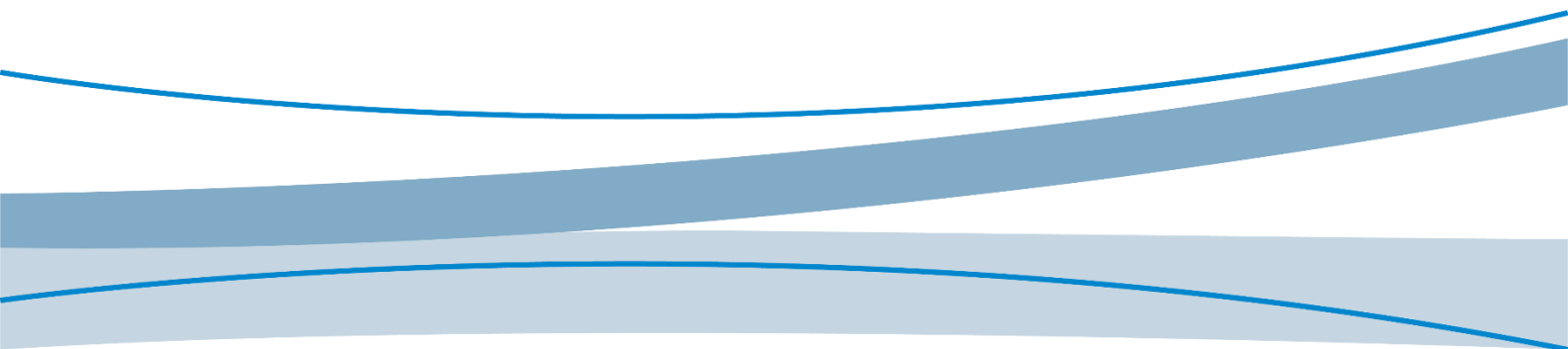




MTC

RIL Integration Guide_Android

V1.8



Disclaimer

Any action you take in the course of using this document is at your own risk, and Fibocom shall not be liable for any damages or losses under any circumstances. Due to product version upgrade or other reasons, Fibocom reserves the right to modify any information in this document at any time without prior notice and any responsibility. Unless otherwise agreed, all statements, information and suggestions in this document do not constitute any express or implied guarantee.

This document may include the third-party information covering products, services, software, data, and so on. Fibocom does not control and assumes no responsibility for the third-party content, including but not limited to the accuracy, compatibility, reliability, availability, legitimacy, appropriateness, performance, non-infringement, and status update, unless otherwise specified in this document. Fibocom does not provide any guarantee or authorization for the third-party content mentioned or referenced in this document. If you need a third-party license, obtain it in an authorized or legal way, unless otherwise specified in this document.

Copyright Notice

Copyright © 2025 Fibocom Wireless Inc. All rights reserved.

Unless specially authorized by Fibocom, the recipient of the documents shall keep the documents and information received confidential, and shall not use them for any purpose other than the implementation and development of this project. Without the written permission of Fibocom, no unit or individual shall extract or copy part or all of the contents of this document without authorization, or transmit them in any form. Fibocom has the right to investigate legal liabilities for any offense and tort in connection with violation of confidentiality obligations, or unauthorized use or malicious use of the said documents and information in other illegal forms.

Trademark Statement

 The trademark is registered and owned by Fibocom Wireless Inc.

Other trademarks, product names, service names and company names appearing in this document are owned by their respective owners.

Contact Information

Website: <https://www.fibocom.com>

Address: 10/F-14/F, Block A, Building 6, Shenzhen International Innovation Valley, Dashi First Road, Xili Community, Xili Subdistrict, Nanshan District, Shenzhen

Tel: 0755-26733555

Contents

Applicable Model.....	1
Change History.....	2
1 About This Document	3
2 RIL Driver Overview.....	4
2.1 Supported Functions.....	4
2.2 Supported Android Versions.....	4
3 RIL Integration Configuration	6
3.1 Start ril-daemon	6
3.1.1 Versions Before Android 8.....	6
3.1.2 Versions After Android 8.....	7
3.1.3 Parameter Description	7
3.2 Modify RILD Permissions	7
3.3 Modify RILC.....	9
3.3.1 Replace Libril.so Library	9
3.3.1.1 Confirm Current Version of libril.so Library	10
3.3.2 Modify RIL_VERSION	10
3.3.3 Modify manifest.xml.....	11
3.3.3.1 Radio1.1	11
3.3.3.2 Radio1.4.....	12
3.3.3.3 Radio1.6	12
3.4 Modify config_mobile_data_capable	14
3.5 Dial-up Driver Integration.....	15
3.5.1 QMI_WWAN Driver Integration	15
3.6 Load RIL Library.....	15
3.7 Choose Data Dialing Method.....	17
4 FAQs.....	19
4.1 How to Capture RIL Log	19
4.2 Unable to Access Internet.....	19
4.2.1 Check Basic Information	19

4.2.2 Check If Network Card Name Is Correct (ECM)	20
4.2.3 Check If Routing Is Correct (ECM).....	20
4.2.3.1 Problem Description	20
4.2.3.2 Solution	22
4.2.4 Check Files ip-up and ip-down (PPP).....	22
4.2.4.1 ip-up.....	22
4.2.4.2 ip-down	23
4.2.4.3 Modify Permissions.....	23
4.2.5 Capture Log for Fibocom to Analyze	23
4.3 Unable to Make a Call	24
4.4 Unable to Send SMS.....	24
4.5 How to Set APN.....	24
4.5.1 Set APN on UI Interface	24
4.5.2 Modify apns-conf.xml File	24
4.6 How to Modify Network Card Name Used by ECM.....	25
4.7 ght_ril_config.ini File	26
4.8 'Permission denied' Appears in RIL Log.....	26
4.8.1 Problem Description	26
4.8.2 Solution	27
4.9 How to Set System Properties	27
4.9.1 Property Setting Methods	27
4.9.2 ril.fibocom.dialmode	29
4.9.3 ril.fibocom.NetifName.....	30
4.9.4 ril.fibocom.usbmode	30
4.10 How to Solve Empty Triangles	31
4.10.1 Problem Description.....	31
4.10.2 Solution	32
Appendix A: Dial Mode and USBMODE	33
Appendix B: Acronyms and Abbreviations.....	34
Appendix C: Tags in Logs.....	35

Applicable Model

No.	Applicable Model	Description
1	L61x&LC61x&LG61x&MC61x&MG61x series	L610, LC610N, LG610, MC610, MC615, MC617, MC619
2	MC11x&MG110 series	MC116, MG110
3	NL668&NL668T series	NL668, NL668T
4	L71x series	L716, L718
5	MC919 series	MC919
6	MC66x&MG66x series	MC660, MC661, MC665, MC667, MC669, MG661, MG662
7	FG132 series	FG132
8	MA510 series	MA510

Change History

V1.8 (2024-09-11)	<ol style="list-style-type: none">1. Added applicable model MA510.2. Added content about Radio1.4 and Radio1.6. For details, see Section 3.3.
V1.7 (2024-06-11)	<ol style="list-style-type: none">1. Added applicable model FG132.2. Added QMI_WWAN data dialing.
V1.6 (2024-01-16)	<ol style="list-style-type: none">1. Fixed formatting errors.2. Removed applicable model LEx.
V1.5 (2023-09-26)	<ol style="list-style-type: none">1. Optimized the "Applicable Model" chapter.2. Optimized some screenshots.
V1.4 (2023-08-23)	<ol style="list-style-type: none">1. Added applicable model MG661.2. Added the adaptation of Android 12.
V1.3 (2023-06-07)	<ol style="list-style-type: none">1. Added applicable model MC619.2. Added description of the ght_ril_config.ini configuration file.
V1.2 (2023-05-10)	Added applicable model LE1x.
V1.1 (2023-04-26)	Added applicable model MG110.
V1.0 (2023-04-15)	Initial version.

1 About This Document

This document introduces how to integrate the Radio Interface Layer (RIL) driver to the customer's target system and how to modify the configuration file that starts the RIL service.

2 RIL Driver Overview

2.1 Supported Functions

Table 1. Supported functions

Function	Support
SMS	YES
Voice call	YES
Data service	YES
SIM card toolkit	NO

2.2 Supported Android Versions

Fibocom_Android_RIL currently supports Android versions listed in the following table.

Table 2. Supported Android versions

Function	Support
Android 2.x	NO
Android 3.x	NO
Android 4.x	YES
Android 5.x	YES
Android 6.x	YES
Android 7.x	YES
Android 8.x	YES
Android 9.x	YES
Android 10.x	YES
Android 11.x	YES
Android 12.x	YES
Android 13.x	YES
Android 14.x	YES



It is recommended to **use the 5G module on Android 11 or later versions**. Otherwise, 5G signal values, icons, and other information may display abnormally.

3 RIL Integration Configuration

This chapter describes the procedures for integrating the RIL library into the customer's Android system, including configuring the startup script, modifying the execution permission of the Android RILD and loading the RIL library.

3.1 Start ril-daemon

The ril-daemon process is responsible for finding and starting the RIL library, so it is important to first ensure that the ril-daemon is started and use the `-l` parameter to specify the path to the RIL library. The specific methods are as follows.

3.1.1 Versions Before Android 8

For versions earlier than Android 8 (excluding), you need to modify the `<Android_Dir>/device/xx/xx/init.rc` file to start ril-daemon.

- 32-bit Device

In the `init.rc` file, add the content in red font to specify the path to the RIL library as `"/system/lib/libreference-ril.so"`:

```
service ril-daemon /system/bin/rild -l /system/lib/libreference-ril.so -- -w 1
class main
socket rild stream 660 root radio
socket rild-debug stream 660 radio system
user root
group radio cache inet misc audio sdcard_rw log
```

- 64-bit Device

In the `init.rc` file, add the content in red font to specify the path to the RIL library as `"/system/lib64/libreference-ril.so"`:

```
service ril-daemon /system/bin/rild -l /system/lib64/libreference-ril.so -- -w 1
class main
socket rild stream 660 root radio
socket rild-debug stream 660 radio system
user root
group radio cache inet misc audio sdcard_rw log
```



`init.rc` file path varies depending on devices, for example `/device/rockchip/rk3399/init.rk3399.rc`.

3.1.2 Versions After Android 8



RIL libraries for Android 8 and later cannot be placed under the `/system` directory, otherwise, RIL's upper RILD cannot access it.

For Android 8 and later versions, it is necessary to modify the `<Android_Dir>/hardware/ril/rild.rc` file to start the ril-daemon.

- 32-bit Device

In the `rild.rc` file, add the content in red font to specify the path to the RIL library as `"/vendor/lib/libreference-ril.so"`:

```
service ril-daemon /vendor/bin/hw/rild -l /vendor/lib/libreference-ril.so -- -w 1
    class main
    user root
    group radio cache inet misc audio log readproc wakelock
    capabilities BLOCK_SUSPEND NET_ADMIN NET_RAW
```

- 64-bit Device

In the `rild.rc` file, add the content in red font to specify the path to the RIL library as `"/vendor/lib64/libreference-ril.so"`:

```
service ril-daemon /vendor/bin/hw/rild -l /vendor/lib64/libreference-ril.so -- -w 1
    class main
    user root
    group radio cache inet misc audio log readproc wakelock
    capabilities BLOCK_SUSPEND NET_ADMIN NET_RAW
```

3.1.3 Parameter Description

In the `init.rc` (or `rild.rc`) script, the `-l` and `-w` parameters work as follows:

- `-l`: Specifies the RIL library load path.
- `-w`: Specifying the dialing Mode.
 - 0: ppp Dial mode
 - 1: ECM dial mode
 - 3: QMI_WWAN dial mode

3.2 Modify RILD Permissions

ril-daemon is the name of the process and it actually runs the executable RILD, so some adaptation changes need to be made for RILD as well.

1. Because root permission is required during the execution of the RILD program, it is necessary to remove switchuser() from the main function in the rild.c file (<\$android dir> /hardware/ril/rild/rild.c), as shown in the following figure.



Some Android versions do not have the switchUser() function, such as Android 10, so you can directly proceed to the next step.

```

224     arg_device[sizeof(arg_device)-1] = 0;
225     arg_overrides[1] = "-d";
226     arg_overrides[2] = arg_device;
227     done = 1;
228
229     } while (0);
230
231     if (done) {
232         argv = arg_overrides;
233         argc = 3;
234         i = 1;
235         hasLibArgs = 1;
236         rilLibPath = REFERENCE_RIL_PATH;
237
238         | RLOGD("overriding with %s %s", arg_overrides[1], arg_overrides[2]);
239     }
240 }
241 OpenLib:
242 #endif
243     //switchUser();
244
245     dlHandle = dlopen(rilLibPath, RTLD_NOW);
246
247     if (dlHandle == NULL) {
248         RLOGE("dlopen failed: %s", dlerror());
249         exit(-1);
250     }
251

```

Figure 1. Modify the main function in RILD

2. To ensure that RILD participates in compilation, check whether the following content is present in the <Android_Dir>/device/xx/device.mk file, and if not, add it as shown in the following figure.

```
PRODUCT_PACKAGES += rild
```

```

605 PRODUCT_PROPERTY_OVERRIDES += \
606     rild.libpath=/system/lib/libril-rk29-dataonly.so
607 endif
608 endif
609
610 ifeq ($(strip $(BOARD_HAS_RK_4G_MODEM)),true)
611 PRODUCT_PACKAGES += \
612     rild \
613     librk-ril \
614     dhcpcd
615

```

Figure 2. Compile RILD



device.mk file path varies depending on devices, for example, /device/rockchip/rk3399/device.mk.

3.3 Modify RILC

Starting from Android 8, the android.hardware.radio service has been introduced and its versions are constantly being updated, such as radio1.1, radio1.4, and radio1.6. Following is the recommended radio version.

For **Fibocom_Android_RIL_1.0.0.12 and earlier** versions, use **radio1.1**.

The recommended radio versions for **Fiocom_Android_RIL_1.0.0.13 and later** RIL versions are as follows.

Table 3. Supported functions

Android	Radio
4-7	NA
8-9	1.1
10-14	1.4

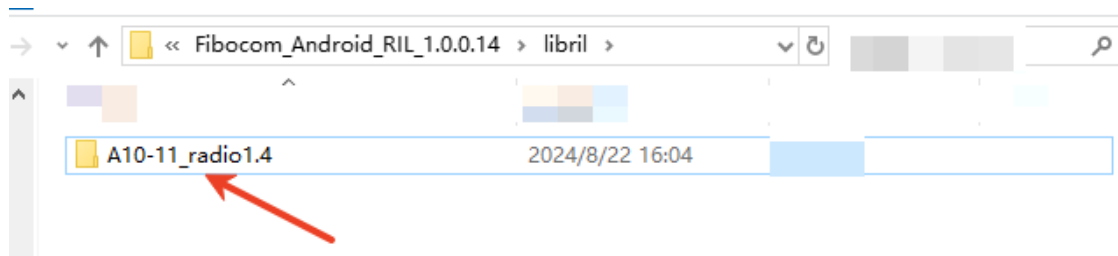
To use the corresponding radio version, make modifications described in the following sections.



- The remaining content of this chapter is only applicable to **Fiocom_Android_RIL_1.0.0.13 and later** RIL versions
- The Android and radio versions in the table do not have a mandatory mapping. For example, Android 11 can also use radio 1.1, as long as correct mapping among **libril.so**, **RIL_VERSION**, and **manifest.xml** is ensured in the following text.

3.3.1 Replace Libril.so Library

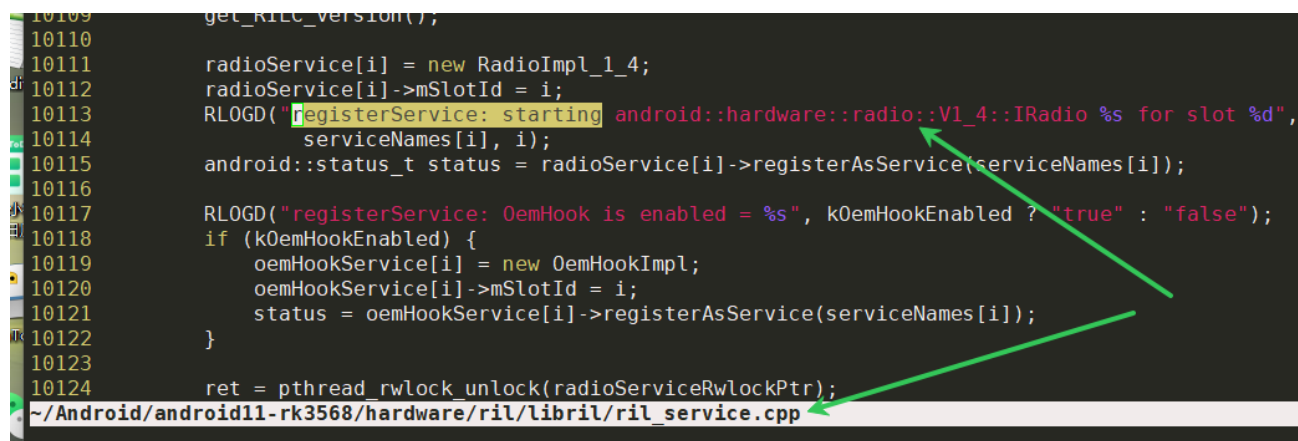
By default, Fibocom provides the **libril.so** library for radio1.4, and places the **libril.so** library in the **/vendor/lib(64)/libril.so** path on Android devices to take effect.



radio1.1 is usually the default version of Android and does not require special provisioning. If you need radio1.6, contact Fibocom technical support for assistance.

3.3.1.1 Confirm Current Version of libril.so Library

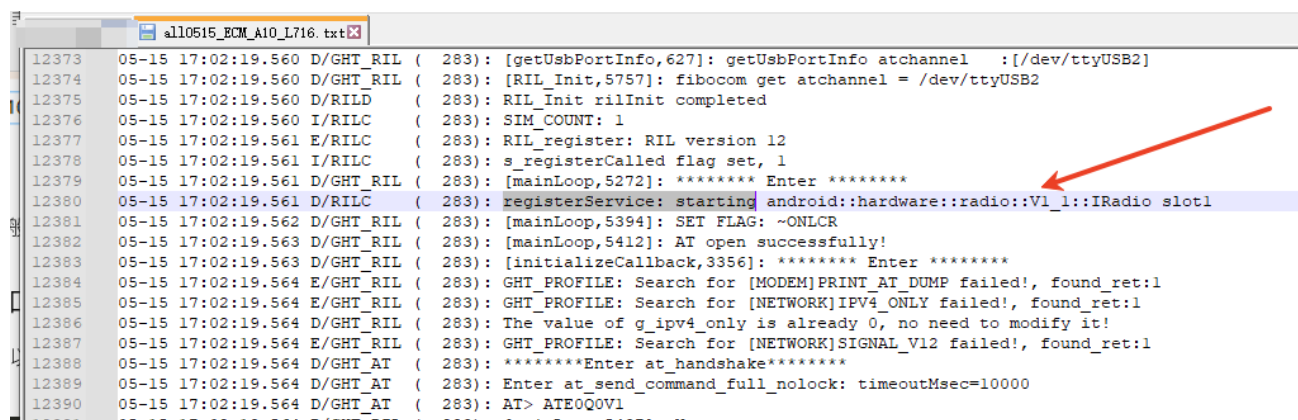
For Android version 8 and later, search for the keyword "registerService: starting" in the <Android>/hardware/ril/libril/ril_service.cpp file or Android log. If the result contains the keyword "V1_1", it indicates that it is a radio1.1 library; If it is "V1_4", it means it is a library for radio1.4, and so on.



```

10109 get_RILC_version();
10110
10111 radioService[i] = new RadioImpl_1_4;
10112 radioService[i]->mSlotId = i;
10113 RLOGD("registerService: starting android::hardware::radio::V1_4::IRadio %s for slot %d",
10114       serviceNames[i], i);
10115 android::status_t status = radioService[i]->registerAsService(serviceNames[i]);
10116
10117 RLOGD("registerService: OemHook is enabled = %s", kOemHookEnabled ? "true" : "false");
10118 if (kOemHookEnabled) {
10119     oemHookService[i] = new OemHookImpl;
10120     oemHookService[i]->mSlotId = i;
10121     status = oemHookService[i]->registerAsService(serviceNames[i]);
10122 }
10123
10124 ret = pthread_rwlock_unlock(radioServiceRwlockPtr);
~/Android/android11-rk3568/hardware/ril/libril/ril_service.cpp

```



```

12373 05-15 17:02:19.560 D/GHT_RIL ( 283): [getUsbPortInfo,627]: getUsbPortInfo atchannel :[/dev/ttyUSB2]
12374 05-15 17:02:19.560 D/GHT_RIL ( 283): [RIL_Init,5757]: fibocom get atchannel = /dev/ttyUSB2
12375 05-15 17:02:19.560 D/RILD ( 283): RIL_Init rilInit completed
12376 05-15 17:02:19.560 I/RILC ( 283): SIM_COUNT: 1
12377 05-15 17:02:19.561 E/RILC ( 283): RIL_register: RIL version 12
12378 05-15 17:02:19.561 I/RILC ( 283): s_registerCalled flag set, 1
12379 05-15 17:02:19.561 D/GHT_RIL ( 283): [mainLoop,5272]: ***** Enter *****
12380 05-15 17:02:19.561 D/RILC ( 283): registerService: starting android::hardware::radio::V1_1::IRadio slot1
12381 05-15 17:02:19.562 D/GHT_RIL ( 283): [mainLoop,5394]: SET FLAG: ~ONLCR
12382 05-15 17:02:19.563 D/GHT_RIL ( 283): [mainLoop,5412]: AT open successfully!
12383 05-15 17:02:19.563 D/GHT_RIL ( 283): [initializeCallback,3356]: ***** Enter *****
12384 05-15 17:02:19.564 E/GHT_RIL ( 283): GHT_PROFILE: Search for [MODEM]PRINT AT_DUMP failed!, found_ret:1
12385 05-15 17:02:19.564 E/GHT_RIL ( 283): GHT_PROFILE: Search for [NETWORK]IPV4_ONLY failed!, found_ret:1
12386 05-15 17:02:19.564 D/GHT_RIL ( 283): The value of g_ipv4_only is already 0, no need to modify it!
12387 05-15 17:02:19.564 E/GHT_RIL ( 283): GHT_PROFILE: Search for [NETWORK]SIGNAL_V12 failed!, found_ret:1
12388 05-15 17:02:19.564 D/GHT_AT ( 283): *****Enter at_handshake*****
12389 05-15 17:02:19.564 D/GHT_AT ( 283): Enter at_send_command_full_nolock: timeoutMsec=10000
12390 05-15 17:02:19.564 D/GHT_AT ( 283): AT> ATE0Q0V1

```

3.3.2 Modify RIL_VERSION

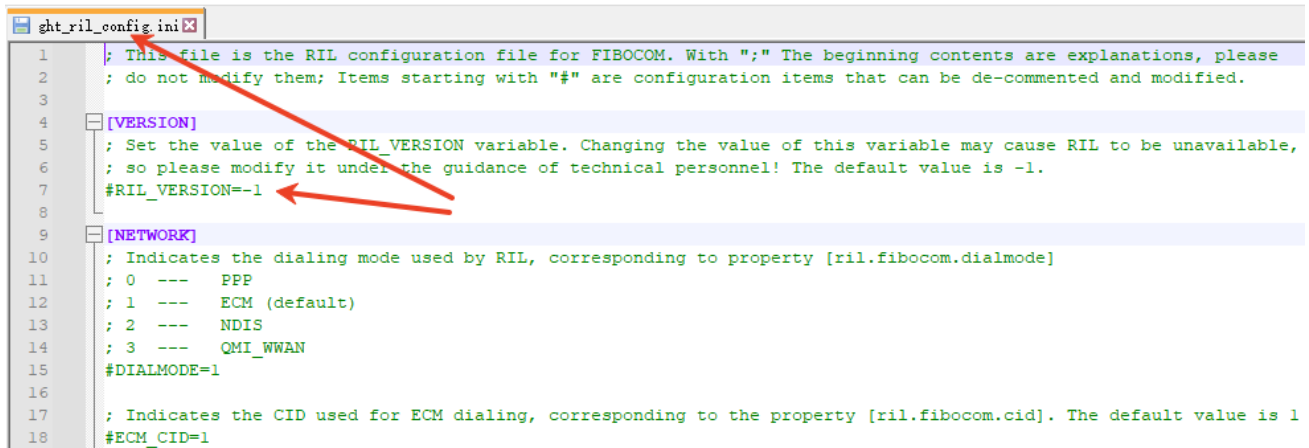
Change the RIL_VERSION in the [ghl_ril_config.ini file](#) to the corresponding value in the following table.



It is not recommended to use radio versions that are not listed in the table below.

Table 4. RIL_VERSION for different radio versions

Radio Version	RIL_VERSION
radio1.1	12
radio1.4	14
radio1.6	16



```

1 ; This file is the RIL configuration file for FIBOCOM. With ";" The beginning contents are explanations, please
2 ; do not modify them; Items starting with "#" are configuration items that can be de-commented and modified.
3
4 [VERSION]
5 ; Set the value of the RIL_VERSION variable. Changing the value of this variable may cause RIL to be unavailable,
6 ; so please modify it under the guidance of technical personnel! The default value is -1.
7 #RIL_VERSION=-1
8
9 [NETWORK]
10 ; Indicates the dialing mode used by RIL, corresponding to property [ril.fibocom.dialmode]
11 ; 0 --- PPP
12 ; 1 --- ECM (default)
13 ; 2 --- NDIS
14 ; 3 --- QMI_WWAN
15 #DIALMODE=1
16
17 ; Indicates the CID used for ECM dialing, corresponding to the property [ril.fibocom.cid]. The default value is 1
18 #ECM_CID=1

```

3.3.3 Modify manifest.xml

For Android 8 and later versions, the following content needs to be added to the `<Android>/device/xx/xx/manifest.xml` file to ensure the normal communication between the upper layer of Android and ril-daemon.



manifest.xml path varies depending on devices, for example, `/device/dockchip/rk3399/manifest.xml`.

3.3.3.1 Radio1.1

```

<hal format="hidl">
    <name>android.hardware.radio</name>
    <transport>hwbinder</transport>
    <fqname>@1.1::IRadio/slot1</fqname>
    <fqname>@1.1::IRadio/slot2</fqname>
    <fqname>@1.2::ISap/slot1</fqname>
</hal>

<hal format="hidl">
    <name>android.hardware.radio.deprecated</name>
    <transport>hwbinder</transport>
    <version>1.0</version>
    <interface>
        <name>IOemHook</name>
        <instance>slot1</instance>
    </interface>
    <fqname>@1.0::IOemHook/slot1</fqname>
</hal>

```

3.3.3.2 Radio1.4

```
<hal format="hidl">
  <name>android.hardware.radio</name>
  <transport>hwbinder</transport>
  <fqname>@1.4::IRadio/slot1</fqname>
  <fqname>@1.4::IRadio/slot2</fqname>
  <fqname>@1.2::ISap/slot1</fqname>
  <fqname>@1.2::ISap/slot2</fqname>
</hal>

<hal format="hidl">
  <name>android.hardware.radio.config</name>
  <transport>hwbinder</transport>
  <version>1.1</version>
  <interface>
    <name>IRadioConfig</name>
    <instance>default</instance>
  </interface>
</hal>
```

3.3.3.3 Radio1.6

The suggested additions for Android 13 and 14 are slightly different, as follows:

3.3.3.3.1 Android13

```
<hal format="aidl">
  <name>android.hardware.radio.config</name>
  <fqname>IRadioConfig/default</fqname>
</hal>

<hal format="aidl">
  <name>android.hardware.radio.data</name>
  <fqname>IRadioData/slot1</fqname>
</hal>

<hal format="aidl">
  <name>android.hardware.radio.messaging</name>
  <fqname>IRadioMessaging/slot1</fqname>
</hal>

<hal format="aidl">
  <name>android.hardware.radio.modem</name>
```



```
<fqname>IRadioModem/slot1</fqname>
</hal>
<hal format="aidl">
  <name>android.hardware.radio.network</name>
  <fqname>IRadioNetwork/slot1</fqname>
</hal>
<hal format="aidl">
  <name>android.hardware.radio.sim</name>
  <fqname>IRadioSim/slot1</fqname>
</hal>
<hal format="aidl">
  <name>android.hardware.radio.voice</name>
  <fqname>IRadioVoice/slot1</fqname>
</hal>
```

3.3.3.3.2 Android14

```
<hal format="aidl">
  <name>android.hardware.radio.config</name>
  <version>2</version>
  <fqname>IRadioConfig/default</fqname>
</hal>
<hal format="aidl">
  <name>android.hardware.radio.data</name>
  <version>2</version>
  <fqname>IRadioData/slot1</fqname>
</hal>
<hal format="aidl">
  <name>android.hardware.radio.ims</name>
  <fqname>IRadioIms/slot1</fqname>
</hal>
<hal format="aidl">
  <name>android.hardware.radio.ims.media</name>
  <fqname>IImsMedia/default</fqname>
</hal>
<hal format="aidl">
  <name>android.hardware.radio.messaging</name>
  <version>2</version>
  <fqname>IRadioMessaging/slot1</fqname>
</hal>
```

```
<hal format="aidl">
  <name>android.hardware.radio.modem</name>
  <version>2</version>
  <fqname>IRadioModem/slot1</fqname>
</hal>
<hal format="aidl">
  <name>android.hardware.radio.network</name>
  <version>2</version>
  <fqname>IRadioNetwork/slot1</fqname>
</hal>
<hal format="aidl">
  <name>android.hardware.radio.sim</name>
  <version>2</version>
  <fqname>IRadioSim/slot1</fqname>
</hal>
<hal format="aidl">
  <name>android.hardware.radio.sap</name>
  <fqname>ISap/slot1</fqname>
</hal>
<hal format="aidl">
  <name>android.hardware.radio.voice</name>
  <version>2</version>
  <fqname>IRadioVoice/slot1</fqname>
</hal>
```

3.4 Modify config_mobile_data_capable

Check the value of `config_mobile_data_capable` in file

`<Android>/device/./overlay/frameworks/base/core/res/res/values/config.xml` or

`<Android>/frameworks/base/core/res/res/values/config.xml`. If it is `false`, change it to `true`.



Focus on checking the `config_mobile_data_capable` value in the **device** directory, as the values in this directory **have higher priority** and will overwrite the values in the framework directory.

```

H AD      config.xml      35 <bool name="config_mobile_data_capable">false</bool>
H AD      config.xml      130 <bool name="config_mobile_data_capable">false</bool>
H AD      config.xml      605 <bool name="config_mobile_data_capable">false</bool>
H AD      symbols.xml     319 <java-symbol type="bool" name="config_mobile_data_capable" />
H AD      config.xml      2173 <bool name="config_mobile_data_capable">true</bool>

```

3.5 Dial-up Driver Integration

3.5.1 QMI_WWAN Driver Integration



- Skip this chapter for non-Qualcomm platform modules.
- For Qualcomm platform modules such as FG132, NL668, or NL668T, ECM dialing is recommended. This section is not mandatory and can be skipped.

Before using QMI_WWAN for dialing, it is necessary to integrate the QMI_WWAN driver in the Android source code. For the specific process, refer to the "QMI_WWAN Dial-up" section in the *Fibocom MTC Dial-up Guide Linux* document.

3.6 Load RIL Library

1. After completing the previous steps, you can compile your Android image (The compilation methods vary depending on the actual situation). After compilation, download the image to the Android device.
2. If the QMI_WWAN driver was integrated earlier, you need to confirm whether the integration is successful through the following steps (if the QMI_WWAN driver is not integrated, please skip this step).
 - a. Refer to Chapter [Dial Mode and USBMODE](#) and the product model to find the USBMODE corresponding to QMI_WWAN, and then change the USBMODE of the module to the corresponding value. The following figure shows the USBMODE of the FG132 module is changed to **36** required by QMI WWAN on an Android device.

```
[10.111.ecc11st]: [112, 911]
rk3568_firefly_aioj:/ # stop ril-daemon
rk3568_firefly_aioj:/ # cat /dev/ttyUSB1 &
[1] 2211
rk3568_firefly_aioj:/ # echo -en "AT+GTUSBMODE=36\r" > /dev/ttyUSB1
rk3568_firefly_aioj:/ #
```

OK

```
rk3568_firefly_aioj:/ #
```

b. Connect the module to Android, and if the QMI_WWAN driver is successfully integrated, the following "wwan0" network card is expected to appear. If it does not appear, carefully check chapter [QMI WWAN Driver Integration](#) or contact Fibocom technical support.

```
rk3568_firefly_aioj:/ # ifconfig wwan0
wwan0      Link encap:Ethernet  HWaddr 26:64:27:a8:d4:e2  Driver qmi_wwan_f
           NOARP  MTU:1500  Metric:1
           RX packets:0 errors:0 dropped:0 overruns:0 frame:0
           TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:0 TX bytes:0
```

3. Use the `getprop | grep ril` command to query whether the status of ril-daemon is "running", as shown in the following figure.

```
console:/system/lib # getprop | grep ril
[init.svc.vendor.ril-daemon]: [running]
[init.svc_debug_pid.vendor.ril-daemon]: [502]
[rild.libpath]: [/system/lib/libreference-ril.so]
[ro.boottime.vendor.ril-daemon]: [9581217087]
console:/system/lib # getprop | grep ril
[init.svc.vendor.ril-daemon]: [running]
[init.svc_debug_pid.vendor.ril-daemon]: [502]
[rild.libpath]: [/system/lib/libreference-ril.so]
[ro.boottime.vendor.ril-daemon]: [9581217087]
```

4. Push the RIL library into the Android device to run.

a. Use the following command to enable Android device file permissions:

```
adb root
adb remount
```

b. Push the RIL library to different paths for different systems. The details are as follows:

Before Android 8, 32-bit systems:

```
adb push <local-dir>/libreference-ril.so /system/lib
```

Before Android 8, 64-bit systems:

```
adb push <local-dir>/libreference-ril.so /system/lib64
```

Android 8 or later, 32-bit:

```
adb push <local-dir>/libreference-ril.so /vendor/lib
```

Android 8 or later, 64-bit:

```
adb push <local-dir>/libreference-ril.so /vendor/lib64
```

c. Reboot the Android device, if there is "ril.fibocom.version" property in the result of "getprop | grep ril", it means that the RIL library is running properly, as shown in the following figure.

```

2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 57.935/60.009/62.084/2.088 ms
rk3399_roc_pc_plus:/ $ getprop|grep ril
[gsm.version.ril-impl]: [Fibocom_RIL_V10X.06.V1.0.7_001]
[init.svc.ril-daemon]: [running]
[persist.ril.current_usbmode]: [17]
[ril.currentapntype]: [default]
[ril.fibocom.version]: [Fibocom_RIL_V10X.06.V1.0.7_001]
[ril.function.dataonly]: [1]
[rild.libargs]: [-d]
[rild.libpath]: [/vendor/lib64/libreference-ril.so]
[ro.boot.noril]: [true]
[ro.boottime.ril-daemon]: [3637641642]
[ro.ril.ecclist]: [112,911]
rk3399_roc_pc_plus:/ $ su
:/ # find -name rild 2>/dev/null
./vendor/bin/hw/rild
:/ #

```

Now that the RIL library has been successfully integrated, choose a dialing method.

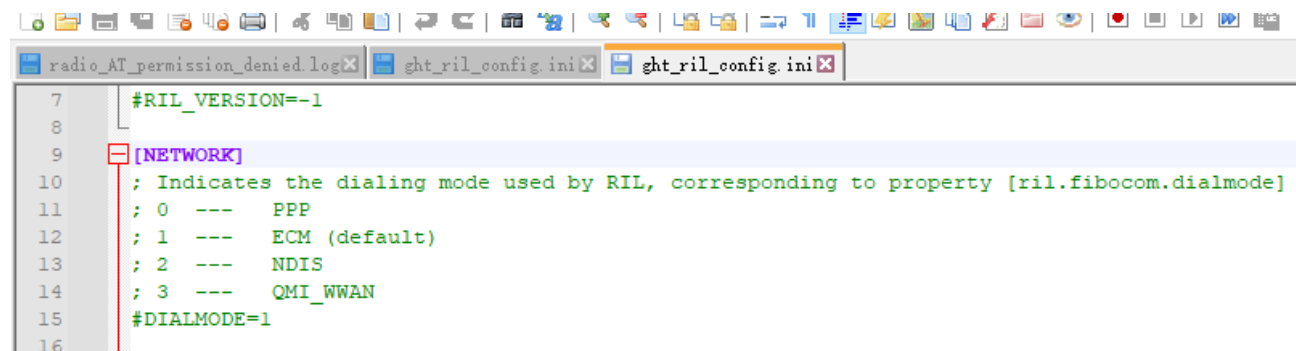
3.7 Choose Data Dialing Method

At present, there are three dialing methods available: ECM, PPP, and QMI_WWAN. Fibocom RIL library uses ECM dialing by default and you can skip this section if you use ECM dialing. If you want to switch to another dialing method, refer to the following steps..

1. The methods for modifying dialing methods vary depending on different RIL versions, as follows:

- **Fibocom_Android_RIL_1.0.6 and later**

For Fibocom_Android_RIL_1.0.6 and later versions of RIL, it is recommended to modify the **DIALMODE** value in the `ght_ril_config.ini` file, where **value 2 for NDIS dialing is not recommended**.



- **Before Fibocom_Android_RIL_1.0.6**

For versions earlier than Fibocom_Android_RIL_1.0.6, RIL did not support the `ght_ril_config.ini` file, so the following property values need to be set:

```

ril.fibocom.dialmode=<dialmode>    //0 : PPP
                                   //1 : ECM
                                   //3 : QMI_WWAN

```

Please refer to Chapter [How to Set System Properties?](#) for specific methods of setting properties.

2. Connect the module to the Android device and insert the SIM card, then reboot the Android device

and wait for about 30 seconds.

- If the device is able to access the internet or the ping command returns normally, it indicates successful dialing, as shown in the following figure.

```
rk3399_roc_pc_plus:/ $ ping www.qq.com
PING ins-r23tsuuf.ias.tencent-cloud.net (183.194.238.19) 56(84) bytes of data.
64 bytes from (183.194.238.19): icmp_seq=1 ttl=246 time=70.1 ms
64 bytes from (183.194.238.19): icmp_seq=2 ttl=246 time=86.3 ms
64 bytes from (183.194.238.19): icmp_seq=3 ttl=246 time=86.8 ms
64 bytes from (183.194.238.19): icmp_seq=4 ttl=246 time=83.7 ms
64 bytes from (183.194.238.19): icmp_seq=5 ttl=246 time=60.8 ms
64 bytes from (183.194.238.19): icmp_seq=6 ttl=246 time=62.3 ms
rk3399_roc_pc_plus:/ $
```

- If the device is unable to access the internet, refer to Chapter [Unable to Access the Internet](#) for troubleshooting

4 FAQs

4.1 How to Capture RIL Log

In the cmd window of the Windows operating system, enter the following command:

```
adb logcat -b all -v time > radio.txt
```

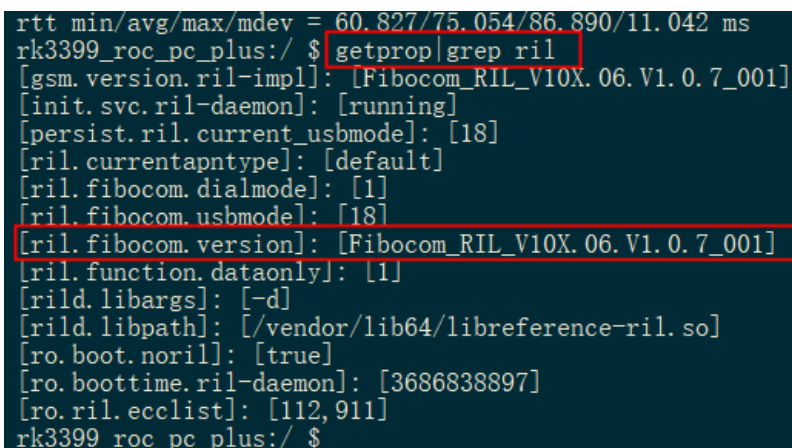
The captured logs will be saved to the **radio.txt** file.

4.2 Unable to Access Internet

If the Android device cannot access the internet, you can follow the following steps to troubleshoot the problem.

4.2.1 Check Basic Information

1. Use the **adb shell** command to enter the Android device, and then use the **getprop | grep ril** command to check if there is a "ril.fibocom.version" property, as shown in the following figure.



```
rtt min/avg/max/mdev = 60.827/75.054/86.890/11.042 ms
rk3399_roc_pc_plus:/ $ getprop | grep ril
[gsm.version.ril-impl]: [Fibocom_RIL_V10X.06.V1.0.7_001]
[init.svc.ril-daemon]: [running]
[persist.ril.current_usbmode]: [18]
[ril.currentapntype]: [default]
[ril.fibocom.dialmode]: [1]
[ril.fibocom.usbmode]: [18]
[ril.fibocom.version]: [Fibocom_RIL_V10X.06.V1.0.7_001]
[ril.function.dataonly]: [1]
[rild.libargs]: [-d]
[rild.libpath]: [/vendor/lib64/libreference-ril.so]
[ro.boot.noril]: [true]
[ro.boottime.ril-daemon]: [3686838897]
[ro.ril.ecclist]: [112,911]
rk3399_roc_pc_plus:/ $
```

- a. If this property is not found, the RIL library has not been loaded properly. Please refer to chapter "RIL Integration Configuration" to configure RIL library first.
- b. If the RIL library is running normally, use the **ls /dev/ttyUSB*** command to confirm if the module recognizes the port:

```
root@android:/ # ls /dev/ttyUSB*
```

- If the device is mounted normally, the following content will be returned:

```
ls /dev/ttyUSB*
/dev/ttyUSB0
/dev/ttyUSB1
/dev/ttyUSB2
/dev/ttyUSB3
```


- If these ports are not recognized, it indicates that the module has not been properly mounted. First, determine whether the module is compatible with RIL based on the [applicable model](#), and then check whether the module power supply is normal and whether the connection with Android devices is normal.
2. If the module recognizes the port normally, confirm that the SIM card is functioning properly, there are no arrears, cancellations, and the antenna connection is normal.

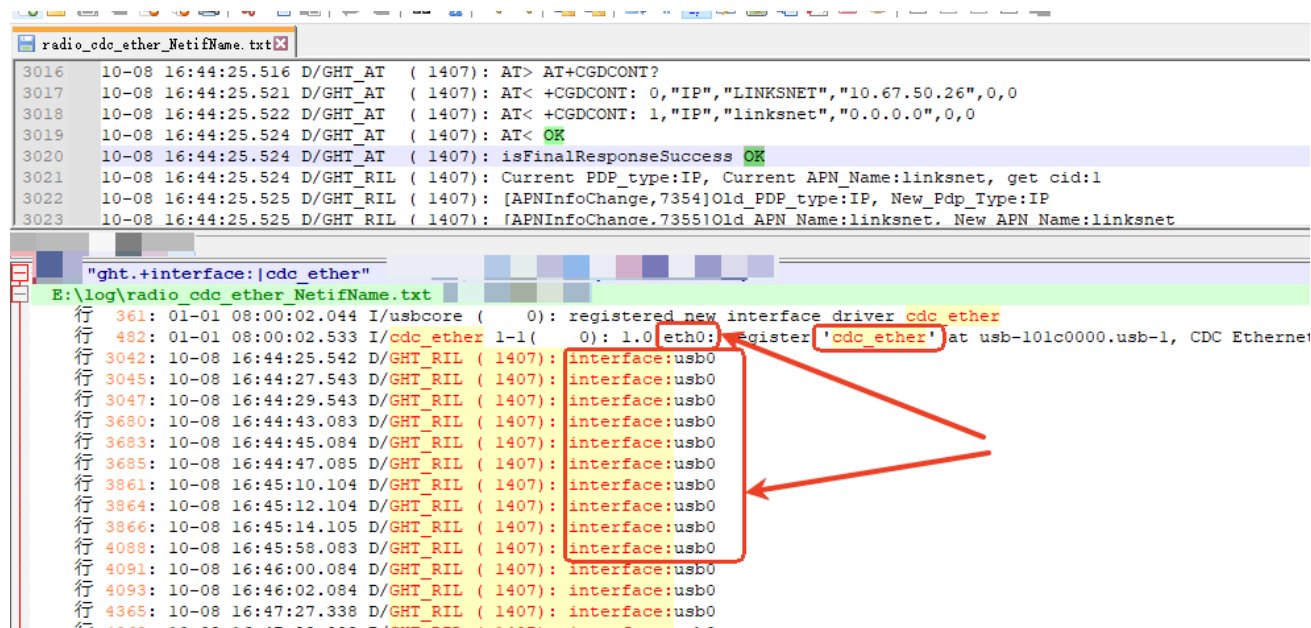
4.2.2 Check If Network Card Name Is Correct (ECM)

If you are currently using ECM dial-up and it is Fibocom_Android_RIL_1.0.7 or earlier RIL versions, you need to check if the name of the ECM network card meets expectations. The specific method is as follows:

1. Start capturing RIL logs using the `logcat -b all` command after booting, and capture logs for about 2-3 minutes.
2. Search for the regular expression `glt.+interface:|cdc_ether` in RIL log, and the network card names corresponding to the two must be the **same** in the search results. The following figure shows examples of inconsistent names. In this case, the following attributes need to be added:

```
ril.fibocom.NetifName=<cdc_name>    //<cdc_name> is the network card name found by
                                     "cdc_ether", and in the following picture it is
                                     "eth0"
```

Please refer to Chapter [How to Set System Properties?](#) for the method of adding attributes.



4.2.3 Check If Routing Is Correct (ECM)

4.2.3.1 Problem Description

If it is ECM dial-up and the Android UI interface displays a 4G icon but cannot access the internet, the problem may be caused by a conflict between the module network and the Android Ethernet network,

resulting in a failed routing setting.

This is generally more likely to occur in higher versions of Android (such as Android 13, 14, etc.), and can be confirmed via the following phenomenon:

1. The network card name recognized by Android using `ipconfig` is "ethx". There is an IP address but the ping fails.

```
D:\wu\cmdr $ adb shell
android:/ $ ifconfig
lo          Link encap:Local Loopback
            inet addr:127.0.0.1  Mask:255.0.0.0
            inet6 addr: ::1/128 Scope: Host
            UP LOOPBACK RUNNING  MTU:65536  Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 TX bytes:0

eth1        Link encap:Ethernet  HWaddr f0:4b:b3:b9:eb:e5  Driver cdc_ether
            inet addr:10.233.43.209 Bcast:10.233.43.255  Mask:255.255.255.0
            inet6 addr: fe80::ed38:8bfb:3de:17b1/64 Scope: Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:28 errors:0 dropped:0 overruns:0 frame:0
            TX packets:48 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:7071 TX bytes:6013

android:/ $ ping 8.8.8.8
connect: Network is unreachable
2|android:/ $ |
```

2. The routing result queried using "`netstat -r`" does not contain the keyword "default", but when the "`net.xx.gw`" property queried using "`getprop | grep net`" contains a gateway address.

```
2|android:/ $ netstat -r
Kernel IP routing table
Destination      Gateway          Genmask          Flags      MSS Window  irtt Iface
10.233.43.0      *                255.255.255.0    U           0 0        0 eth1
android:/ $ getprop | grep net
[gsm.network.type]: [LTE]
[init.svc.netd]: [running]
[init.svc.debug_pid.netd]: [451]
[net.bt.name]: [Android]
[net.eth1.dns1]: [211.137.130.2]
[net.eth1.dns2]: [211.137.130.18]
[net.eth1.gw]: [10.233.43.1]
[net.eth1.local-ip]: [10.233.43.209]
[net.tcp_def_init_rwnd]: [60]
[persist.netd.stable_secret]: [60e2:ba12:5ab0:4674:211b:9a0b:9b8d:97bb]
[ro.boottime.netd]: [9813101712]
[ro.rk.ethernet_settings]: [true]
```

3. After successfully setting the default gateway with the following "`ip route`" command, the "default" keyword can be seen in the "`netstat - r`" result, and network access succeeds as expected.

```
su
```

```
ip route add default via <gw> dev <netif> //<gw> is the IPV4 gateway address in
// the "net.xx.gw" property.
// <netif> is the name of the current
//network card, as shown in the figure
//above as "eth1"
```

```
[persist.netd.stable_secret]: [60e2:ba12:5ab0:4674:211b:9a0b:9b8d:97bb]
[ro.boottime.netd]: [9813101712]
[ro.rk.ethernet_settings]: [true]
[ro.telephony.default_network]: [9]
android:/ $ su
android:/ # ip route add default via 10.233.43.1 dev eth1
android:/ # netstat -r
Kernel IP routing table
Destination      Gateway          Genmask         Flags   MSS Window  irtt Iface
default          10.233.43.1     0.0.0.0         UG      0 0        0 eth1
10.233.43.0      *               255.255.255.0   U        0 0        0 eth1
android:/ # ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=51 time=142 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=51 time=77.3 ms
^C
--- 8.8.8.8 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 77.379/109.926/142.474/32.549 ms
android:/ #
```

4.2.3.2 Solution

If all the above phenomena occur, change the network card name recognized by the module to a non "ethx" name (such as "lte") according to Chapter [Modifying the Network Card Name Used by the ECM](#) to avoid conflicts with Ethernet.

4.2.4 Check Files ip-up and ip-down (PPP)

If it is PPP dialing, write the ip-up and ip-down files as follows and place them in the `/system/etc/ppp` directory.

4.2.4.1 ip-up

```
#!/system/bin/sh

/system/bin/setprop "net.interfaces.defaultroute" "ppp0"

/system/bin/setprop "net.ppp0.dns1" "$DNS1"

/system/bin/setprop "net.ppp0.dns2" "$DNS2"

/system/bin/setprop "net.ppp0.local-ip" "$IPLOCAL"

/system/bin/setprop "net.ppp0.remote-ip" "$IPREMOTE"
```

```
exit 0
```

4.2.4.2 ip-down

```
#!/system/bin/sh
case $1 in
ppp0)
echo 0 > /proc/sys/net/ipv4/ip_forward;
;;
esac
rm /etc/ppp/ppp*.pid
# Use interface name if linkname is not available
NAME=${LINKNAME:-"$1"}
/system/bin/setprop "net.$NAME.local-ip" ""
/system/bin/setprop "net.$NAME.remote-ip" ""

/system/bin/setprop "net.interfaces.defaultroute" ""
/system/bin/setprop "net.ppp0.dns1" ""
/system/bin/setprop "net.ppp0.dns2" ""
/system/bin/setprop "net.ppp0.local-ip" ""
/system/bin/setprop "net.ppp0.remote-ip" ""
```

4.2.4.3 Modify Permissions

Use the following commands to change the execution permissions of ip-up and ip-down to 555:

```
chmod 555 /system/etc/ppp/ip-up
chmod 555 /system/etc/ppp/ip-down
```

4.2.5 Capture Log for Fibocom to Analyze

If you still cannot access the internet after following the above steps, capture the log and contact Fibocom technical personnel (please refer to Section [Capturing RIL LOG](#) for the method of log capture).

4.3 Unable to Make a Call

Check whether the SIM card supports the voice function. If so, confirm with Fibocom technical support personnel whether the module supports the call function under the current network standard.

4.4 Unable to Send SMS

Check whether the SIM card supports the SMS function. If so, confirm with Fibocom technical support personnel whether the module supports the SMS function under the current network standard.

4.5 How to Set APN

There are two ways to set APN: setting APN on the Android UI and modifying the `/system/etc/apns-conf.xml` file in the Android device. The details are as follows.

4.5.1 Set APN on UI Interface

Perform the following operations (for debugging only):

1. Install the module on the Android development board.
2. Insert a SIM card into the development board.
3. Power on the board and wait.
4. On the Android UI interface, choose "Settings ->Wireless and Network Cable (More) ->Mobile Network ->Access Point Name (APN)" and select the specified APN to modify, or click on the menu button in the upper-right corner and select "New APN" to add an APN. Enter **Name**, **APN**, **MCC**, **MNC**, etc., and click the menu button in the upper-right corner to save the settings.

4.5.2 Modify apns-conf.xml File

When modifying the `apns-conf.xml` file, add information according to the actual information of the SIM card. The main information to be added includes APN, MCC, MNC, user, password, type, protocol, etc. For example:

```
<apn carrier="China Mobile"
    apn="cmnet"
    mcc="460"
    mnc="04"
    user=""
    server=""
    password=""
    proxy=""
    port=""
    mmsproxy=""
    mmsport=""
```

```

mmSC=""
type="default,net,supl"
preferred="true"
localized_name="APN_NAME_CMNET"
protocol="IPV4V6"
roaming_protocol="IPV4V6"

```

```

/>

```

After updating the APN configuration list, **telephony.db** has been initialized because the Android device has been started up for many times. You must delete the database file and then start the device again to load the new configuration file. The delete command is as follows:

```
rm /data/user_de/0/com.android.providers.telephony/databases/telephony.db
```



The path to **telephony.db** may vary slightly on different devices, depending on the actual situation.

4.6 How to Modify Network Card Name Used by ECM

If you want to modify the network card name used by the ECM, you can modify the **kernel/drivers/net/USB/usbnet.c** file of the Android source code to match the network card name through VID/PID, as follows:

```

if ((dev->driver_info->flags & FLAG_ETHER) != 0 &&
    ((dev->driver_info->flags & FLAG_POINTTOPOINT) == 0 ||
     (net->dev_addr[0] & 0x02) == 0)) {
    #if 1
    if ((le16_to_cpu(
        xdev->descriptor.idVendor) == 0x19d2) &&
        (le16_to_cpu(
            xdev->descriptor.idProduct) == 0x0199))
        strcpy(net->name, "lte%d");
    else if ((le16_to_cpu(
        xdev->descriptor.idVendor) == 0x2CB7) &&
        (le16_to_cpu(
            xdev->descriptor.idProduct) == 0x0001))
        strcpy(net->name, "lte%d");
    else
    #endif
        strcpy(net->name, "eth%d");
}

```

"lte%d" in the figure above can be changed to other names, such as "usb%d", depending on your needs.

For Fibocom_Android_RIL_1.0.7 and earlier versions of RIL, after modifying the **usbnet.c** file, it is necessary to modify the network card name used by the RIL library at the same time, otherwise it may cause RIL dialing failure.

For example, if the network card name found by the **ifconfig -a** command after modification is "lte0", the following property need to be added (refer to Chapter [How to Set System Properties?](#) for modification

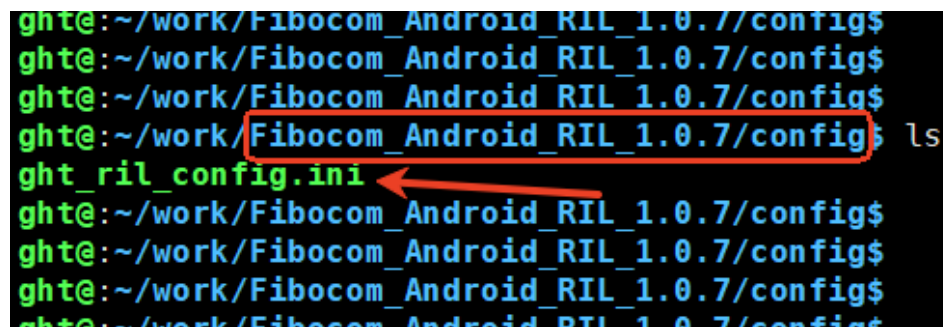
methods);

```
ril.fibocom.NetifName=lte0
```

This operation is not required for RIL versions later than Fibocom_Android_RIL_1.0.7.

4.7 ght_ril_config.ini File

The **ght_ril_config.ini** file is used to configure some options of RIL, such as dialing mode, USBMODE, etc. It is suitable for RIL versions Fibocom_Android_RIL_1.0.6 and later. The path of the file in the RIL version folder is as follows.

A terminal window screenshot with a black background and green text. The prompt is 'ght@:~/work/Fibocom_Android_RIL_1.0.7/config\$'. The user enters 'ls' and the output is 'ght_ril_config.ini'. A red rectangle highlights the path '~/work/Fibocom_Android_RIL_1.0.7/config' and a red arrow points from this rectangle to the file name 'ght_ril_config.ini' in the output.

This file takes effect when placed in the **/system/etc** directory on an Android device (For Fibocom_Android_RIL_1.0.0.13 and later RIL versions, this file can be placed in the **/vendor/etc** or **/system/etc** path, with the **/vendor/etc** path taking precedence, and it is recommended to place it in **/vendor/etc**).

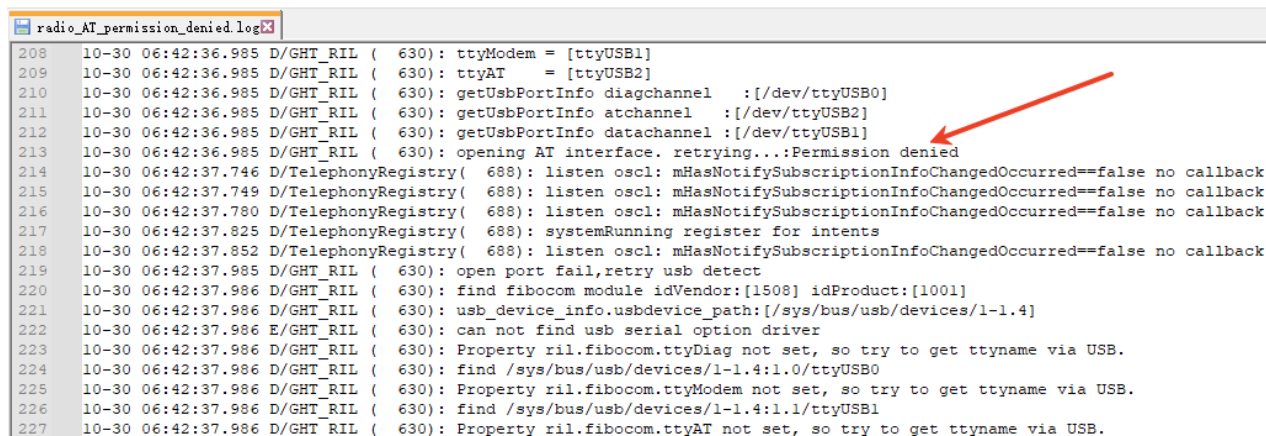


Please use this configuration file under the guidance of Fibocom technical support.

4.8 'Permission denied' Appears in RIL Log

4.8.1 Problem Description

If no AT command is sent in the RIL log and "**Permission denied**" is printed, it indicates that RIL currently does not have permission to operate the AT port and cannot send or receive AT commands.



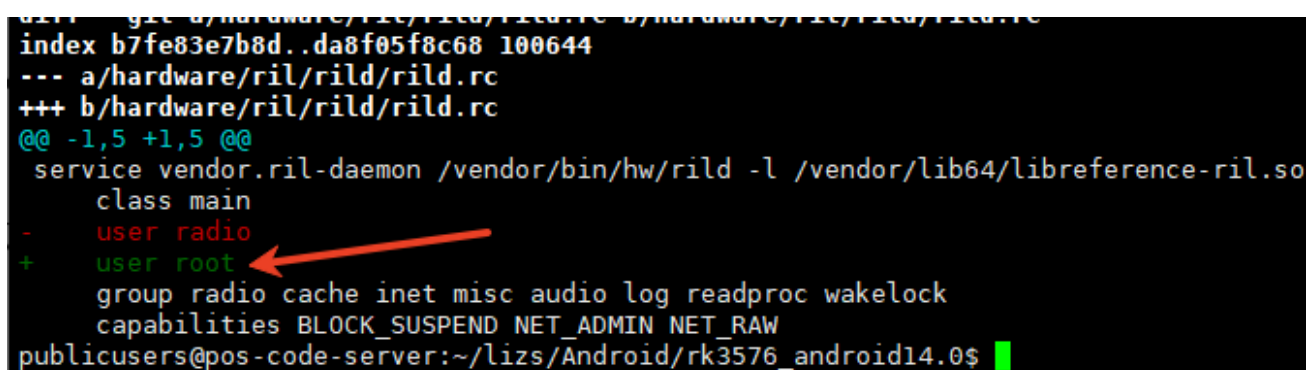
```

208 10-30 06:42:36.985 D/GHT_RIL ( 630): ttyModem = [ttyUSB1]
209 10-30 06:42:36.985 D/GHT_RIL ( 630): ttyAT = [ttyUSB2]
210 10-30 06:42:36.985 D/GHT_RIL ( 630): getUsbPortInfo diagchannel :[/dev/ttyUSB0]
211 10-30 06:42:36.985 D/GHT_RIL ( 630): getUsbPortInfo atchannel :[/dev/ttyUSB2]
212 10-30 06:42:36.985 D/GHT_RIL ( 630): getUsbPortInfo datachannel :[/dev/ttyUSB1]
213 10-30 06:42:36.985 D/GHT_RIL ( 630): opening AT interface. retrying...:Permission denied
214 10-30 06:42:37.746 D/TelephonyRegistry( 688): listen oscl: mHasNotifySubscriptionInfoChangedOccurred==false no callback
215 10-30 06:42:37.749 D/TelephonyRegistry( 688): listen oscl: mHasNotifySubscriptionInfoChangedOccurred==false no callback
216 10-30 06:42:37.780 D/TelephonyRegistry( 688): listen oscl: mHasNotifySubscriptionInfoChangedOccurred==false no callback
217 10-30 06:42:37.825 D/TelephonyRegistry( 688): systemRunning register for intents
218 10-30 06:42:37.852 D/TelephonyRegistry( 688): listen oscl: mHasNotifySubscriptionInfoChangedOccurred==false no callback
219 10-30 06:42:37.985 D/GHT_RIL ( 630): open port fail,retry usb detect
220 10-30 06:42:37.986 D/GHT_RIL ( 630): find fibocom module idVendor:[1508] idProduct:[1001]
221 10-30 06:42:37.986 D/GHT_RIL ( 630): usb_device_info.usbdevice_path:[/sys/bus/usb/devices/l-1.4]
222 10-30 06:42:37.986 E/GHT_RIL ( 630): can not find usb serial option driver
223 10-30 06:42:37.986 D/GHT_RIL ( 630): Property ril.fibocom.ttyDiag not set, so try to get ttyname via USB.
224 10-30 06:42:37.986 D/GHT_RIL ( 630): find /sys/bus/usb/devices/l-1.4:1.0/ttyUSB0
225 10-30 06:42:37.986 D/GHT_RIL ( 630): Property ril.fibocom.ttyModem not set, so try to get ttyname via USB.
226 10-30 06:42:37.986 D/GHT_RIL ( 630): find /sys/bus/usb/devices/l-1.4:1.1/ttyUSB1
227 10-30 06:42:37.986 D/GHT_RIL ( 630): Property ril.fibocom.ttyAT not set, so try to get ttyname via USB.

```

4.8.2 Solution

Check if there is a keyword "user radio" in the RIL service (usually "service ril daemon" or "service vendor.ril daemon") in `init.rc` or `rild.rc`. If there is, change it to "user root".



```

index b7fe83e7b8d..da8f05f8c68 100644
--- a/hardware/ril/rild/rild.rc
+++ b/hardware/ril/rild/rild.rc
@@ -1,5 +1,5 @@
service vendor.ril-daemon /vendor/bin/hw/rild -l /vendor/lib64/libreference-ril.so
class main
- user radio
+ user root
group radio cache inet misc audio log readproc wakelock
capabilities BLOCK_SUSPEND NET_ADMIN NET_RAW
publicusers@pos-code-server:~/liza/Android/rk3576_android14.0$

```

The path of `init.rc` or `rild.rc` in Android source code is generally as follows:

```

hardware/ril/rild/rild.rc
device/xx/xx/init.rc

```

4.9 How to Set System Properties

Fibocom has many custom system properties that are used to control the dialing method and network card name of RIL, etc. You can modify the values of these properties to modify certain configurations.

4.9.1 Property Setting Methods

By adding properties and their corresponding values to the property file, these properties can be permanently saved. The path to the property file is as follows:

```

/system/build.prop    //❶ Recommended for Android 7 and earlier
/vendor/build.prop    //❷ Recommended for Android 8 and later versions

```

Taking the `/vendor/build.prop` file as an example, the following is an introduction to the specific steps of setting properties. For Android 7 and earlier devices, replace "vendor" with "system":

1. Run the **adb pull** command to pull the **/vendor/build.prop** file from the device.

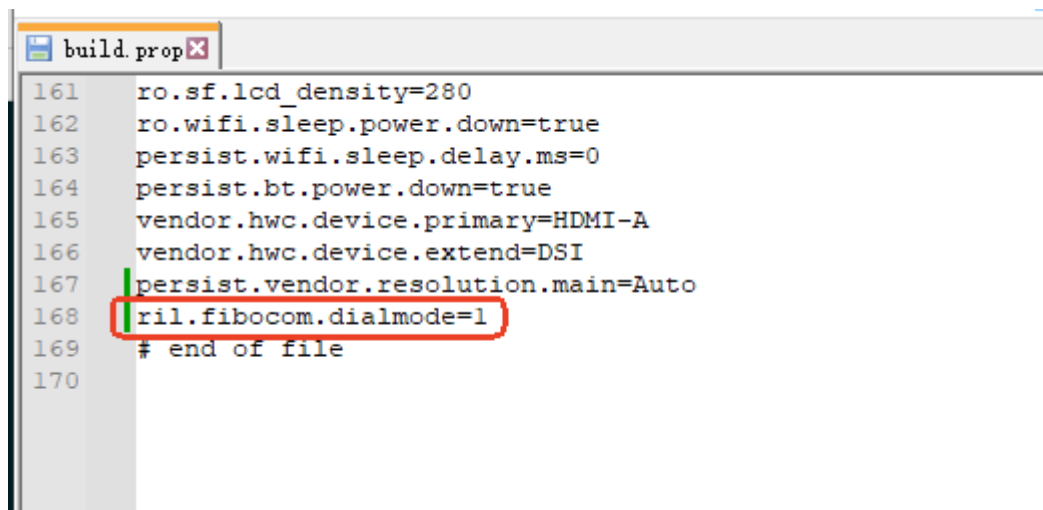
<code>adb root</code>	① Enter the root mode.
<code>adb remount</code>	② Re-mount the system partition to the read and write mode.
<code>adb pull /vendor/build.prop <LOCAL_PATH></code>	③ Pull the build.prop file.

```
$ adb root
restarting adbd as root

$ adb remount
AVB verification is disabled, disabling verity state may have no effect
Remounted /system as RW
Remounted /vendor as RW
Remounted /odm as RW
Remounted /system_dlkm as RW
Remounted /system_ext as RW
Remounted /vendor_dlkm as RW
Remounted /odm_dlkm as RW
Remounted /product as RW
Remount succeeded

$ adb pull /vendor/build.prop e:/log/
/vendor/build.prop: 1 file pulled, 0 skipped. 0.1 MB/s (5771 bytes in 0.038s)
```

2. Add the property (taking **ril.fibocom.dialmode** as an example) to the end of the **build.prop** file. Save the modification.



```
build.prop
161 ro.sf.lcd_density=280
162 ro.wifi.sleep.power.down=true
163 persist.wifi.sleep.delay.ms=0
164 persist.bt.power.down=true
165 vendor.hwc.device.primary=HDMI-A
166 vendor.hwc.device.extend=DSI
167 persist.vendor.resolution.main=Auto
168 ril.fibocom.dialmode=1
169 # end of file
170
```

3. Push the **build.prop** file to the original location in the device and restart the device.

<code>adb push d:/log/build.prop /vendor/build.prop</code>	① Push the build.prop file to the device again.
<code>adb reboot</code>	② Reboot the device.


```
$ adb push e:/log/build.prop /vendor/build.prop
e:/log/build.prop: 1 file pushed, 0 skipped. 2.0 MB/s (5794 bytes in 0.003s)
```

4. Run the **adb shell** command to enter the Android device. Then, you can run the **getprop|grep ril** command to view the set property. Example:

adb shell ❶ Enter the Andriod device.

getprop | grep ril ❷ Find the property related to RIL.

```
D:\wu\cmdex $
D:\wu\cmdex $
D:\wu\cmdex $ adb shell
rk3568_firefly_aioj:/ # getprop|grep ril
[init.svc.ril-daemon]: [running]
[init.svc.debug_pid.ril-daemon]: [11621]
[persist.ril.current_usbmode]: [32]
[ril.fibocom.dialmode]: [1]
[ril.fibocom.version]: [Fibocom_Android_RIL_V11X.06.V1.0.0.8]
[ril.function.dataonly]: [1]
[rild.libargs]: [-d]
[rild.libpath]: [/vendor/lib64/libquectel-ril.so]
[ro.boot.noril]: [false]
[ro.boottime.ril-daemon]: [6984883646]
[ro.ril.ecclist]: [112,911]
rk3568_firefly_aioj:/ #
```

The property that we just set ourselves

4.9.2 ril.fibocom.dialmode

Description

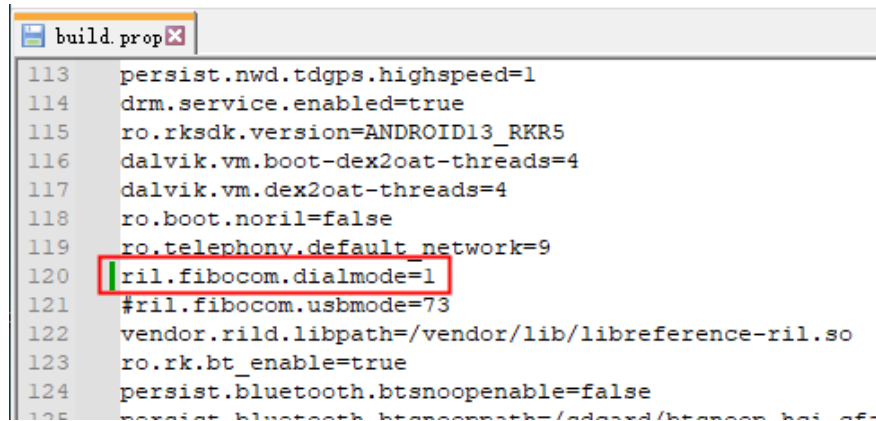
This property specifies whether the dial-up mode used by RIL is ECM or PPP.

Value

Table 5. Value range of dial-up mode

ril.fibocom.dialmode	Dial-up Mode
0	PPP
1 (Default)	ECM
3	QMI_WWAN

Example



```
build.prop
113 persist.nwd.tdgps.highspeed=1
114 drm.service.enabled=true
115 ro.rksdk.version=ANDROID13_RKR5
116 dalvik.vm.boot-dex2oat-threads=4
117 dalvik.vm.dex2oat-threads=4
118 ro.boot.noril=false
119 ro.telephony.default_network=9
120 ril.fibocom.dialmode=1
121 #ril.fibocom.usbmode=73
122 vendor.rild.libpath=/vendor/lib/libreference-ril.so
123 ro.rk.bt_enable=true
124 persist.bluetooth.btsnoopenable=false
125 persist.bluetooth.btsnooppath=/sdcard/btsnoop_hci.cfa
```

4.9.3 ril.fibocom.NetifName

Description

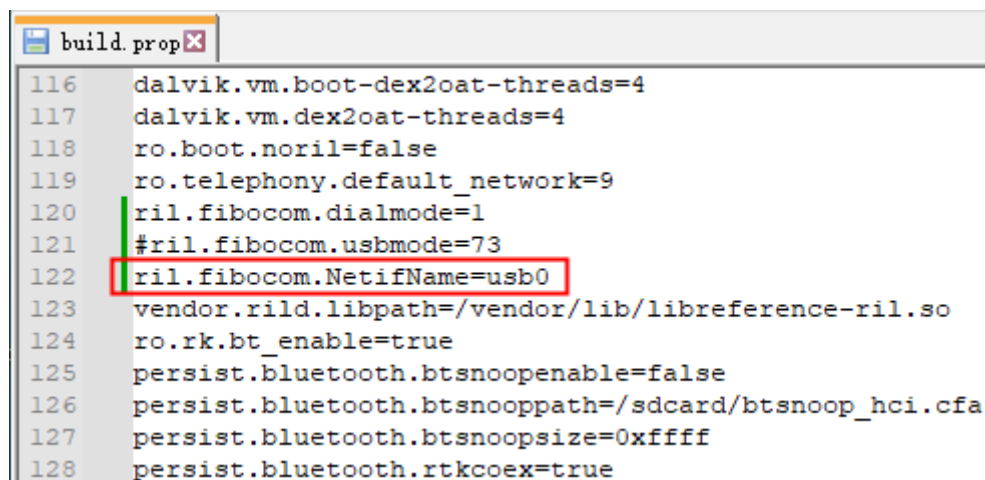
Customize the NIC used for RIL dial-up.

When RIL initiates dial-up, the sequence of NIC selection is as follows: NetifName specifies -> Default NIC (for example, usb0 or ppp0).

Value

The common values include usb0, eth0, and eth1.

Example



```
build.prop
116 dalvik.vm.boot-dex2oat-threads=4
117 dalvik.vm.dex2oat-threads=4
118 ro.boot.noril=false
119 ro.telephony.default_network=9
120 ril.fibocom.dialmode=1
121 #ril.fibocom.usbmode=73
122 ril.fibocom.NetifName=usb0
123 vendor.rild.libpath=/vendor/lib/libreference-ril.so
124 ro.rk.bt_enable=true
125 persist.bluetooth.btsnoopenable=false
126 persist.bluetooth.btsnooppath=/sdcard/btsnoop_hci.cfa
127 persist.bluetooth.btsnoopsize=0xffff
128 persist.bluetooth.rtkcoex=true
```

4.9.4 ril.fibocom.usbmode

Description

Modify the USB mode to a specified value.

Value

The value range is 0 and 17-75. If this property is not set or is set to 0, the USB mode of the module is not modified. At this time, there is no effect caused.

The default value is 0.

Example

```
build.prop
116 dalvik.vm.boot-dex2oat-threads=4
117 dalvik.vm.dex2oat-threads=4
118 ro.boot.noril=false
119 ro.telephony.default_network=9
120 ril.fibocom.dialmode=1
121 ril.fibocom.usbmode=0
122 ril.fibocom.NetifName=usb0
123 vendor.rild.libpath=/vendor/lib/libreference-ril.so
124 ro.rk.bt_enable=true
125 persist.bluetooth.btsnoopenable=false
```

4.10 How to Solve Empty Triangles

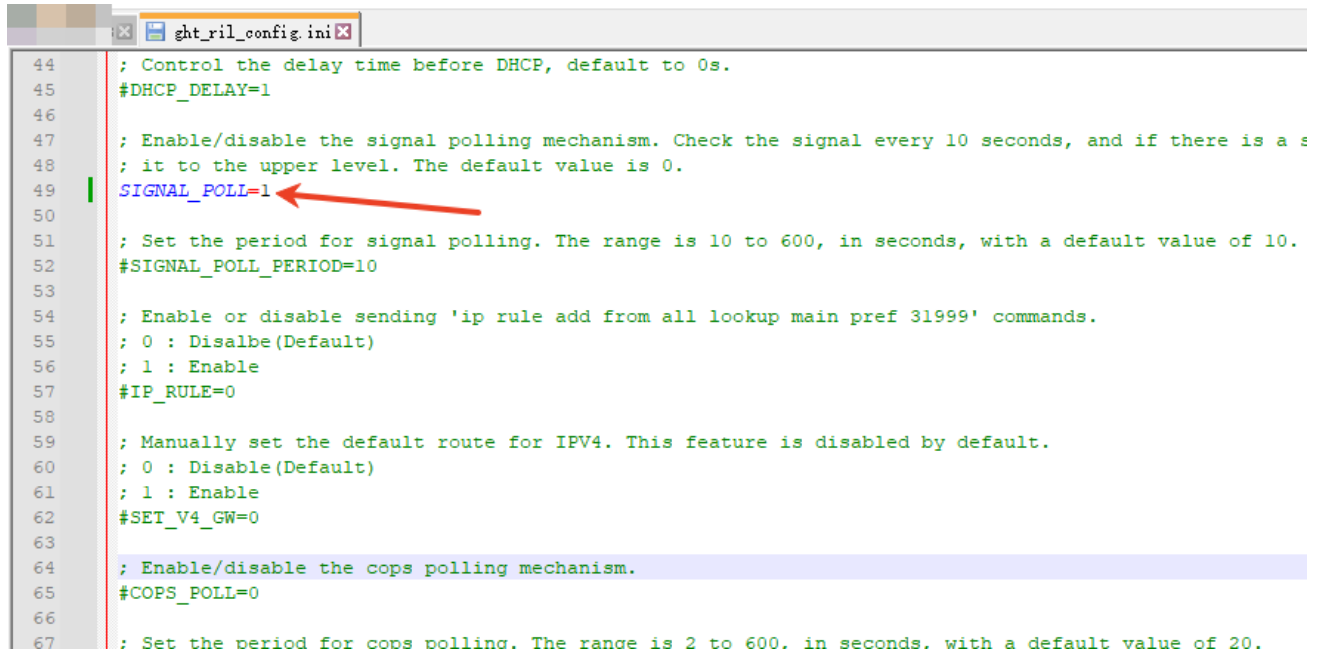
4.10.1 Problem Description

On higher versions of Android devices (such as Android 13 and 14), there may be a phenomenon where the data connection is normal but the signal icon displays as an empty triangle, as shown in the following figure.



4.10.2 Solution

For RIL versions Fibocom_Android_RIL_1.0.0.9 and later, change the value of **SIGNAL_POLL** in the [ght_ril_config.ini file](#) to 1 to solve this problem.



```
44 ; Control the delay time before DHCP, default to 0s.
45 #DHCP_DELAY=1
46
47 ; Enable/disable the signal polling mechanism. Check the signal every 10 seconds, and if there is a s
48 ; it to the upper level. The default value is 0.
49 SIGNAL_POLL=1
50
51 ; Set the period for signal polling. The range is 10 to 600, in seconds, with a default value of 10.
52 #SIGNAL_POLL_PERIOD=10
53
54 ; Enable or disable sending 'ip rule add from all lookup main pref 31999' commands.
55 ; 0 : Disalbe(Default)
56 ; 1 : Enable
57 #IP_RULE=0
58
59 ; Manually set the default route for IPV4. This feature is disabled by default.
60 ; 0 : Disable(Default)
61 ; 1 : Enable
62 #SET_V4_GW=0
63
64 ; Enable/disable the cops polling mechanism.
65 #COPS_POLL=0
66
67 ; Set the period for cops polling. The range is 2 to 600. in seconds. with a default value of 20.
```

The expected normal display after modification is shown in the following figure.



Appendix A: Dial Mode and USBMODE

Different dialing modes (such as PPP, ECM, or QMI_WWAN) correspond to different USBMODE. The specific relationship is as follows:

Table 6. Supported modules

Module Types	VID;PID	USBMODE		
		PPP	ECM	QMI_WWAN
L610	1782;4D10 1782;4D11	31	32	NA
MC61x: MC615, MC619	2CB7;0A01 2CBC;0A02 2CB7;0A03	31	32	NA
NL668: NL668-CN, NL668-JP	1508;1001	18	18	NA
MG110	1508;1001	18	18	NA
NL668T: NL668T-GL	2CA3;4009	NA	31	30
L71x: L716, L718	2CB7;0001 2CB7;0579	10 or 11	10	NA
MC919: MC919-CN	2CB7;0C01 2CB7;0C02 2CB7;0C03	72	70	NA
MC66x: MC660/MC661/MC665/MC669 MG66x: MG661	2CB7;0A0A 2CB7;0A0B 2CB7;0A0C	73	74	NA
FG132	2CB7;010B 2CB7;0112 2CB7;0113	NA	38	36
MA510	2CB7;0106 2CB7;010A	31 or 32	31 or 32	NA



"NA" indicates that the data dialing mode is not supported.

Appendix B: Acronyms and Abbreviations

Common terms and abbreviations used in RIL are listed in the following table.

Table 7. Terms and abbreviations

Acronym and Abbreviation	Full Name
GSM	Global System for Mobile Communications
VID	Vendor ID
PID	Product ID
RIL	Radio Interface Layer
RILD	Radio Interface Layer Daemon
APN	Access Point Name

Appendix C: Tags in Logs

Process logs are stored in the same file and they can be identified based on tags. The following table provides logs and their respective tags.

Table 8. Tags in the RIL log

Tag	Process
GHT-RIL	hardware/ril/reference-ril/reference-ril.c
GHT-AT	hardware/ril/reference-ril/atchannel.c
RILC	hardware/ril/libril/ril.cpp
RILD	hardware/ril/rild/rild.c
RILJ	frameworks/opt/telephony/src/java/com/android/internal/telephony/RIL.java