

FC06E&FC2x&FC6xE&FC900E&FGE57xQ Series Linux Bluetooth User Guide

Wi-Fi&Bluetooth Module Series

Version: 1.0

Date: 2023-10-27

Status: Released



At Quectel, our aim is to provide timely and comprehensive services to our customers. If you require any assistance, please contact our headquarters:

Quectel Wireless Solutions Co., Ltd.

Building 5, Shanghai Business Park Phase III (Area B), No.1016 Tianlin Road, Minhang District, Shanghai 200233, China

Tel: +86 21 5108 6236

Email: info@quectel.com

Or our local offices. For more information, please visit:

<http://www.quectel.com/support/sales.htm>.

For technical support, or to report documentation errors, please visit:

<http://www.quectel.com/support/technical.htm>.

Or email us at: support@quectel.com.

Legal Notices

We offer information as a service to you. The provided information is based on your requirements and we make every effort to ensure its quality. You agree that you are responsible for using independent analysis and evaluation in designing intended products, and we provide reference designs for illustrative purposes only. Before using any hardware, software or service guided by this document, please read this notice carefully. Even though we employ commercially reasonable efforts to provide the best possible experience, you hereby acknowledge and agree that this document and related services hereunder are provided to you on an “as available” basis. We may revise or restate this document from time to time at our sole discretion without any prior notice to you.

Use and Disclosure Restrictions

License Agreements

Documents and information provided by us shall be kept confidential, unless specific permission is granted. They shall not be accessed or used for any purpose except as expressly provided herein.

Copyright

Our and third-party products hereunder may contain copyrighted material. Such copyrighted material shall not be copied, reproduced, distributed, merged, published, translated, or modified without prior written consent. We and the third party have exclusive rights over copyrighted material. No license shall be granted or conveyed under any patents, copyrights, trademarks, or service mark rights. To avoid ambiguities, purchasing in any form cannot be deemed as granting a license other than the normal non-exclusive, royalty-free license to use the material. We reserve the right to take legal action for noncompliance with abovementioned requirements, unauthorized use, or other illegal or malicious use of the material.

Trademarks

Except as otherwise set forth herein, nothing in this document shall be construed as conferring any rights to use any trademark, trade name or name, abbreviation, or counterfeit product thereof owned by Quectel or any third party in advertising, publicity, or other aspects.

Third-Party Rights

This document may refer to hardware, software and/or documentation owned by one or more third parties ("third-party materials"). Use of such third-party materials shall be governed by all restrictions and obligations applicable thereto.

We make no warranty or representation, either express or implied, regarding the third-party materials, including but not limited to any implied or statutory, warranties of merchantability or fitness for a particular purpose, quiet enjoyment, system integration, information accuracy, and non-infringement of any third-party intellectual property rights with regard to the licensed technology or use thereof. Nothing herein constitutes a representation or warranty by us to either develop, enhance, modify, distribute, market, sell, offer for sale, or otherwise maintain production of any our products or any other hardware, software, device, tool, information, or product. We moreover disclaim any and all warranties arising from the course of dealing or usage of trade.

Privacy Policy

To implement module functionality, certain device data are uploaded to Quectel's or third-party's servers, including carriers, chipset suppliers or customer-designated servers. Quectel, strictly abiding by the relevant laws and regulations, shall retain, use, disclose or otherwise process relevant data for the purpose of performing the service only or as permitted by applicable laws. Before data interaction with third parties, please be informed of their privacy and data security policy.

Disclaimer

- a) We acknowledge no liability for any injury or damage arising from the reliance upon the information.
- b) We shall bear no liability resulting from any inaccuracies or omissions, or from the use of the information contained herein.
- c) While we have made every effort to ensure that the functions and features under development are free from errors, it is possible that they could contain errors, inaccuracies, and omissions. Unless otherwise provided by valid agreement, we make no warranties of any kind, either implied or express, and exclude all liability for any loss or damage suffered in connection with the use of features and functions under development, to the maximum extent permitted by law, regardless of whether such loss or damage may have been foreseeable.
- d) We are not responsible for the accessibility, safety, accuracy, availability, legality, or completeness of information, advertising, commercial offers, products, services, and materials on third-party websites and third-party resources.

Copyright © Quectel Wireless Solutions Co., Ltd. 2023. All rights reserved.

About the Document

Revision History

Version	Date	Author	Description
-	2023-08-31	Noah YANG	Creation of the document
1.0	2023-10-27	Noah YANG	First official release

Contents

About the Document.....	3
Contents	4
Table Index.....	6
Figure Index	7
1 Introduction	8
1.1. Applicable Modules	8
2 Environment Preparations	9
2.1. Hardware Environment	9
2.2. Software Environment.....	11
3 Integration and Compilation	13
3.1. Code Integration.....	13
3.1.1. Patching BlueZ Source Code of Linux Platform	13
3.1.2. Copying Bluetooth Firmware to Linux Platform	14
3.2. Compilation	14
3.3. Image Downloading	14
4 Function Verification	15
4.1. Verifying Basic Bluetooth Function	15
4.1.1. Enabling Bluetooth Module	15
4.1.2. Initializing Bluetooth Module	15
4.1.3. Loading Bluetooth Daemon	16
4.2. Verifying Audio Play (Sink).....	17
4.2.1. Loading PulseAudio Service	17
4.2.2. Scanning and Connecting Device.....	17
4.2.3. Playing Audio.....	17
4.3. Verifying Bluetooth Audio (Source)	19
4.3.1. Loading PulseAudio Service	19
4.3.2. Scanning and Connecting Device.....	19
4.3.3. Playing Audio.....	19
4.4. Verifying BLE Protocol (Server)	20
4.4.1. Creating Server and Advertising	20
4.4.2. Test for Reading and Writing.....	21
4.5. Verifying BLE Protocol (Client)	23
4.5.1. Enabling Server and Advertising.....	23
4.5.2. Scanning and Connecting Device.....	23
4.5.3. Test for Reading and Writing.....	24
4.5.3.1. Test for Reading	24
4.5.3.2. Test for Writing	24
5 Usage of Tool	25
5.1. qlbt.....	25

5.1.1.	Enabling Bluetooth Module	25
5.1.2.	Obtaining Version	25
5.2.	Btdiag	26
5.2.1.	Enabling Bluetooth Module	26
5.2.2.	Using Btdiag with QRCT (Ethernet)	26
6	Appendix References	28

Table Index

Table 1: Applicable Modules.....	8
Table 2: Hardware Environment.....	9
Table 3: Software Environment	11
Table 4: Commands for Obtaining Bluetooth Driver Package	11
Table 5: Bluetooth Files.....	12
Table 6: Commands for Patching BlueZ Source Code	13
Table 7: Terms and Abbreviations	28

Figure Index

Figure 1: Top View of RK3568-WF EVB	10
Figure 2: Bottom View of RK3568-WF EVB.....	10
Figure 3: "SCANNER" interface	21
Figure 4: Created Characteristics	22
Figure 5: Enable Server and Advertising	23
Figure 6: QRCT Configuration	27

1 Introduction

Quectel FC2x, FC06E, FC6xE, FC900E, and FGE57xQ series modules support Bluetooth feature on the Linux platform. This document takes Quectel RK3568-WF EVB as an example to describe how to enable and verify Bluetooth feature with Quectel FC2x, FC06E, FC6xE, FC900E, and FGE57xQ modules on Linux platform.

1.1. Applicable Modules

Table 1: Applicable Modules

Module Family	Module
-	FC06E
FC2x	FC20 Series
	FC21
FC6xE	FC62E
	FC64E
	FC65E
	FC66E Series
-	FC900E
FGE57xQ	FGE573Q
	FGE576Q

2 Environment Preparations

2.1. Hardware Environment

Table 2: Hardware Environment

Hardware	Quantity
Quectel RK3568-WF EVB	1
Quectel Bluetooth Module	1
Antenna	1
USB Type-C Cable	1
Power Cable	1

NOTE

1. The development board model is for reference only. If you use other models, please consult the corresponding manufacturer for details.
2. RK3568-WF EVB is connected to the module through the M.2 interface. Please check whether the connection with each interface/GPIO of the module is correct according to the actual situation of the development board.
3. Please ensure that the Bluetooth antenna is connected properly. (The position of the antenna interface varies according to the module model. Please contact Quectel Technical Support for more details.)

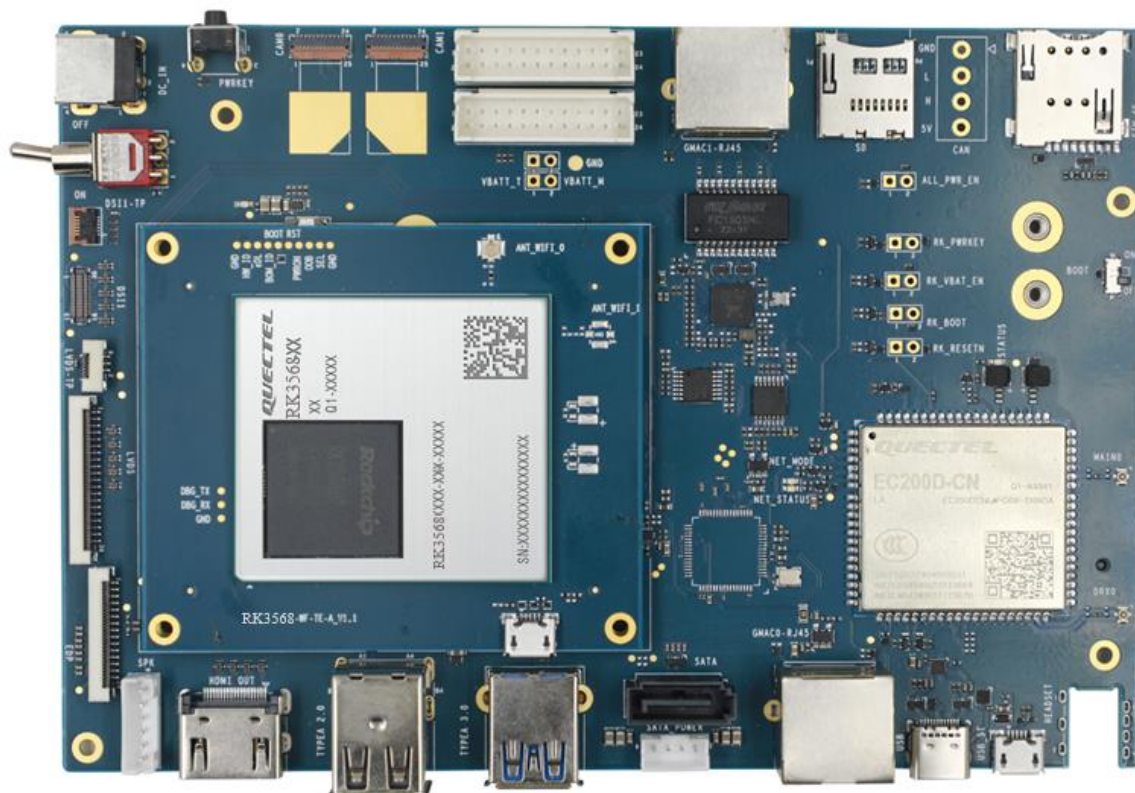


Figure 1: Top View of RK3568-WF EVB

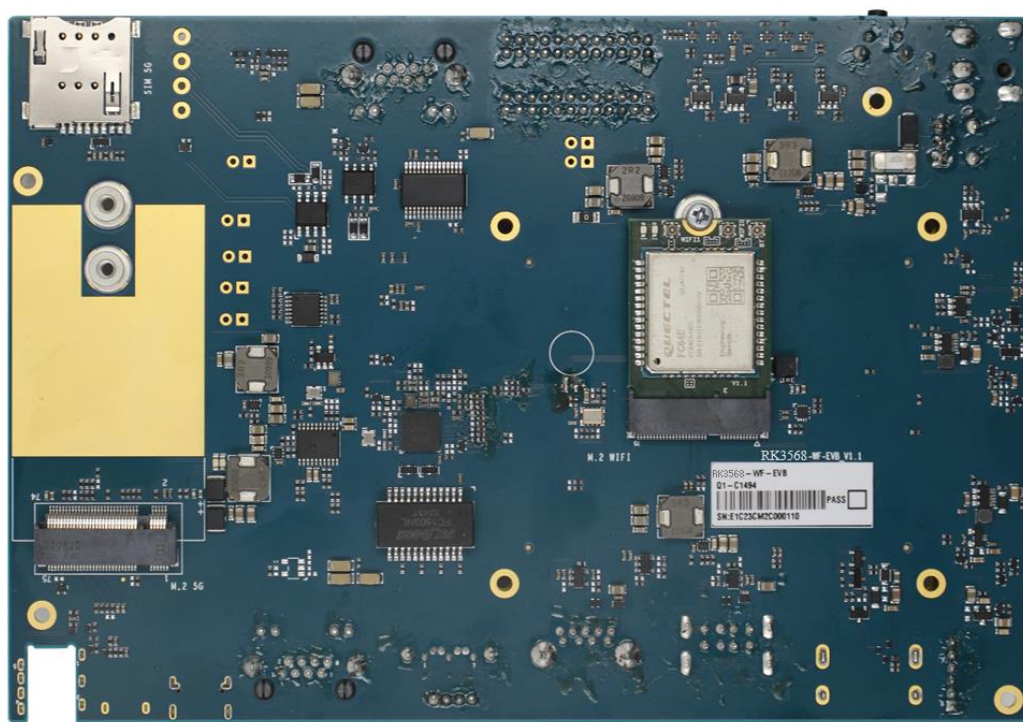


Figure 2: Bottom View of RK3568-WF EVB

2.2. Software Environment

This document uses the Linux BSP of Quectel RK3568-WF EVB as the software environment. You should prepare the Linux BSP that is compatible with the actual development board and meet the following requirements.

Table 3: Software Environment

Type	Description
Code environment	Linux 4.19
Driver package	Quectel Bluetooth driver package (include Bluetooth driver source code)
Compilation environment	Ubuntu 20.04
Tool	<ol style="list-style-type: none"> 1. qlbt and Btdiag tool 2. QRCT test tool 3. nRF Connect tool

Execute the following commands to obtain the Bluetooth driver package.

```
mkdir Quectel
cd Quectel
git clone <specific repository name>
```

Table 4: Commands for Obtaining Bluetooth Driver Package

Module	Command
FC20 series	git clone https://git-master.quectel.com/wifi.bt/fc2x
FC21	git clone https://git-master.quectel.com/wifi.bt/fc2x
FC900E	git clone https://git-master.quectel.com/wifi.bt/fc2x
FC06E	git clone https://git-master.quectel.com/wifi.bt/fc6xe
FC62E	git clone https://git-master.quectel.com/wifi.bt/fc6xe
FC64E	git clone https://git-master.quectel.com/wifi.bt/fc6xe
FC65E	git clone https://git-master.quectel.com/wifi.bt/fc6xe

FC66E series	git clone https://git-master.quectel.com/wifi.bt/fc6xe
FGE573Q	git clone https://git-master.quectel.com/wifi.bt/qcc207x
FGE576Q	git clone https://git-master.quectel.com/wifi.bt/qcc207x

The Quectel Bluetooth driver package contains the following information:

Table 5: Bluetooth Files

File Folder	Description
<i>BT/Linux/patch</i>	Includes patch files for the BlueZ source code package to provide support for the Quectel Bluetooth modules
<i>BT/Linux/tool</i>	Various toolkits for Bluetooth test <ul style="list-style-type: none"> ● qlbt is used to verify whether the communication between the Linux device (RK3568-WF EVB) and Quectel module is normal ● Btdiag is used for RF test with QRCT
<i>BT/FW</i>	Bluetooth firmware (users need to obtain the firmware from the subdirectory with the same name as the module model under <i>BT/FW</i> directory)
<i>BT/Android</i>	Bringup software package for Android platform (No need to pay attention)
<i>Doc/EN</i>	Wi-Fi and Bluetooth user guide documents in English version
<i>Doc/CN</i>	Wi-Fi and Bluetooth user guide documents in Chinese version
<i>WiFi</i>	Wi-Fi Bringup driver package (No need to pay attention)

NOTE

1. When obtaining the Bluetooth driver package, if necessary, please contact Quectel Technical Support to obtain git path access permission.
2. Linux BSP must support hardware flow control for UART, and should support at least 115200 bps and 3000000 bps baud rate.
3. The Linux BSP must support enabling the Bluetooth feature of the module through the BT_EN.

3 Integration and Compilation

3.1. Code Integration

3.1.1. Patching BlueZ Source Code of Linux Platform

Execute the following command to patch the BlueZ source code.

Table 6: Commands for Patching BlueZ Source Code

Module	Command
FC20 series	git am Quectel/fc2x-master/BT/Linux/patch/*.patch
FC21	git am Quectel/fc2x-master/BT/Linux/patch/*.patch
FC900E	git am Quectel/fc2x-master/BT/Linux/patch/*.patch
FC06E	git am Quectel/fc6xe-master/BT/Linux/patch/*.patch
FC62E	git am Quectel/fc6xe-master/BT/Linux/patch/*.patch
FC64E	git am Quectel/fc6xe-master/BT/Linux/patch/*.patch
FC65E	git am Quectel/fc6xe-master/BT/Linux/patch/*.patch
FC66E series	git am Quectel/fc6xe-master/BT/Linux/patch/*.patch
FGE573Q	git am Quectel/qcc207x-master/BT/Linux/patch/*.patch
FGE576Q	git am Quectel/qcc207x-master/BT/Linux/patch/*.patch

3.1.2. Copying Bluetooth Firmware to Linux Platform

Copy the *.tlv* and *.bin* files in the module Bluetooth firmware to the *lib/firmware/qca* directory under the Linux 4.19 source code.

NOTE

Since multiple modules share Bluetooth firmware, if the Bluetooth firmware of all modules is copied to rootfs at the same time, the RF parameters will not match. Therefore, please copy the specific module's Bluetooth firmware to rootfs according to your actual needs.

3.2. Compilation

Execute the following command in the *BlueZ* directory of the Linux 4.19 source code to compile BlueZ. (If BlueZ comes with the system, it is not necessary to compile it.)

```
./bootstrap-configure --disable-android --prefix=<user-defined BlueZ installation path>  
make  
make install
```

3.3. Image Downloading

After completing the above compilation steps, download the generated image file to the Linux platform (RK3568-WF EVB). For specific compilation and downloading steps, refer to: https://wiki.t-firefly.com/en/ROC-RK3568-PC/Windows_upgrade_firmware.html. Contact Quectel Technical Support for assistance, if necessary.

4 Function Verification

This chapter uses RK3568-WF EVB installed with Quectel Bluetooth module as an example to describe how to verify the Bluetooth function on Linux device.

NOTE

The function verification process for other Linux platforms is similar to the Bluetooth function verification described in this chapter. You can contact Quectel Technical Support if needed.

4.1. Verifying Basic Bluetooth Function

4.1.1. Enabling Bluetooth Module

Execute the following commands to enable module Bluetooth function.

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

4.1.2. Initializing Bluetooth Module

Execute the following commands to initialize the module.

```
hciattach /dev/ttyS1 qca 3000000 flow
hciconfig hci0 up
hciconfig -a
```



```
qca_soc_init: Download TLV file successfully
rome_set_baudrate_req: HCI CMD: 0x1 0x48 0xfc 0x1 0xe
## serial_set_baud: 14
read_vs_hci_event: Wait for HCI-Vendor Specific Event from SOC
wait_for_data: HCI-VS-EVENT available in TTY Serial buffer
read_vs_hci_event: Wait for HCI-Vendor Specific Event from SOC, buf[0] - 0x4
read_vs_hci_event: Wait for HCI-Vendor Specific Event from SOC, buf[1] - 0xe
get_vs_hci_event: Received HCI-Vendor Specific event
get_vs_hci_event: 0x 4 0x e 0x 4 0x 1 0x48 0xfc 0x 1 0x 0
get_vs_hci_event: Command response: 0x48
get_vs_hci_event: Response type : 0x0
get_vs_hci_event: EDL_SET_BAUDRATE_CMD_OCF err=0!!!
read_vs_hci_event: Wait for HCI-Vendor Specific Event from SOC, count - 0x7
rome_set_baudrate_req: Received HCI-Vendor Specific Event from SOC
rome_set_baudrate_req
rome_hci_reset_req: HCI RESET
rome_hci_reset_req: HCI CMD: 0x1 0x3 0xc 0x0
## serial_set_baud: 14
HCI Reset is done
Setting TTY to N_HCI line discipline
Device setup complete
```

```
hci0: Type: Primary Bus: UART
BD Address: 64:C4:03:88:FC:F1 ACL MTU: 1024:8 SCO MTU: 240:4
UP RUNNING
RX bytes:1422 acl:0 sco:0 events:80 errors:0
TX bytes:908 acl:0 sco:0 commands:80 errors:0
Features: 0xff 0xfe 0x8f 0xfe 0xd8 0x3f 0x5b 0x87
Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
Link policy: RSWITCH HOLD SNIFF
Link mode: PERIPHERAL ACCEPT
Name: ''
Class: 0x000000
Service Classes: Unspecified
Device Class: Miscellaneous,
HCI Version: (0xc) Revision: 0x0
LMP Version: (0xc) Subversion: 0x3a98
Manufacturer: Qualcomm (29)
```

4.1.3. Loading Bluetooth Deamon

Execute the following command in Linux 4.19 system to enable the Bluetooth daemon.

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

NOTE

1. On RK3568-WF EVB, it needs to execute the **rftkill** command to enable the Bluetooth module, and you can flexibly adjust it according to the actual situation.
2. Parameter `/dev/ttyS1` indicates the device number corresponding to the serial port of the Bluetooth module, and you can flexibly adjust it according to the actual situation.

3. The above operations are in the Linux 4.19 system, some steps may require sudo permissions, please flexibly adjust it according to the actual situation.

4.2. Verifying Audio Play (Sink)

4.2.1. Loading PulseAudio Service

Execute the following command in Linux 4.19 system to start the PulseAudio service.

```
pulseaudio &
```

4.2.2. Scanning and Connecting Device

Execute the following commands to scan and connect to the device.

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

NOTE

xx:xx:xx:xx:xx:xx indicates the Bluetooth address of the target device.

4.2.3. Playing Audio

Step 1: Check whether the Bluetooth connection icon is displayed on the phone to confirm whether the A2DP connection is successful. If the Bluetooth connection icon is displayed, it indicates that the A2DP connection is successful.

Step 2: Execute the following commands to set the sink of the PulseAudio service.

```
pactl list sinks
pactl set-default-sink 2 (2 needs to correspond to the actual sink#)
```

```
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
```

Step 3: Execute the following commands to set up the audio input device for the PulseAudio service.

```
pactl list sources
pactl set-default-source 5 (5 needs to correspond to the actual source#)
```

```
Source #5
State: SUSPENDED
Name: bluez_source.74_15_75_DC_F2_3C.a2dp_source
Description: Redmi K40
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 of Sink: n/a
Latency: 0 usec, configured 0 usec
Flags: HARDWARE DECIBEL_VOLUME LATENCY
```

Step 4: Play the audio file on the phone. If Bluetooth audio can be heard in the wired headset and audio-related information can be seen on the system terminal, it means that the audio playback function is normal.

```
[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: 0x0000001d (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:
```

NOTE

It's necessary to execute **amixer -c 0 cset numid=1 3** to enable the RK3568-WF EVB speaker. You can set mixer through the **amixer** command according to actual needs.

4.3. Verifying Bluetooth Audio (Source)

4.3.1. Loading PulseAudio Service

See **Chapter 4.2.1** for details.

4.3.2. Scanning and Connecting Device

See **Chapter 4.2.2** for details.

4.3.3. Playing Audio

Step 1: Check whether the Bluetooth connection icon is displayed on the phone to confirm whether the A2DP connection is successful. If the Bluetooth connection icon is displayed, it indicates that the A2DP connection is successful.

Step 2: Execute the following commands to set the sink of the PulseAudio service.

```
pactl list sinks
pactl set-default-sink 3 (3 needs to correspond to the actual sink#)
```

```
Sink #3
State: SUSPENDED
Name: bluez_sink.CD_45_71_E8_29_D4.a2dp_sink
Description: BT-SPEAKER
Driver: module-bluez5-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
```

Step 3: Play the audio file on the development board and see whether you can hear the audio from the Bluetooth headphones or Bluetooth speaker. If you can hear the audio, it means that the Bluetooth audio playback function is normal.

4.4. Verifying BLE Protocol (Server)

4.4.1. Creating Server and Advertising

Step 1: Execute `bluetoothctl` to run `bluetoothctl`.

Step 2: Execute `[bluetooth]# menu gatt` to enter the `gatt` subdirectory.

```
[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
```

Step 3: Execute `[bluetooth]# register-service e2d36f99-8909-4136-9a49-d825508b297b` to register the service `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
```

Step 4: Execute `[bluetooth]# register-characteristic 0x1234 read` to register a readable characteristic.

```
[bluetooth]# register-characteristic 0x1234 read
[00NEW00] Characteristic (Handle 0x0000)
      /org/bluez/app/service0/chrc0
      0x1234
      Unknown
[/org/bluez/app/service0/chrc0] Enter value: 1
```

Step 5: Execute `[bluetooth]# register-characteristic 0x5678 write` to register a writable characteristic.

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

Step 6: Execute the following commands to start advertising.

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

4.4.2. Test for Reading and Writing

Step 1: Install and open the application nRF Connect on the phone and enter the "SCANNER" interface.

Step 2: Find the target device and then click "**CONNECT**" to find the two created characteristics.

Step 3: Click the arrow on the phone to test reading and writing.

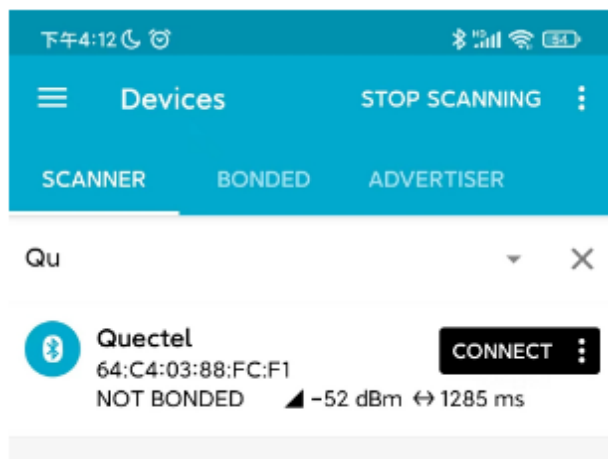


Figure 3: "SCANNER" interface

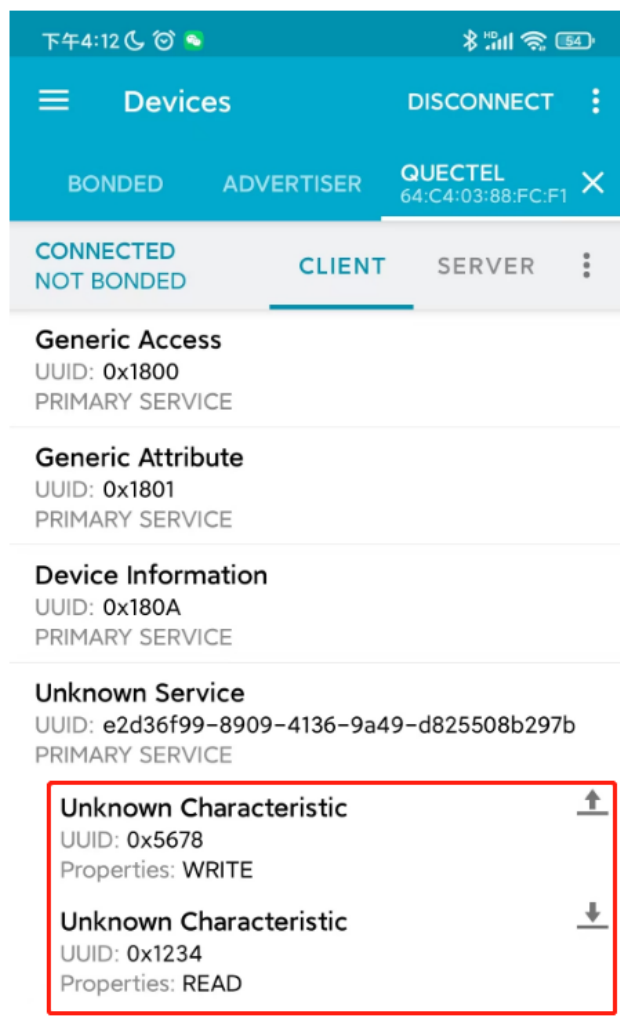


Figure 4: Created Characteristics

NOTE

For the downloading and using of nRF Connect tool, please refer to the official website: <https://www.nordicsemi.com/Products/Development-tools/nRF-Connect-for-mobile>.

4.5. Verifying BLE Protocol (Client)

4.5.1. Enabling Server and Advertising

Step 1: Open the application nRF Connect on the phone and enter the "ADVERTISER" interface to enable server and advertising.

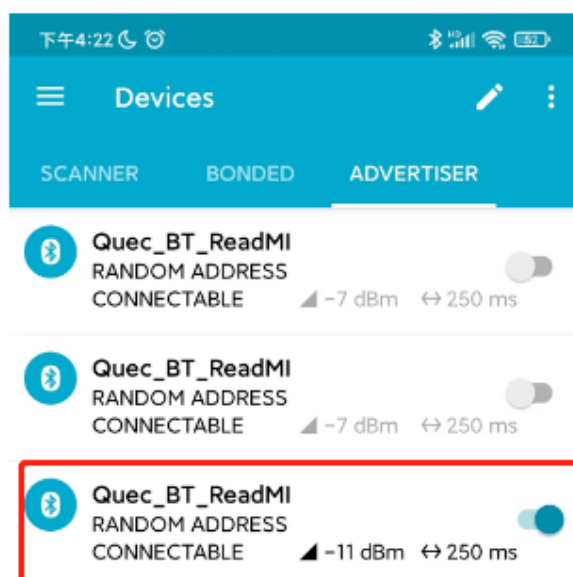


Figure 5: Enable Server and Advertising

4.5.2. Scanning and Connecting Device

Execute the following commands to scan and connect to the device.

```
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. Test for Reading and Writing

4.5.3.1. Test for Reading

Step 1: Execute the following command to select the attribute whose UUID is fff1 to receive data from the mobile phone.

```
select-attribute 0000fff1-0000-1000-8000-00805f9b34fb
```

Step 2: Execute the following command to enable notification function.

```
notify on
```

Step 3: Write data through nRF Connect on the phone. The UUID for writing data must correspond to the UUID in **select-attribute** command.

Step 4: Check whether the received data can be viewed in the printed information on the BlueZ running interface.

4.5.3.2. Test for Writing

Step 1: Execute the following command to select the attribute whose UUID is fff2 to write data to the mobile phone.

```
select-attribute 0000fff20000-1000-8000-00805f9b34fb
```

Step 2: Execute **write 1** to write data 1 (1 can be any value).

Step 3: Check the received data in the fff2 box of nRF Connect on the phone. If the received data is 1, it means that the data is written successfully.

5 Usage of Tool

This chapter takes Quectel RK3568-WF EVB as an example to introduce the use of Bluetooth related tools in the git repository.

5.1. qlbt

This tool is used to check whether the Linux device can communicate with the Bluetooth module through the UART when Bluetooth initialization fails.

5.1.1. Enabling Bluetooth Module

See **Chapter 4.1.1** for details.

5.1.2. Obtaining Version

Execute the following HCI commands to obtain the version of Bluetooth chip.

```
./qlbt -v /dev/ttyS1
```

If the information as shown in the red box in the figure below, it indicates that the main controller of the Linux platform can communicate with the Bluetooth module normally through the UART.



```
root@rockchip:/# ./qlbt -v /dev/ttyS1
send : 0x01 0x00 0xfc 0x01 0x19
recv : 0x04 0x0e 0x12 0x01 0x00 0xfc 0x00 0x19 0x0c 0x19 0x00 0x00 0x00 0xfb 0x43 0x00 0x02 0x00 0x02 0x17 0x40
```

5.2. Btdiag

This tool is used to verify the RF function of Bluetooth with QRCT tool.

5.2.1. Enabling Bluetooth Module

See **Chapter 4.1.1** for details.

5.2.2. Using Btdiag with QRCT (Ethernet)

Step 1: Execute **hciconfig -a** to get Bluetooth device ID.

Step 2: Execute the following command to enable Btdiag.

```
./Btdiag UDT=yes PORT=2390 IOType=USB QDARTIOType=ethernet BT-DEVICE=hci0
```

NOTE

Please choose the device ID according to your development board.

Step 3: Execute **ifconfig** to check the Ethernet IP address of the development board.

```
root@rockchip:/# ifconfig
eth0      Link encap:Ethernet  HWaddr 16:CA:8F:1A:82:A0
          inet addr:10.88.110.33  Bcast:10.88.110.255  Mask:255.255.255.0
          inet6 addr: fe80::b2aa:7525:5c3:2ed9/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3755  errors:0  dropped:230  overruns:0  frame:0
          TX packets:965  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:318061 (310.6 KiB)  TX bytes:125072 (122.1 KiB)
          Interrupt:41
```

Step 4: Open the QRCT tool and enter the TCP port set in **Step 2** into the input box next to “TCP Port”. Enter the IP address obtained in **Step 3** into the input box next to “IP Address”. Then, click the **"Connect"** button to connect the Btdiag tool. As shown in the figure below:



Figure 6: QRCT Configuration

After completing the above steps, you can proceed with the subsequent RF testing process. For specific RF testing procedures, please contact Quectel Technical Support.

NOTE

1. Using QRCT test tool requires authorization from Qualcomm.
2. The PC on which QRCT is used must be on the same network segment as the development board on which Btdiag runs.

6 Appendix References

Table 7: Terms and Abbreviations

Abbreviation	Description
A2DP	Advanced Audio Distribution Profile
ARM	Advanced RISC Machine
BLE	Bluetooth Low Energy
BSP	Board Support Package
BT	Bluetooth
CTS	Clear To Send
EVB	Evaluation Board
GATT	Generic Attribute Profile
GPIO	General-Purpose Input/Output
HCI	Host Controller Interface
ID	Mostly refers to Identifier in terms of software
IP	Internet Protocol
PC	Personal Computer
PCM	Pulse Code Modulation
RTS	Request to Send
Rx	Receive
TCP	Transmission Control Protocol
Tx	Transmit
UART	Universal Asynchronous Receiver/Transmitter

USB	Universal Serial Bus
UUID	Universally Unique Identifier
