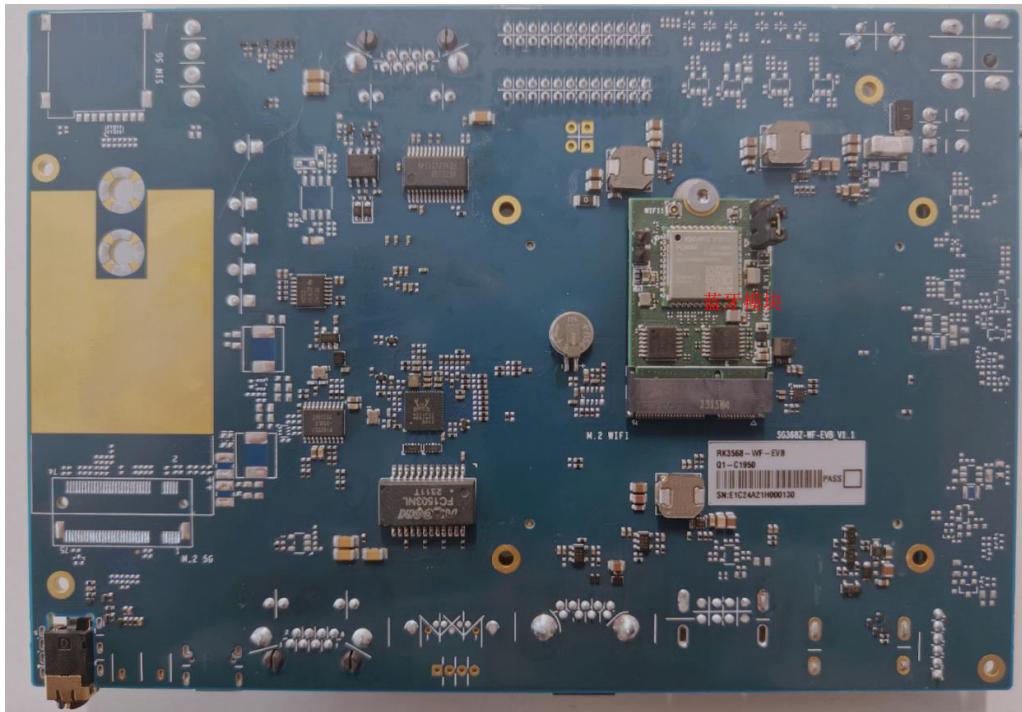


图 2：RK3568-WF EVB 背面图

2.2. 软件环境

本文档以移远通信 RK3568-WF EVB 的 Linux BSP 作为软件环境。用户应自行准备与实际使用的开发板配套的 Linux BSP，且需满足以下要求：

表 2：软件环境

类型	描述
代码环境	Linux 4.19
驱动包	移远通信蓝牙驱动包
编译环境	Ubuntu 20.04

备注

1. Linux BSP 必须支持 UART 硬件流控，且波特率应该至少支持 115200 bps 和 1500000 bps。且必须支持 BT_EN GPIO 的控制功能，能够通过某种方式控制模块蓝牙的上下电。
2. FCS851U 和 FCS950U 模块使用 SDIO 驱动，因此 Linux BSP 必须支持 SDIO。

2.3. 蓝牙驱动包

移远通信蓝牙驱动包包含以下信息：

表 3：蓝牙文件

文件夹	描述
BT/Linux/Driver	包含原厂提供的 Linux 驱动、工具代码。
BT/Linux/FW	<i>bt_*.ini</i> 配置文件。
BT/Android	Android 平台 Bring-up 软件包（本文档不涉及）
Doc/EN	移远通信相关说明文档（包括本文档）英文版本
Doc/CN	移远通信相关说明文档（包括本文档）中文版本
WiFi	Wi-Fi Bring-up 相关驱动包（本文档不涉及）

表 4: 蓝牙驱动目录结构

一级目录	二级目录	描述
kernel	<i>unisocbt</i>	<i>prdbt_tty.ko</i> 内核驱动代码 (FCS950U)
	<i>tty-sdio</i>	<i>prdbt_tty.ko</i> 内核驱动代码 (FCS851U)
	<i>unisocwifi</i>	Wi-Fi 内核驱动代码 (此文档不涉及)
	<i>unisocwcn</i>	WCN 内核驱动代码 (此文档只使用该代码, 具体适配信息请联系移远通信技术支持)
userspace	<i>hciattach</i>	<i>hciattach</i> 代码
	<i>hciconfig</i>	<i>hciconfig</i> 代码
	<i>libbt</i>	<i>libbt-sprd-vendor.so</i> 库
	<i>btsuite</i>	<i>libbt-sprd_suite.so</i> 库
	<i>bdtunisoc</i>	<i>bdt_unisoc</i> 测试工具

3 集成与编译

3.1. 集成代码

3.1.1. 获取蓝牙驱动源码

在 Ubuntu 环境下执行如下命令获取蓝牙驱动源码。

```
mkdir Quectel
cd Quectel
git clone <特定的仓库名称>
```

表 5：获取蓝牙驱动源码的命令

模块	命令
FCS851U	git clone https://git-master.quectel.com/wifi.bt/fcs851u
FCS950U	git clone https://git-master.quectel.com/wifi.bt/fcs950u

3.1.2. 拷贝蓝牙固件

以 FCS851U 模块为例，将模块 `fw` 目录下的蓝牙配置文件 `bt*.ini` 拷贝至 `rootfs` 中的 `/etc/bluetooth/uwe5621` 目录下。该步骤为集成的核心步骤。

备注

1. 若使用 FCS950U 模块，需将 `bt*.ini` 拷贝至 `/etc/bluetooth/uwe5622` 目录。
2. 以上操作仅为示例，由于用户开发板环境可能不一致，本文档中的操作可能不完全匹配用户环境，请用户自行修改部分操作细节。

3.2. 编译代码

- 在 *bluez* 目录下执行以下命令编译 BlueZ（若系统自带 BlueZ，则无需编译）。

```
./bootstrap-configure --disable-android --prefix=<用户自定义的 BlueZ 安装路径>
make
make install
```

- 执行以下命令编译 *tty-sdio* 目录下的蓝牙驱动源码，编译完成后在该目录下生成 *sprd_bt_tty.ko* 文件。

```
# cd tty-sdio
# make
```

备注

若使用 FCS950U 模块，上述命令中的 **tty-sdio** 需改为 **unisocbt**。

- 执行以下命令编译 *hciattach* 目录下的蓝牙驱动源码，编译完成后在该目录下生成 *hciattach_sprd* 文件。

```
# cd hciattach
# make
```

- 执行以下命令编译 *hciconfig* 目录下的蓝牙驱动源码，编译完成后在该目录下生成 *hciconfig_sprd* 文件。

```
# cd hciconfig
# make
```

- 执行以下命令编译 *libbt* 目录下的蓝牙驱动源码，编译完成后在该目录下生成 *libbt-sprd-vendor.so* 文件。

```
# cd libbt
# make all
```

- 执行以下命令编译 *btsuite* 目录下的蓝牙驱动源码，编译完成后在该目录下生成 *libbt-sprd-suite.so* 文件。

```
# cd btsuite
# make
```

- 执行以下命令编译 *bdtunisoc* 目录下的蓝牙驱动源码，编译完成后在该目录下生成 *bdt_unisoc* 文件。

```
# cd bdt_unisoc  
# make
```

3.3. 导入相关文件

- 执行以下命令将生成的 *sprdbt_tty.ko* 文件拷贝至 */usr/lib/modules/fcs851u/* 目录下。

```
# adb root  
# adb remount  
# adb shell mount -o remount rw /  
# adb push sprdbt_tty.ko /usr/lib/modules/fcs851u/  
# adb shell ls /usr/lib/modules/fcs851u/sprdbt_tty.ko
```

备注

若使用 FCS950U 模块，上述命令和目录中的“fcs851u”需为“fcs950u”。

- 执行以下命令将生成的 *hciattach_sprd* 文件拷贝至 */usr/bin/* 目录下。

```
# adb root  
# adb remount  
# adb shell mount -o remount rw /  
# adb push hciattach_sprd /usr/bin/  
# adb shell ls /usr/bin/hciattach_sprd
```

- 执行以下命令将生成的 *hciconfig_sprd* 文件拷贝至 */usr/bin/* 目录下。

```
# adb root  
# adb remount  
# adb shell mount -o remount rw /  
# adb push hciconfig_sprd /usr/bin/  
# adb shell ls /usr/bin/hciconfig_sprd
```

- 执行以下命令将生成的 *libbt-sprd-vendor.so* 文件拷贝至 */usr/lib/* 目录下。

```
# adb root  
# adb remount  
# adb shell mount -o remount rw /  
# adb push libbt-sprd-vendor.so /usr/lib/
```

```
# adb shell ls /usr/lib/libbt-sprd-vendor.so
```

- 执行以下命令将生成的 *libbt-sprd_suite.so* 文件拷贝至 */usr/lib/* 目录下。

```
# adb root
# adb remount
# adb shell mount -o remount rw /
# adb push libbt-sprd_suite.so /usr/lib/
# adb shell ls /usr/lib/libbt-sprd_suite.so
```

- 执行以下命令将生成的 *bdt_unisoc* 文件拷贝至 */usr/bin/* 目录下。

```
# adb root
# adb remount
# adb shell mount -o remount rw /
# adb push bdt_unisoc /usr/bin/
# adb shell ls /usr/bin/bdt_unisoc
```

备注

1. 不同的编译环境需要提前配置不同的交叉编译工具链。
2. 不同的开发环境依赖的组件可能不同，如果遇到编译错误，可联系移远通信技术支持。
3. 不同的功能需要依赖的配置选项可能不同，用户可根据需求自行开启或者关闭特定的功能。
4. 模块蓝牙调试需要在 Wi-Fi 调试完成之后进行，详细的调试步骤请联系移远通信技术支持。

4 功能验证

本章节以安装移远通信蓝牙模块的 RK3568-WF EVB 为例，介绍如何在 Linux 平台上验证模块蓝牙基本功能。

备注

其他 Linux 平台的功能验证流程类似。如需协助，请联系移远通信技术支持。

4.1. 验证蓝牙基本功能

4.1.1. 蓝牙模块上电

执行如下命令将蓝牙模块的 BT_EN 引脚上电。

```
echo 0 > /sys/class/rfkill/rfkill0/state
echo 1 > /sys/class/rfkill/rfkill0/state
```

备注

RK3568-WF EVB 需执行 **rfkill** 命令使能模块的蓝牙功能，用户可根据开发板自有方式操作 BT_EN 引脚。

4.1.2. 加载蓝牙内核驱动

执行如下命令加载蓝牙内核驱动。

- FCS851U 模块：

```
insmod usr/lib/modules/fcs851u/uwe5621_bsp_sdio.ko
insmod usr/lib/modules/fcs851u/sprdbt_tty.ko
```

- FCS950U 模块：

```
insmod usr/lib/modules/fcs950u/uwe5622_bsp_sdio.ko
insmod usr/lib/modules/fcs950u/sprdbt_tty.ko
```

4.1.3. 初始化蓝牙模块

执行如下命令初始化蓝牙模块。

```
chmod 777 /usr/bin/hciattach_sprd  
chmod 777 /usr/bin/hciconfig_sprd  
hciattach_sprd /dev/ttyBT0 sprd &  
hciconfig_sprd -a
```

```
root@rockchip:/# /usr/bin/unisoc/hciconfig_sprd -a
hci0:  Type: Primary  Bus: UART
        BD Address: 40:45:DA:7E:60:90  ACL MTU: 1021:8  SCO MTU: 240:3
        DOWN
        RX bytes:669 acl:0 sco:0 events:36 errors:0
        TX bytes:430 acl:0 sco:0 commands:36 errors:0
        Features: 0xbf 0xfe 0x8d 0xfe 0xdb 0x3d 0x7b 0xc7
        Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
        Link policy: RSWITCH SNIFF
        Link mode: SLAVE ACCEPT
```

4.1.4. 加载蓝牙守护进程

执行如下命令启动蓝牙守护进程。

```
/usr/libexec/bluetooth/bluetoothd -n -d &
```

备注

以上操作的某些步骤可能需要 **sudo** 权限, 请根据实际情况进行调整。

4.2. 验证音频播放 (Sink)

4.2.1. 加载 PulseAudio 音频服务

执行如下命令启动 PulseAudio 音频服务。

```
pulseaudio &
```

4.2.2. 搜索&连接设备

执行如下命令搜索并连接设备。

```
bluetoothctl
[bluetooth]# scan on
[bluetooth]# scan off
[bluetooth]# devices
[bluetooth]# connect xx:xx:xx:xx:xx:xx
```

备注

xx:xx:xx:xx:xx:xx 表示目标设备的蓝牙地址。

4.2.3. 播放音频

步骤一: 在手机端查看是否显示蓝牙连接图标以确认 A2DP 是否连接成功;

步骤二: 执行以下命令设置 PulseAudio 音频服务的音频通道标号:

```
pactl list sinks
pactl set-default-sink 2
```

备注

上述命令中的 **sink #** 需根据实际情况进行设置。

Sink #2

```

State: SUSPENDED
Name: alsa_output.0.HiFi_hw_rockchiprk809co_sink
Description: Built-in Audio Speaker Playback + Headphones Playback
Driver: module-alsa-card.c
Sample Specification: s16le 2ch 44100Hz
Channel Map: front-left,front-right
Owner Module: 8
Mute: no
Volume: front-left: 65536 / 100% / 0.00 dB, front-right: 65536 / 100% / 0.00 dB
        balance 0.00
Base Volume: 65536 / 100% / 0.00 dB
Monitor Source: alsa_output.0.HiFi_hw_rockchiprk809co_sink.monitor
Latency: 0 usec, configured 0 usec
Flags: HARDWARE DECIBEL_VOLUME LATENCY

```

步骤三：在手机端播放音频。若从开发板耳机中可以听到蓝牙音频，并且可以在系统终端看到音频相关信息，则表示音频播放功能正常。

```

[Redmi K40]# bluetoothd[1356]: profiles/audio/player.c:media_player_set_playlist_item() 0
bluetoothd[1356]: profiles/audio/player.c:media_player_set_metadata() Title: Bye Bye Bye
bluetoothd[1356]: profiles/audio/player.c:media_player_set_metadata() Artist: Lovestoned
bluetoothd[1356]: profiles/audio/player.c:media_player_set_metadata() Album: Rising Love
bluetoothd[1356]: profiles/audio/player.c:media_player_set_metadata() TrackNumber: 3
bluetoothd[1356]: profiles/audio/player.c:media_player_set_metadata() NumberOfTracks: 397
bluetoothd[1356]: profiles/audio/player.c:media_player_set_metadata() Genre:
bluetoothd[1356]: profiles/audio/player.c:media_player_set_metadata() Duration: 151510
[CHG] Player /org/bluez/hci0/dev_74_15_75_DC_F2_3C/player0 Title: Bye Bye Bye
[CHG] Player /org/bluez/hci0/dev_74_15_75_DC_F2_3C/player0 TrackNumber: 0x00000003 (3)
[CHG] Player /org/bluez/hci0/dev_74_15_75_DC_F2_3C/player0 NumberOfTracks: 0x0000018d (397)
[CHG] Player /org/bluez/hci0/dev_74_15_75_DC_F2_3C/player0 Duration: 0x00024fd6 (151510)
[CHG] Player /org/bluez/hci0/dev_74_15_75_DC_F2_3C/player0 Album: Rising Love
[CHG] Player /org/bluez/hci0/dev_74_15_75_DC_F2_3C/player0 Artist: Lovestoned
[CHG] Player /org/bluez/hci0/dev_74_15_75_DC_F2_3C/player0 Genre:

```

备注

针对部分开发板，用户需自行设置 mixer 使声音通过耳机或者扬声器发出。RK3568-WF EVB 的 mixer 设置命令示例如下：**amixer -c 0 cset numid=1 3**。

4.3. 验证蓝牙音频（Source）

4.3.1. 加载 PulseAudio 音频服务

详见第 4.2.1 章。

4.3.2. 搜索&连接设备

详见第 4.2.2 章。

4.3.3. 播放音频

步骤一：在手机端查看是否显示蓝牙连接图标以确认 A2DP 是否连接成功。

步骤二：执行如下命令设置 PulseAudio 音频服务的音频通道标号：

```
pactl list sinks  
pactl set-default-sink 3
```

备注

上述命令中的 **sink #** 需根据实际情况进行设置。

```
Sink #3  
  State: SUSPENDED  
  Name: bluez_sink.CD_45_71_E8_29_D4.a2dp_sink  
  Description: BT-SPEAKER  
  Driver: module-bluetooth-device.c  
  Sample Specification: s16le 2ch 44100Hz  
  Channel Map: front-left,front-right  
  Owner Module: 24  
  Mute: no  
  Volume: front-left: 65536 / 100% / 0.00 dB, front-right: 65536 / 100% / 0.00 dB  
        balance 0.00  
  Base Volume: 65536 / 100% / 0.00 dB  
  Monitor Source: bluez_sink.CD_45_71_E8_29_D4.a2dp_sink.monitor  
  Latency: 0 usec, configured 0 usec  
  Flags: HARDWARE DECIBEL_VOLUME LATENCY
```

步骤三：在开发板端播放音频。若可以从蓝牙耳机或蓝牙音箱中听到音频，则表示音频播放功能正常。

4.4. 验证蓝牙低功耗协议（Server）

4.4.1. 创建 Server 并广播

步骤一：执行 **bluetoothctl** 开启 bluetoothctl。

步骤二：执行**[bluetooth]# menu gatt** 进入 *gatt* 子目录。

```
[bluetooth]# menu gatt
Menu gatt:
Available commands:
-----
list-attributes [dev/local]           List attributes
select-attribute <attribute/UUID>    Select attribute
attribute-info [attribute/UUID]        Select attribute
read [offset]                         Read attribute value
write <data=xx xx ...> [offset] [type] Write attribute value
acquire-write                         Acquire Write file descriptor
release-write                         Release Write file descriptor
acquire-notify                         Acquire Notify file descriptor
release-notify                         Release Notify file descriptor
notify <on/off>                      Notify attribute value
clone [dev/attribute/UUID]            Clone a device or attribute
register-application [UUID ...]       Register profile to connect
unregister-application                Unregister profile
register-service <UUID> [handle]      Register application service.
unregister-service <UUID/object>       Unregister application service
register-includes <UUID> [handle]      Register as Included service in.
unregister-includes <Service-UUID><Inc-UUID> Unregister Included service.
register-characteristic <UUID> <Flags=read,write,notify...> [handle] Register application characteristic
unregister-characteristic <UUID/object> Unregister application characteristic
register-descriptor <UUID> <Flags=read,write...> [handle] Register application descriptor
unregister-descriptor <UUID/object>    Unregister application descriptor
back                                Return to main menu
version                             Display version
quit                                Quit program
exit                                Quit program
help                                Display help about this program
export                             Print environment variables
```

步骤三：执行**[bluetooth]# register-service e2d36f99-8909-4136-9a49-d825508b297b** 注册服务 e2d36f99-8909-4136-9a49-d825508b297b。

```
[bluetooth]# register-service e2d36f99-8909-4136-9a49-d825508b297b
[NEW] Primary Service (Handle 0x0000)
      /org/bluez/app/service0
      e2d36f99-8909-4136-9a49-d825508b297b
      Vendor specific
[/org/bluez/app/service0] Primary (yes/no): yes
```

步骤四：执行**[bluetooth]# register-characteristic 1234 read** 注册含有读取属性的特征值。

```
[bluetooth]# register-characteristic 1234 Flags=read,write
[NEW] Characteristic (Handle 0x0000)
      /org/bluez/app/service0/chrc0
      1234
      Unknown
[/org/bluez/app/service0/chrc0] Enter value: 0
```

步骤五：执行[bluetooth]# register-characteristic 0x5678 write 注册含有写入属性的特征值。

```
[bluetooth]# register-characteristic 0x5678 write
[NEW] Characteristic (Handle 0x0000)
      /org/bluez/app/service0/chrc1
      0x5678
      Unknown
[/org/bluez/app/service0/chrc1] Enter value: 0
```

步骤六：执行如下命令开启广播。

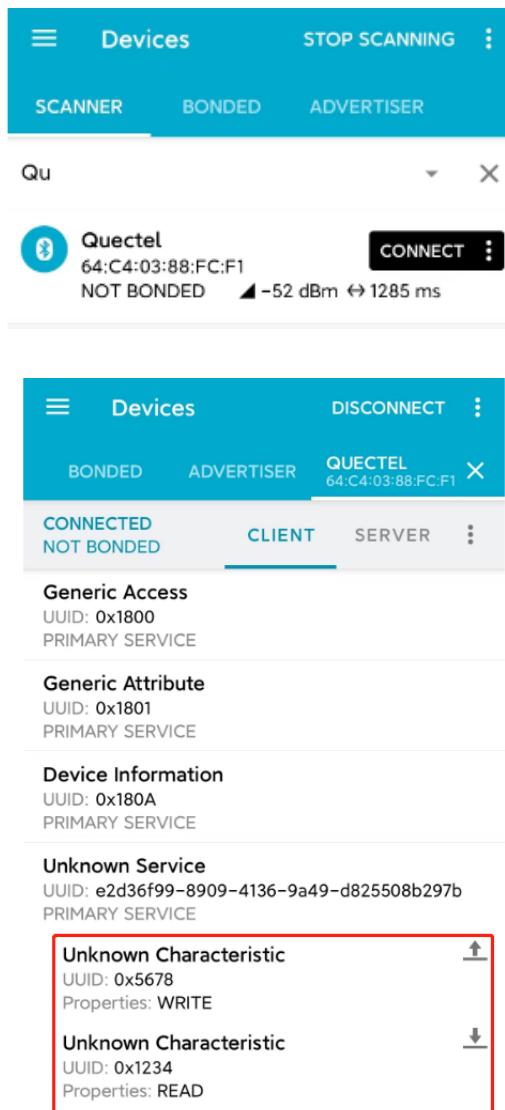
```
[bluetooth]# register-application
[bluetooth]# back
[bluetooth]# menu advertise
[bluetooth]# name Quectel
[bluetooth]# back
[bluetooth]# advertise on
```

备注

关于 nRF Connect 工具的下载和使用，请参考官方网址：
<https://www.nordicsemi.com/Products/Development-tools/nRF-Connect-for-mobile>。

4.4.2. 读写测试

步骤一：打开手机端应用程序 nRF Connect 并进入“SCANNER”界面；点击“CONNECT”，找到创建的两个特征值。

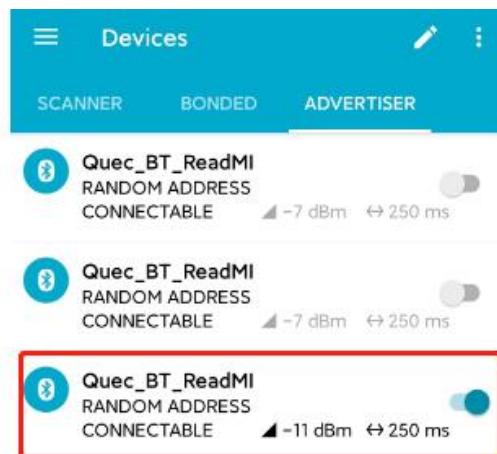


步骤二：点击特征后的箭头标志进行读写操作。

4.5. 验证蓝牙低功耗协议 (Client)

4.5.1. 开启手机端 Server 并广播

打开手机端应用程序 nRF Connect 进入“ADVERTISER”界面，开启 Server 并广播。



4.5.2. 搜索&连接设备

执行如下命令搜索并连接设备。

```
bluetoothctl
[bluetooth]# scan on
[bluetooth]# scan off
[bluetooth]# devices
[bluetooth]# connect xx:xx:xx:xx:xx:xx
[bluetooth]# menu gatt
[bluetooth]# list-attributes
```

4.5.3. 读写测试

4.5.3.1. 读测试

步骤一：执行如下命令选择 UUID 为 fff1 的属性，用于从手机端接收数据：

```
select-attribute 0000fff1-0000-1000-8000-00805f9b34fb #选择一个 Server 的 UUID
```

步骤二：执行如下命令开启通知功能：

```
notify on
```

步骤一： 手机端通过 nRF Connect 向 RK3568-WF EVB 写入数据。写数据时的 UUID 需要和 **select-attribute** 命令中的 UUID 一致。

步骤二： 检查是否能在 BlueZ 运行界面的 log 打印信息中查看到接收到的数据。

4.5.3.2. 写测试

步骤一： 执行如下命令选择 UUID 为 fff2 的属性，用于写入数据到手机：

```
select-attribute 0000fff20000-1000-8000-00805f9b34fb #选择一个 Server 的 UUID
```

步骤二： 执行 **write 1** 写入数据 1（以 1 为例）。

步骤三： 查看手机端 nRF Connect 中 fff2 区域内接收到的数据。如果接收到的数据为 1，则表示写入数据成功。

5 验证测试工具

本章节以移远通信 RK3568-WF EVB 为例，介绍如何使用 **bdt_unisoc** 测试工具在 Linux 平台上启用蓝牙功能，以验证该工具能否正常使用。

步骤一：执行 **adb shell** 进入 shell 命令行窗口。

步骤二：以 FCS950U 模块为例，执行如下命令加载 *uwe5622_bsp_sdio.ko* 和 *sprdbt_tty.ko*。

```
insmod usr/lib/modules/fcs950u/uwe5622_bsp_sdio.ko
insmod usr/lib/modules/fcs950u/sprdbt_tty.ko
```

步骤三：执行 **bdt_unisoc** 运行 **bdt_unisoc** 工具。

步骤四：输入 **enable**，若返回“HAL REQUEST SUCCESS”，则表示 **bdt_unisoc** 测试工具成功启用蓝牙功能，即该工具可正常使用。

```
/ # bdt_unisoc
::::::::::::::::::
:: bdt_unisoc test app starting
INIT BT
>enable
ENABLE BT
HAL REQUEST SUCCESS
>|
```

6 附录 术语缩写

表 6: 术语缩写

缩写	英文全称	中文全称
A2DP	Advanced Audio Distribution Profile	高级音频分发协议
APP	Application	应用
ARM	Advanced RISC Machine	RISC 微处理器
AVRCP	Audio/Video Remote Control Profile	音频/视频远程控制规范
BLE	Bluetooth Low Energy	蓝牙低功耗
BSP	Board Support Package	板级支持包
CTS	Clear To Send	清除发送
EVB	Evaluation Board	评估板
GPIO	General-Purpose Input/Output	通用型输入/输出
HCI	Host Controller Interface	主机控制接口
ID	Mostly refers to Identifier in terms of software	软件中多数指“标识符”
PC	Personal Computer	个人电脑
PCM	Pulse Code Modulation	脉冲编码调制
RTS	Ready To Send/Request to Send	准备发送/请求发送
RX	Receive	接收
SDIO	Secure Digital Input/Output	安全数字输入/输出
TX	Transmit	发送
UART	Universal Asynchronous Receiver/Transmitter	通用异步收发传输器
USB	Universal Serial Bus	通用串行总线