



Fibocom

PERFECT WIRELESS EXPERIENCE

RIL Log Analysis Guide

V1.3

Disclaimer

Customers shall design and develop their products according to the information provided in the document. Fibocom shall not be liable for any damages caused by failure to comply with the relevant operations, specifications or rules. Due to product version update or other reasons, Fibocom reserves the rights to modify any information in this document at any time without prior notice and without any responsibility. Unless otherwise specified, all the statements, information and suggestions contained in the document do not constitute any explicit or implicit guarantee.

Copyright Statement

Copyright © 2024 Fibocom Wireless Inc. All rights reserved.

Unless specifically authorized by Fibocom, the recipient of the document shall keep the received documents and information confidential and shall not use it for any purpose other than the implementation and development of this project. Without the prior written permission of the copyright holder, any company or individual is prohibited to excerpt, copy any part of or the entire document, or distribute the document in any form. For any violation of confidentiality obligations, unauthorized use or malicious use of the document and information in other illegal forms, Fibocom has the rights to pursue legal responsibility.

Trademark Statement

Fibocom The trademark is registered and owned by Fibocom Wireless Inc.

Other trademarks, product names, service names, and company names appearing in this document are the properties of their respective owners.

Contact

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

Address: Floor 10-14, Block A, Building 6, Shenzhen International Innovation Valley, Dashiye Road, Xili Community, Xili Street, Nanshan District, Shenzhen

Tel: 0755-26733555

Safety Instruction

Do not operate wireless communication products in areas where use of radio is not recommended without proper equipment certification. These areas include environment where radio interference may occur, such as flammable and explosive environment, medical equipment, aircraft, or any other equipment that may be subject to any form of radio interference.

Any driver of a vehicle must not operate a wireless communication product while controlling the vehicle. Doing so will reduce the driver's control and operation of the vehicle, posing a safety risk.

The wireless communication product does not guarantee a valid connection under any circumstances, for example, when the (U)SIM is in arrears or invalid. In case of emergency, use the emergency call function in power-on state, and make sure the equipment is in an area with sufficient signal strength.

Contents

Applicable Models	3
Change History	4
1 Overview.....	5
1.1 Log Capture Method.....	5
1.2 LOG TAG (Log label).....	5
1.3 Querying the RIL Version.....	7
2 Module Identification Anomalies	8
2.1 Checking Whether VID&PID Is Identified	8
2.2 Checking Whether the ttyUSB Interface Is Identified	10
2.3 Checking Whether the Upper Layer Delivers Requests.....	11
2.4 Checking Whether the AT Commands Are Sent	13
3 Abnormal Calling Service	15
3.1 Checking Whether the Module Supports the Calling Service	15
3.2 Checking the Calling APP Status	16
3.2.1 Checking Whether There Is a Calling Permission	16
3.2.2 Checking Whether the Calling Function Is Enabled on the Device	16
4 Abnormal SMS Service.....	17
4.1 Checking Whether the Module Supports the SMS Service	17
4.2 Checking the SMS APP Status.....	18
4.2.1 Checking Whether There Is a SMS Sending Permission	18
4.2.2 Checking Whether the SMS Function Is Enabled.....	18
5 Abnormal Data Dial-up.....	19
5.1 Checking Whether the Dial-up Request Is Successful.....	19
5.1.1 ECM Dial-up	19
5.1.2 PPP Dial-up	20
5.2 Checking Whether the APN Is Normal	21
5.3 Checking Whether the Module Registers with the PS Domain.....	22
5.4 Checking Whether the Network Registration Is Successful.....	23
5.4.1 Signal Strength	23

5.4.2 SIM Card Detection.....24

5.4.3 Network System.....24

6 Accessories.....26

6.1 Modification on the AP Side26

Applicable Models

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

Change History

V1.3 (2024-03-05)	Add "Applicable Models"
V1.2 (2023-11-15)	Removed Chinese characters from figures.
V1.1 (2023-09-06)	Changed the time in copyright statement from 2022 to 2023. Updated the log capture methods and LOG TAG.
V1.0 (2021-06-24)	Initial version

1 Overview

If the customer's host computer uses an Android system and there are problems such as failure to access the Internet, failure to make phone calls, or failure to send SMS messages, you need to capture the RIL logs for analysis. The following section describes the log capture and analysis method.

1.1 Log Capture Method

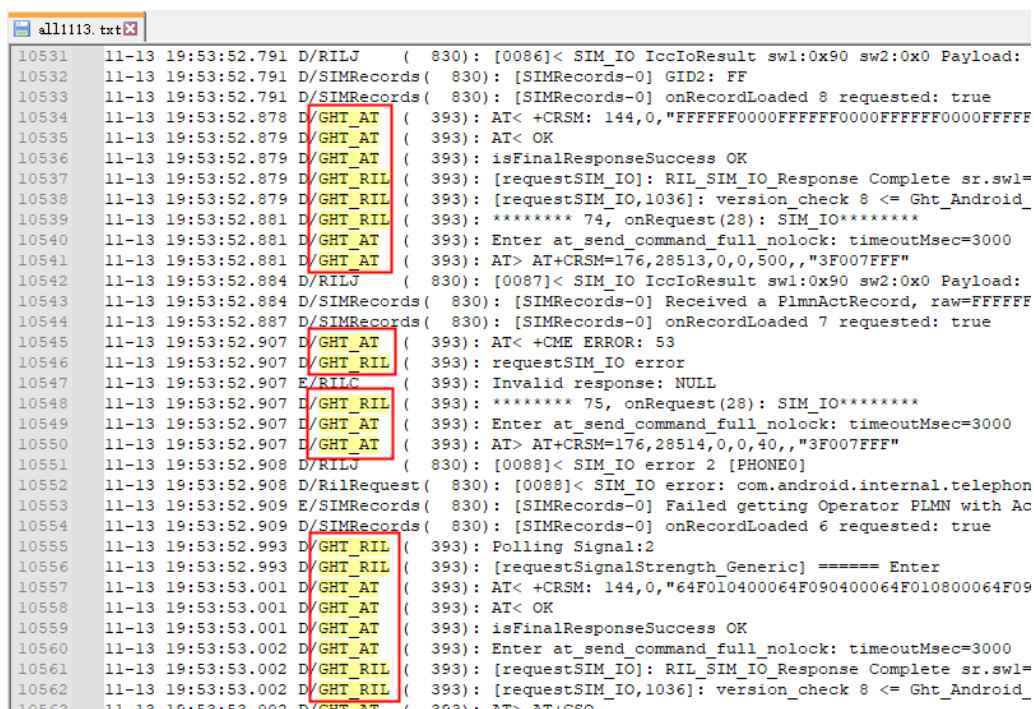
The RIL log capture method is as follows:

```
adb logcat -b all > xx
```

1.2 LOG TAG (Log label)

Logs output by different APPs are distinguished by LOG TAG. In the RIL log, you can search for "GHT_RIL" to get the information output by the RIL, or search for "GHT_AT" to get the AT commands sent and received by the RIL.

You can also search for the regular expression "GHT_RIL | GHT_AT" to get all the information of the RIL and AT at the same time. See Figure 1.



```

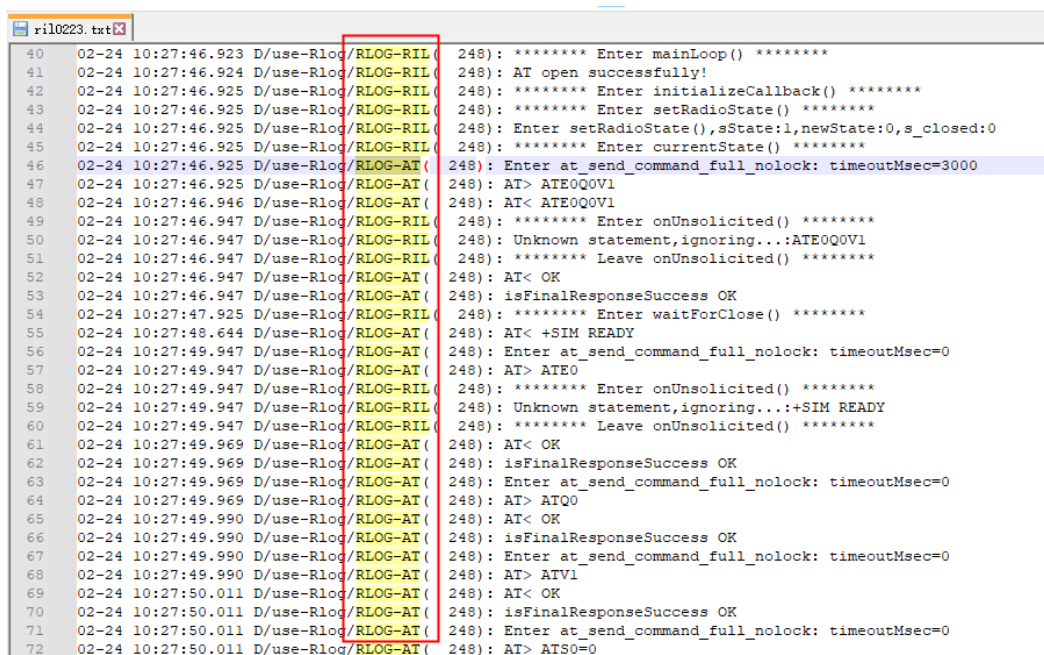
10531 11-13 19:53:52.791 D/RILJ ( 830): [0086]< SIM_IO IccIoResult sw1:0x90 sw2:0x0 Payload:
10532 11-13 19:53:52.791 D/SIMRecords( 830): [SIMRecords-0] GID2: FF
10533 11-13 19:53:52.791 D/SIMRecords( 830): [SIMRecords-0] onRecordLoaded 8 requested: true
10534 11-13 19:53:52.878 D/GHT_AT ( 393): AT< +CRSM: 144,0,"FFFFFF0000FFFFFF0000FFFFFF0000FFFFF
10535 11-13 19:53:52.879 D/GHT_AT ( 393): AT< OK
10536 11-13 19:53:52.879 D/GHT_AT ( 393): isFinalResponseSuccess OK
10537 11-13 19:53:52.879 D/GHT_RIL ( 393): [requestSIM_IO]: RIL_SIM_IO_Response Complete sr.sw1=
10538 11-13 19:53:52.879 D/GHT_RIL ( 393): [requestSIM_IO,1036]: version_check 8 <= Ght_Android_
10539 11-13 19:53:52.881 D/GHT_RIL ( 393): ***** 74, onRequest(28): SIM_IO*****
10540 11-13 19:53:52.881 D/GHT_AT ( 393): Enter at_send_command_full_nolock: timeoutMsec=3000
10541 11-13 19:53:52.881 D/GHT_AT ( 393): AT> AT+CRSM=176,28513,0,0,500,,,"3F007FFF"
10542 11-13 19:53:52.884 D/RILJ ( 830): [0087]< SIM_IO IccIoResult sw1:0x90 sw2:0x0 Payload:
10543 11-13 19:53:52.884 D/SIMRecords( 830): [SIMRecords-0] Received a PlmnActRecord, raw=FFFFFFF
10544 11-13 19:53:52.887 D/SIMRecords( 830): [SIMRecords-0] onRecordLoaded 7 requested: true
10545 11-13 19:53:52.907 D/GHT_AT ( 393): AT< +CME ERROR: 53
10546 11-13 19:53:52.907 D/GHT_RIL ( 393): requestSIM_IO error
10547 11-13 19:53:52.907 E/RILJ ( 393): Invalid response: NULL
10548 11-13 19:53:52.907 D/GHT_RIL ( 393): ***** 75, onRequest(28): SIM_IO*****
10549 11-13 19:53:52.907 D/GHT_AT ( 393): Enter at_send_command_full_nolock: timeoutMsec=3000
10550 11-13 19:53:52.907 D/GHT_AT ( 393): AT> AT+CRSM=176,28514,0,0,40,,,"3F007FFF"
10551 11-13 19:53:52.908 D/RILJ ( 830): [0088]< SIM_IO error 2 [PHONE0]
10552 11-13 19:53:52.908 D/RilRequest( 830): [0088]< SIM_IO error: com.android.internal.telephon
10553 11-13 19:53:52.909 E/SIMRecords( 830): [SIMRecords-0] Failed getting Operator PLMN with Ac
10554 11-13 19:53:52.909 D/SIMRecords( 830): [SIMRecords-0] onRecordLoaded 6 requested: true
10555 11-13 19:53:52.993 D/GHT_RIL ( 393): Polling Signal:2
10556 11-13 19:53:52.993 D/GHT_RIL ( 393): [requestSignalStrength_Generic] ===== Enter
10557 11-13 19:53:53.001 D/GHT_AT ( 393): AT< +CRSM: 144,0,"64F010400064F090400064F010800064F09
10558 11-13 19:53:53.001 D/GHT_AT ( 393): AT< OK
10559 11-13 19:53:53.001 D/GHT_AT ( 393): isFinalResponseSuccess OK
10560 11-13 19:53:53.002 D/GHT_AT ( 393): Enter at_send_command_full_nolock: timeoutMsec=3000
10561 11-13 19:53:53.002 D/GHT_RIL ( 393): [requestSIM_IO]: RIL_SIM_IO_Response Complete sr.sw1=
10562 11-13 19:53:53.002 D/GHT_RIL ( 393): [requestSIM_IO,1036]: version_check 8 <= Ght_Android_
10563 11-13 19:53:53.002 D/GHT_AT ( 393): AT> AT+CRSM=

```

Figure 1. "GHT_RIL|GHT_AT" search result



Very few RIL libraries have older versions, and the LOG TAG is "RLOG-RIL" and "RLOG-AT" (as shown in Figure 2), or "RIL" and "AT" (as shown in Figure 3), which is expected.



```

40 02-24 10:27:46.923 D/use-Rlog/RLOG-RIL 248): ***** Enter mainLoop() *****
41 02-24 10:27:46.924 D/use-Rlog/RLOG-RIL 248): AT open successfully!
42 02-24 10:27:46.925 D/use-Rlog/RLOG-RIL 248): ***** Enter initializeCallback() *****
43 02-24 10:27:46.925 D/use-Rlog/RLOG-RIL 248): ***** Enter setRadioState() *****
44 02-24 10:27:46.925 D/use-Rlog/RLOG-RIL 248): Enter setRadioState(),sState:1,newState:0,s_closed:0
45 02-24 10:27:46.925 D/use-Rlog/RLOG-RIL 248): ***** Enter currentState() *****
46 02-24 10:27:46.925 D/use-Rlog/RLOG-AT ( 248): Enter at_send_command_full_nolock: timeoutMsec=3000
47 02-24 10:27:46.925 D/use-Rlog/RLOG-AT ( 248): AT> ATE0Q0V1
48 02-24 10:27:46.946 D/use-Rlog/RLOG-AT ( 248): AT< ATE0Q0V1
49 02-24 10:27:46.947 D/use-Rlog/RLOG-RIL 248): ***** Enter onUnsolicited() *****
50 02-24 10:27:46.947 D/use-Rlog/RLOG-RIL 248): Unknown statement,ignoring...:ATE0Q0V1
51 02-24 10:27:46.947 D/use-Rlog/RLOG-RIL 248): ***** Leave onUnsolicited() *****
52 02-24 10:27:46.947 D/use-Rlog/RLOG-AT ( 248): AT< OK
53 02-24 10:27:46.947 D/use-Rlog/RLOG-AT ( 248): isFinalResponseSuccess OK
54 02-24 10:27:47.925 D/use-Rlog/RLOG-RIL 248): ***** Enter waitForClose() *****
55 02-24 10:27:48.644 D/use-Rlog/RLOG-AT ( 248): AT< +SIM READY
56 02-24 10:27:49.947 D/use-Rlog/RLOG-AT ( 248): Enter at_send_command_full_nolock: timeoutMsec=0
57 02-24 10:27:49.947 D/use-Rlog/RLOG-AT ( 248): AT> ATE0
58 02-24 10:27:49.947 D/use-Rlog/RLOG-RIL 248): ***** Enter onUnsolicited() *****
59 02-24 10:27:49.947 D/use-Rlog/RLOG-RIL 248): Unknown statement,ignoring...:SIM READY
60 02-24 10:27:49.947 D/use-Rlog/RLOG-RIL 248): ***** Leave onUnsolicited() *****
61 02-24 10:27:49.969 D/use-Rlog/RLOG-AT ( 248): AT< OK
62 02-24 10:27:49.969 D/use-Rlog/RLOG-AT ( 248): isFinalResponseSuccess OK
63 02-24 10:27:49.969 D/use-Rlog/RLOG-AT ( 248): Enter at_send_command_full_nolock: timeoutMsec=0
64 02-24 10:27:49.969 D/use-Rlog/RLOG-AT ( 248): AT> ATQ0
65 02-24 10:27:49.990 D/use-Rlog/RLOG-AT ( 248): AT< OK
66 02-24 10:27:49.990 D/use-Rlog/RLOG-AT ( 248): isFinalResponseSuccess OK
67 02-24 10:27:49.990 D/use-Rlog/RLOG-AT ( 248): Enter at_send_command_full_nolock: timeoutMsec=0
68 02-24 10:27:49.990 D/use-Rlog/RLOG-AT ( 248): AT> ATV1
69 02-24 10:27:50.011 D/use-Rlog/RLOG-AT ( 248): AT< OK
70 02-24 10:27:50.011 D/use-Rlog/RLOG-AT ( 248): isFinalResponseSuccess OK
71 02-24 10:27:50.011 D/use-Rlog/RLOG-AT ( 248): Enter at_send_command_full_nolock: timeoutMsec=0
72 02-24 10:27:50.011 D/use-Rlog/RLOG-AT ( 248): AT> AT$0=0

```

Figure 2. "RLOG-RIL|RLOG-AT" search result

```

ril0609.txt
58 06-10 15:13:06.456 D/AT ( 243): AT< +SIM READY
59 06-10 15:13:07.127 D/TelephonyManager( 504): /proc/cmdline=earlycon=uart8250,mmio32,0xffla0000 swiotl
60 06-10 15:13:08.250 D/AT ( 243): End sleep for 3 seconds.
61 06-10 15:13:08.250 D/AT ( 243): *****Leave at_handshake*****
62 06-10 15:13:08.250 D/AT ( 243): Enter at_send_command_full_nolock: timeoutMsec=10000
63 06-10 15:13:08.250 D/AT ( 243): AT> ATE0
64 06-10 15:13:08.250 D/RIL ( 243): ***** Enter onUnsolicited() *****
65 06-10 15:13:08.250 D/RIL ( 243): [onUnsolicited,2846]Report SIM Inserted, sState:0(RADIO_OFF)
66 06-10 15:13:08.251 D/RIL ( 243): ***** Leave onUnsolicited() *****
67 06-10 15:13:08.271 D/AT ( 243): AT< OK
68 06-10 15:13:08.272 D/AT ( 243): isFinalResponseSuccess OK
69 06-10 15:13:08.272 D/AT ( 243): Enter at_send_command_full_nolock: timeoutMsec=10000
70 06-10 15:13:08.272 D/AT ( 243): AT> ATQ0
71 06-10 15:13:08.292 D/AT ( 243): AT< OK
72 06-10 15:13:08.293 D/AT ( 243): isFinalResponseSuccess OK
73 06-10 15:13:08.293 D/AT ( 243): Enter at_send_command_full_nolock: timeoutMsec=10000
74 06-10 15:13:08.293 D/AT ( 243): AT> ATV1
75 06-10 15:13:08.314 D/AT ( 243): AT< OK
76 06-10 15:13:08.314 D/AT ( 243): isFinalResponseSuccess OK
77 06-10 15:13:08.314 D/AT ( 243): Enter at_send_command_full_nolock: timeoutMsec=10000
78 06-10 15:13:08.314 D/AT ( 243): AT> AT$0=0
79 06-10 15:13:08.335 D/AT ( 243): AT< OK
80 06-10 15:13:08.335 D/AT ( 243): isFinalResponseSuccess OK
81 06-10 15:13:08.335 D/AT ( 243): Enter at_send_command_full_nolock: timeoutMsec=10000
82 06-10 15:13:08.335 D/AT ( 243): AT> AT+CMEE=1
83 06-10 15:13:08.356 D/AT ( 243): AT< OK
84 06-10 15:13:08.356 D/AT ( 243): isFinalResponseSuccess OK
85 06-10 15:13:08.356 D/AT ( 243): Enter at_send_command_full_nolock: timeoutMsec=10000

```

Figure 3. "RIL|AT" search result

1.3 Querying the RIL Version

You can query the current RIL version from the RIL LOG, which is usually located at the beginning of the RIL LOG, under the "getVersion()" information. The following figure shows the log in the version of "L61x&MC61x_RIL_V7X.06.V1.0.0".

```

ril0223.txt
1 02-24 10:27:44.720 D/RILD ( 248): **RIL Daemon Started**
2 02-24 10:27:44.720 D/RILD ( 248): **RILD param count=6**
3 02-24 10:27:44.740 W/RILD ( 248): RIL_SAP_Init not defined or exported in /system/lib64/libreference
4 02-24 10:27:44.740 D/RILD ( 248): RIL_Init argc = 5 clientId = 0
5 02-24 10:27:44.740 D/use-Rlog/RLOG-RIL( 248): ***** Enter RIL_Init() *****
6 02-24 10:27:44.740 D/use-Rlog/RLOG-RIL( 248): setenforce 0
7 02-24 10:27:44.814 D/use-Rlog/RLOG-RIL( 248): ***** Enter getVersion() *****
8 02-24 10:27:44.815 D/use-Rlog/RLOG-RIL( 248): L61x&MC61x_RIL_V7X.06.V1.0.0
9 02-24 10:27:44.815 D/use-Rlog/RLOG-RIL( 248): uname -r > /data/ODM_KERNEL_VERSION
10 02-24 10:27:44.895 D/use-Rlog/RLOG-RIL( 248): ODM_KERNEL_VERSION :: 4.4.103
11 02-24 10:27:44.896 D/use-Rlog/RLOG-RIL( 248): 4.4.103
12 02-24 10:27:45.897 D/RIL ( 248): find fibocom module idVendor:[1782] idProduct:[4d11]
13 02-24 10:27:45.897 D/RIL ( 248): usb_device_info.usbdevice_path:[/sys/bus/usb/devices/3-1]
14 02-24 10:27:45.897 D/RIL ( 248): find usb serial option driver, but do not contain fibocom vid&pid
15 02-24 10:27:46.879 D/TelephonyManager( 200): No /proc/cmdline exception=java.io.FileNotFoundException:
16 02-24 10:27:46.879 D/TelephonyManager( 200): /proc/cmdline=
17 02-24 10:27:46.880 D/TelephonyManager( 201): No /proc/cmdline exception=java.io.FileNotFoundException:
18 02-24 10:27:46.880 D/TelephonyManager( 201): /proc/cmdline=

```

Figure 4. RIL version

2 Module Identification Anomalies

This chapter describes several methods to check whether the module identification is abnormal.

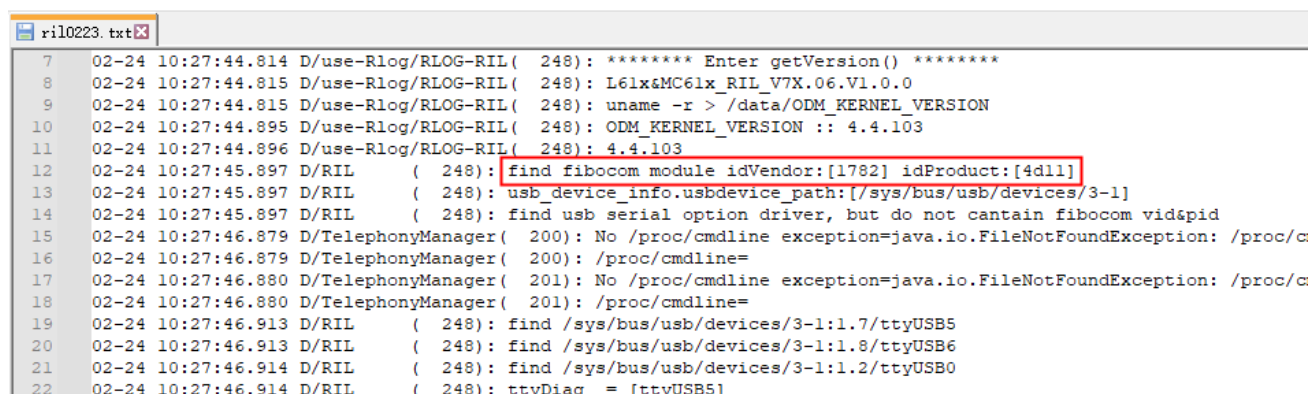
- Check whether VID&PID is identified.
- Check whether the ttyUSB interface is identified.
- Check whether the upper layer delivers requests.
- Check whether the AT commands are successfully sent.

If there are no problems in the above checks, the module is successfully identified.

2.1 Checking Whether VID&PID Is Identified

If the RIL successfully identifies the VID&PID of the module, it outputs the information in the RIL LOG.

For example, if the VID&PID of the current module is "1782 4d11", and the RIL successfully identifies the module, the following information will be output in the RIL log:



```

7 02-24 10:27:44.814 D/use-Rlog/RLOG-RIL( 248): ***** Enter getVersion() *****
8 02-24 10:27:44.815 D/use-Rlog/RLOG-RIL( 248): L61x&MC61x_RIL_V7X.06.V1.0.0
9 02-24 10:27:44.815 D/use-Rlog/RLOG-RIL( 248): uname -r > /data/ODM_KERNEL_VERSION
10 02-24 10:27:44.895 D/use-Rlog/RLOG-RIL( 248): ODM_KERNEL_VERSION :: 4.4.103
11 02-24 10:27:44.896 D/use-Rlog/RLOG-RIL( 248): 4.4.103
12 02-24 10:27:45.897 D/RIL ( 248): find fibocom module idVendor:[1782] idProduct:[4d11]
13 02-24 10:27:45.897 D/RIL ( 248): usb_device_info.usbdevice_path:[/sys/bus/usb/devices/3-1]
14 02-24 10:27:45.897 D/RIL ( 248): find usb serial option driver, but do not contain fibocom vid&pid
15 02-24 10:27:46.879 D/TelephonyManager( 200): No /proc/cmdline exception=java.io.FileNotFoundException: /proc/c
16 02-24 10:27:46.879 D/TelephonyManager( 200): /proc/cmdline=
17 02-24 10:27:46.880 D/TelephonyManager( 201): No /proc/cmdline exception=java.io.FileNotFoundException: /proc/c
18 02-24 10:27:46.880 D/TelephonyManager( 201): /proc/cmdline=
19 02-24 10:27:46.913 D/RIL ( 248): find /sys/bus/usb/devices/3-1:1.7/ttyUSB5
20 02-24 10:27:46.913 D/RIL ( 248): find /sys/bus/usb/devices/3-1:1.8/ttyUSB6
21 02-24 10:27:46.914 D/RIL ( 248): find /sys/bus/usb/devices/3-1:1.2/ttyUSB0
22 02-24 10:27:46.914 D/RIL ( 248): ttvDiaa = [ttvUSB5]

```

Figure 5. RIL successfully identified the VID&PID

If the module is connected to the Android device, but the RIL does not successfully identify the VID&PID, the RIL will keep trying to find the available device and output the following log:

```

ril0624.txt
1 06-24 12:06:36.991 D/RILD ( 287): **RIL Daemon Started**
2 06-24 12:06:36.991 D/RILD ( 287): **RILD param count=3**
3 06-24 12:06:37.029 W/RILD ( 287): RIL_SAP_Init not defined or exported in /vendor/lib64/libquctel-ril.so: undefined s
4 06-24 12:06:37.029 D/RILD ( 287): RIL_Init argc = 3 clientId = 0
5 06-24 12:06:37.029 D/use-Rlog/RLOG-RIL( 287): ***** Enter RIL_Init() *****
6 06-24 12:06:37.029 D/use-Rlog/RLOG-RIL( 287): setenforce 0
7 06-24 12:06:37.180 D/use-Rlog/RLOG-RIL( 287): ***** Enter getVersion() *****
8 06-24 12:06:37.181 D/use-Rlog/RLOG-RIL( 287): L61x&MC61x_RIL_V10X.06.V1.0.3
9 06-24 12:06:37.181 D/use-Rlog/RLOG-RIL( 287): uname -r > /data/ODM_KERNEL_VERSION
10 06-24 12:06:37.247 D/use-Rlog/RLOG-RIL( 287): ODM_KERNEL_VERSION ::
11 06-24 12:06:37.250 D/use-Rlog/RLOG-RIL( 287):
12 06-24 12:06:38.251 E/RIL ( 287): getUsbPortInfo error errnum:[-1]
13 06-24 12:06:38.251 D/RIL ( 287): please check usb port state,RIL do not detect the usb devices
14 06-24 12:06:38.403 D/TelephonyManager( 232): No /proc/cmdline exception=java.io.FileNotFoundException: /proc/cmdline: oper
15 06-24 12:06:38.403 D/TelephonyManager( 232): /proc/cmdline=
16 06-24 12:06:39.253 E/RIL ( 287): getUsbPortInfo error errnum:[-1]
17 06-24 12:06:39.253 D/RIL ( 287): please check usb port state,RIL do not detect the usb devices
18 06-24 12:06:40.255 E/RIL ( 287): getUsbPortInfo error errnum:[-1]
19 06-24 12:06:40.255 D/RIL ( 287): please check usb port state,RIL do not detect the usb devices
20 06-24 12:06:41.014 E/TelephonyManager( 445): IMSI error: Subscriber Info is null
21 06-24 12:06:41.139 D/TelephonyManager( 233): No /proc/cmdline exception=java.io.FileNotFoundException: /proc/cmdline: oper
22 06-24 12:06:41.139 D/TelephonyManager( 233): /proc/cmdline=
23 06-24 12:06:41.256 E/RIL ( 287): getUsbPortInfo error errnum:[-1]
24 06-24 12:06:41.257 D/RIL ( 287): please check usb port state,RIL do not detect the usb devices
25 06-24 12:06:42.258 E/RIL ( 287): getUsbPortInfo error errnum:[-1]
26 06-24 12:06:42.258 D/RIL ( 287): please check usb port state,RIL do not detect the usb devices
27 06-24 12:06:43.260 E/RIL ( 287): getUsbPortInfo error errnum:[-1]
28 06-24 12:06:43.260 D/RIL ( 287): please check usb port state,RIL do not detect the usb devices
29 06-24 12:06:44.262 E/RIL ( 287): getUsbPortInfo error errnum:[-1]
30 06-24 12:06:44.262 D/RIL ( 287): please check usb port state,RIL do not detect the usb devices
31 06-24 12:06:45.263 E/RIL ( 287): getUsbPortInfo error errnum:[-1]
32 06-24 12:06:45.264 D/RIL ( 287): please check usb port state,RIL do not detect the usb devices
33 06-24 12:06:46.157 D/TelephonyRegistrv( 445): listen oscl: mHasNotifvSubscriptionInfoChangedOccurred==false no callback

```

Figure 6. RIL unsuccessfully identified the VID&PID

In such a case, check whether the module is normally connected to the Android device. Run the following command to confirm:

```
adb shell
lsusb
```

```

rk3399_roc_pc_plus:/ $ exit
D:\wu\cmdex $ adb shell
rk3399_roc_pc_plus:/ $ lsusb
Bus 005 Device 001: ID 1d6b:0002
Bus 003 Device 001: ID 1d6b:0002
Bus 001 Device 001: ID 1d6b:0002
Bus 006 Device 001: ID 1d6b:0001
Bus 001 Device 002: ID 0bda:d723
Bus 004 Device 001: ID 1d6b:0003
Bus 002 Device 001: ID 1d6b:0001
rk3399_roc_pc_plus:/ $

```

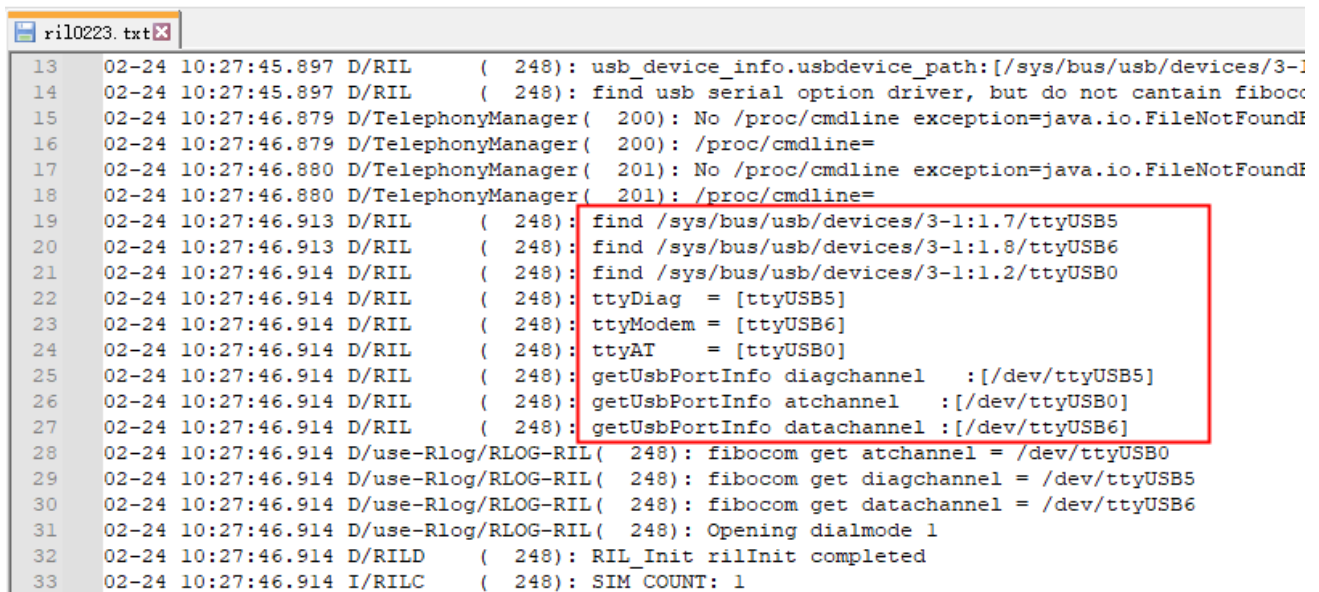
Figure 7. Checking whether the module is connected

If the result does not contain the module VID and PID, check whether the module is powered on and started normally. If necessary, seek assistant from hardware personnel.

If the result contains the module VID and PID, the module is normally connected to the Android device. Request the RIL personnel to check whether the RIL has been adapted to this module.

2.2 Checking Whether the ttyUSB Interface Is Identified

If the RIL identifies the ttyUSB interface correctly, the following information will be output in the RIL log to indicate which ttyUSB interface is used by Diag, Modem, and AT:



```
13 02-24 10:27:45.897 D/RIL ( 248): usb_device_info.usbdevice_path:[/sys/bus/usb/devices/3-1
14 02-24 10:27:45.897 D/RIL ( 248): find usb serial option driver, but do not contain fibocom
15 02-24 10:27:46.879 D/TelephonyManager( 200): No /proc/cmdline exception=java.io.FileNotFoundException
16 02-24 10:27:46.879 D/TelephonyManager( 200): /proc/cmdline=
17 02-24 10:27:46.880 D/TelephonyManager( 201): No /proc/cmdline exception=java.io.FileNotFoundException
18 02-24 10:27:46.880 D/TelephonyManager( 201): /proc/cmdline=
19 02-24 10:27:46.913 D/RIL ( 248): find /sys/bus/usb/devices/3-1:1.7/ttyUSB5
20 02-24 10:27:46.913 D/RIL ( 248): find /sys/bus/usb/devices/3-1:1.8/ttyUSB6
21 02-24 10:27:46.914 D/RIL ( 248): find /sys/bus/usb/devices/3-1:1.2/ttyUSB0
22 02-24 10:27:46.914 D/RIL ( 248): ttyDiag = [ttyUSB5]
23 02-24 10:27:46.914 D/RIL ( 248): ttyModem = [ttyUSB6]
24 02-24 10:27:46.914 D/RIL ( 248): ttyAT = [ttyUSB0]
25 02-24 10:27:46.914 D/RIL ( 248): getUsbPortInfo diagchannel :[/dev/ttyUSB5]
26 02-24 10:27:46.914 D/RIL ( 248): getUsbPortInfo atchannel :[/dev/ttyUSB0]
27 02-24 10:27:46.914 D/RIL ( 248): getUsbPortInfo datachannel :[/dev/ttyUSB6]
28 02-24 10:27:46.914 D/use-Rlog/RLOG-RIL( 248): fibocom get atchannel = /dev/ttyUSB0
29 02-24 10:27:46.914 D/use-Rlog/RLOG-RIL( 248): fibocom get diagchannel = /dev/ttyUSB5
30 02-24 10:27:46.914 D/use-Rlog/RLOG-RIL( 248): fibocom get datachannel = /dev/ttyUSB6
31 02-24 10:27:46.914 D/use-Rlog/RLOG-RIL( 248): Opening dialmode 1
32 02-24 10:27:46.914 D/RILD ( 248): RIL_Init rilInit completed
33 02-24 10:27:46.914 I/RILC ( 248): SIM_COUNT: 1
```

Figure 8. RIL successfully identified the ttyUSB interface

If the RIL does not identify the ttyUSB interface properly, it will report the following error information:


```

I PhoneFactory: Network Mode set to 9
D RILJ : RIL: init preferredNetworkType=9 cdmaSubscription=0 [PHONENull]
D TelephonyRegistry: notifyCellLocationForSubscriber: subId=0 cellLocation=null
D RIL : find fibocom module idVendor:[1782] idProduct:[4d10]
D RIL : usb_device_info.usbdevice_path:[/sys/bus/usb/devices/2-1]
D RIL : usb serial option driver match success
D RIL : find /sys/bus/usb/devices/2-1:1.5/ttyUSB3
D RIL : getUsbPortInfo error errnum:[-1]
D RIL : please check usb port state,RIL do not detect the usb devices
W CarrierConfigManager: Error getting config for subId -1 ICarrierConfigLoader is null
D TelephonyRegistry: listen oscl: mHasNotifySubscriptionInfoChangedOccurred==false no callback
D RIL : find fibocom module idVendor:[1782] idProduct:[4d10]
D RIL : usb_device_info.usbdevice_path:[/sys/bus/usb/devices/2-1]
D RIL : usb serial option driver match success
D RIL : find /sys/bus/usb/devices/2-1:1.5/ttyUSB3
D RIL : getUsbPortInfo error errnum:[-1]
D RIL : please check usb port state,RIL do not detect the usb d[evices
962 D R8IL : find fibocom module idVendor:[1782] idProduct:[4d10]
D RIL : usb_device_info.usbdevice_path:[/sys/bus/usb/devices/2-1]
D R7IL : usb serial option driver match success
D RIL : find /sys/bus/usb/devices/2-1:1.5/ttyUSB23
D RIL : getUsbPortInfo error errnum:[-1]
D RIL : please check usb port state,RIL do not detect the usb devices
D RIL : find fibocom module idVendor:[1782] idProduct:[4d10]
D RIL : usb_device_info.usbdevice_path:[/sys/bus/usb/devices/2-1]
D RIL : usb serial option driver match success
D RIL : find /sys/bus/usb/devices/2-1:1.5/ttyUSB3
D RIL : getUsbPortInfo error errnum:[-1]
D RIL : please check usb port state,RIL do not detect the usb d[evices
D RIL : find fibocom module idVendor:[1782] idProduct:[4d10]
D RIL : usb_device_info.usbdevice_path:[/sys/bus/usb/devices/e2-1]
D RIL : usb serial option driver match success
D RIL : find /sys/bus/usb/devices/2-1:1.5/ttyUSB3
D RIL : getUsbPortInfo error errnum:[-1]
D RIL : please check usb port state,RIL do not detect the usb devices
D RIL : find fibocom module idVendor:[1782] idProduct:[4d10]
D RIL : usb_device_info.usbdevice_path:[/sys/bus/usb/devices/2-1]
D RIL : usb serial option driver match success
D RIL : find /sys/bus/usb/devices/2-1:1.5/ttyUSB3
D RIL : getUsbPortInfo error errnum:[-1]
D RIL : please check usb port state,RIL do not detect the usb devices
D RIL : find fibocom module idVendor:[1782] idProduct:[4d10]
D RIL : usb_device_info.usbdevice_path:[/sys/bus/usb/devices/2-1]

```

Figure 9. RIL unsuccessfully identified the ttyUSB interface

This is a rare situation and may be due to differences between the customer's device and the equipment used by Fibocom for self-test. In this case, you can switch to a different USB mode and try again (for example, from 31 to 32, or vice versa), or request the RIL personnel to implement adaptation.

2.3 Checking Whether the Upper Layer Delivers Requests

If the RIL communicates with the upper layer (such as RILC and RILJ) normally, the upper layer will send various requests. The log displays the following information:

```

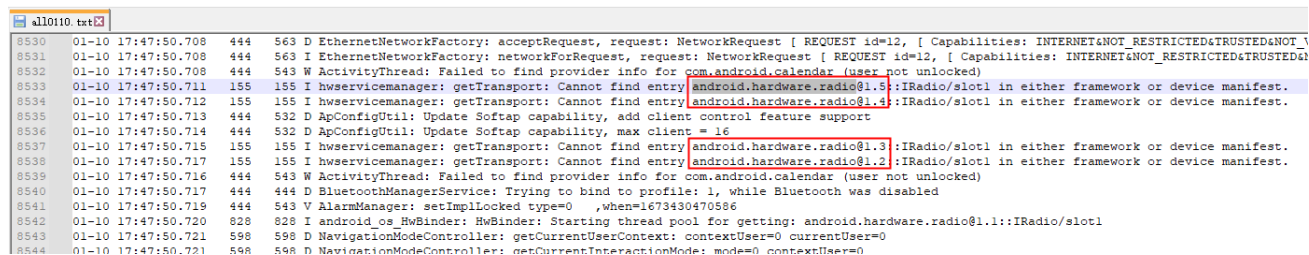
ril0524.txt
675 05-26 20:24:11.111 D/RIL ( 247): ***** 15, onRequest(38): GET_IMEI*****
676 05-26 20:24:11.111 D/RIL ( 247): [requestGetIMEI] ===== Enter
677 05-26 20:24:11.111 D/RIL ( 247): [getIMEI] cmd: [AT+CGSN?]
678 05-26 20:24:11.111 D/RIL ( 247): [getIMEI] prefix: [+CGSN:]
679 05-26 20:24:11.111 D/AT ( 247): Enter at_send_command_full_nolock: timeoutMsec=0
680 05-26 20:24:11.111 D/AT ( 247): AT> AT+CGSN?
681 05-26 20:24:11.132 D/AT ( 247): AT< +CGSN: "352273017386340"
682 05-26 20:24:11.133 D/AT ( 247): AT< OK
683 05-26 20:24:11.133 D/AT ( 247): isFinalResponseSuccess OK
684 05-26 20:24:11.133 D/RIL ( 247): [getIMEI] imei: 352273017386340
685 05-26 20:24:11.133 D/RIL ( 247): [requestGetIMEI] ===== Leave
686 05-26 20:24:11.133 D/RIL ( 247): ***** 16, onRequest(39): GET_IMEISV*****
687 05-26 20:24:11.133 D/RIL ( 247): [requestGetIMEISV] ===== Enter
688 05-26 20:24:11.133 D/RIL ( 247): [getIMEISV] cmd: [AT+CGSN=2]
689 05-26 20:24:11.133 D/RIL ( 247): [getIMEISV] prefix: [+CGSN:]
690 05-26 20:24:11.133 D/AT ( 247): Enter at_send_command_full_nolock: timeoutMsec=0
691 05-26 20:24:11.133 D/AT ( 247): AT> AT+CGSN=2
692 05-26 20:24:11.133 D/RILJ ( 717): [3663]< GET_IMEI [SUB0]
693 05-26 20:24:11.155 D/AT ( 247): AT< +CGSN: 3522730173863400
694 05-26 20:24:11.155 D/AT ( 247): AT< OK
695 05-26 20:24:11.155 D/AT ( 247): isFinalResponseSuccess OK
696 05-26 20:24:11.155 D/RIL ( 247): [getIMEISV] imei: 3522730173863400
697 05-26 20:24:11.155 D/RIL ( 247): requestGetIMEISV imeisv [00]
698 05-26 20:24:11.155 D/RIL ( 247): [requestGetIMEISV] ===== Leave
699 05-26 20:24:11.155 D/RIL ( 247): ***** 17, onRequest(130): RIL_REQUEST_GET_RADIO_CAPABILITY*****
700 05-26 20:24:11.155 D/RIL ( 247): [requestGetRadioCapability] ===== Enter
701 05-26 20:24:11.155 D/AT ( 247): Enter at_send_command_full_nolock: timeoutMsec=0
702 05-26 20:24:11.155 D/AT ( 247): AT> AT+GTRADIOCAP=0
703 05-26 20:24:11.155 D/RILJ ( 717): [3664]< GET_IMEISV [SUB0]
704 05-26 20:24:11.177 D/AT ( 247): AT< +CME ERROR: 58
705 05-26 20:24:11.177 E/RIL ( 247): [getRadioCap] at_send_command_singleline err
706 05-26 20:24:11.177 D/RIL ( 247): [setRadioCap] DEBUG mode flag:[9]
707 05-26 20:24:11.177 D/RIL ( 247): [setRadioCapability] get radioaccessfamily, rat[81926]
708 05-26 20:24:11.177 D/RIL ( 247): [requestGetRadioCapability] ===== Leave
709 05-26 20:24:11.177 D/RIL ( 247): Android Version [7.1.2], support requestGetRadioCapability request
710
711 05-26 20:24:11.177 D/RIL ( 247): ***** 18, onRequest(132): <unknown request>*****
712 05-26 20:24:11.177 D/RIL ( 247): ***** Enter currentState() *****
713 05-26 20:24:11.177 D/RIL ( 247): ***** 19, onRequest(108): VOICE_RADIO_TECH*****
714 05-26 20:24:11.177 D/RIL ( 247): [onRequest] VOICE_RADIO_TECH
715 05-26 20:24:11.178 D/RIL ( 247): ***** 20, onRequest(22): OPERATOR*****
716 05-26 20:24:11.178 D/AT ( 247): Enter at_send_command_full_nolock: timeoutMsec=0
717 05-26 20:24:11.178 D/AT ( 247): AT> AT+CPBS=3.0

```

Figure 10. "onRequest" in RIL log

If there is no result returned when you search for "onRequest", the upper-layer Radio service is not started normally. In this situation, capture the RIL log according to section [1.1](#) and search for "radio" to confirm.

Figure 11 shows the normal main log. For the android.hardware.radio service, search for the device from the latest supported version (as shown in Figure 11 1.5) to the earliest version, and stop the search when an available service is found. As shown in the following log, the Radio services in versions 1.5-1.2 are not found, indicating that the Radio service in version 1.1 has been found (if it is not found, the log will output the corresponding information). If the log shows that the Radio services in versions 1.5-1.0 are not found, it indicates that the Radio service (the earliest version is 1.0) is not found. In this case, request the customer to modify the upper-layer code of the Android device to enable the Radio service.



```

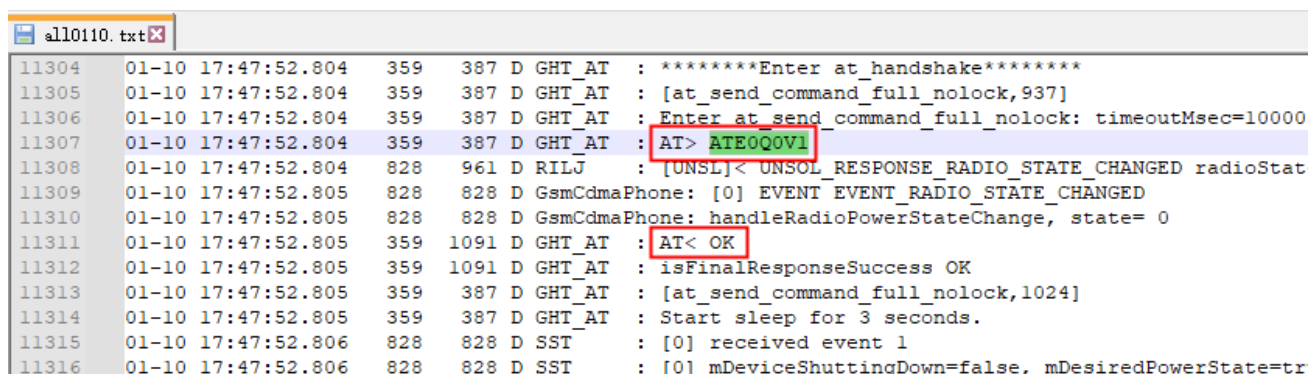
8530 01-10 17:47:50.708 444 563 D EthernetNetworkFactory: acceptRequest, request: NetworkRequest [ REQUEST id=12, [ Capabilities: INTERNET&NOT_RESTRICTED&TRUSTED&NOT_
8531 01-10 17:47:50.708 444 563 I EthernetNetworkFactory: networkForRequest, request: NetworkRequest [ REQUEST id=12, [ Capabilities: INTERNET&NOT_RESTRICTED&TRUSTED&
8532 01-10 17:47:50.708 444 543 W ActivityThread: Failed to find provider info for com.android.calendar (user not unlocked)
8533 01-10 17:47:50.711 155 155 I hwserviceManager: getTransport: Cannot find entry android.hardware.radio@1.3:IRadio/slot1 in either framework or device manifest.
8534 01-10 17:47:50.712 155 155 I hwserviceManager: getTransport: Cannot find entry android.hardware.radio@1.3:IRadio/slot1 in either framework or device manifest.
8535 01-10 17:47:50.713 444 532 D ApConfigUtil: Update Softap capability, add client control feature support
8536 01-10 17:47:50.714 444 532 D ApConfigUtil: Update Softap capability, max client = 16
8537 01-10 17:47:50.715 155 155 I hwserviceManager: getTransport: Cannot find entry android.hardware.radio@1.3:IRadio/slot1 in either framework or device manifest.
8538 01-10 17:47:50.717 155 155 I hwserviceManager: getTransport: Cannot find entry android.hardware.radio@1.2:IRadio/slot1 in either framework or device manifest.
8539 01-10 17:47:50.716 444 543 W ActivityThread: Failed to find provider info for com.android.calendar (user not unlocked)
8540 01-10 17:47:50.717 444 444 D BluetoothManagerService: Trying to bind to profile: 1, while Bluetooth was disabled
8541 01-10 17:47:50.719 444 543 V AlarmManager: setImplLocked type=0 ,when=1673430470586
8542 01-10 17:47:50.720 828 828 I android.os.HwBinder: HwBinder: Starting thread pool for getting: android.hardware.radio@1.1:IRadio/slot1
8543 01-10 17:47:50.721 598 598 D NavigationModeController: getCurrentUserContext: contextUser=0 currentUser=0
8544 01-10 17:47:50.721 598 598 D NavigationModeController: getCurrentInteractionMode: mode=0 contextUser=0

```

Figure 11. Normal main log

2.4 Checking Whether the AT Commands Are Sent

Throughout the operating process, the RIL continuously sends AT commands to the module and receives responses and other messages from the module. Generally, the first AT command sent by RIL to the module is "ATE0Q0V1" and RIL is expected to receive an "OK" response:



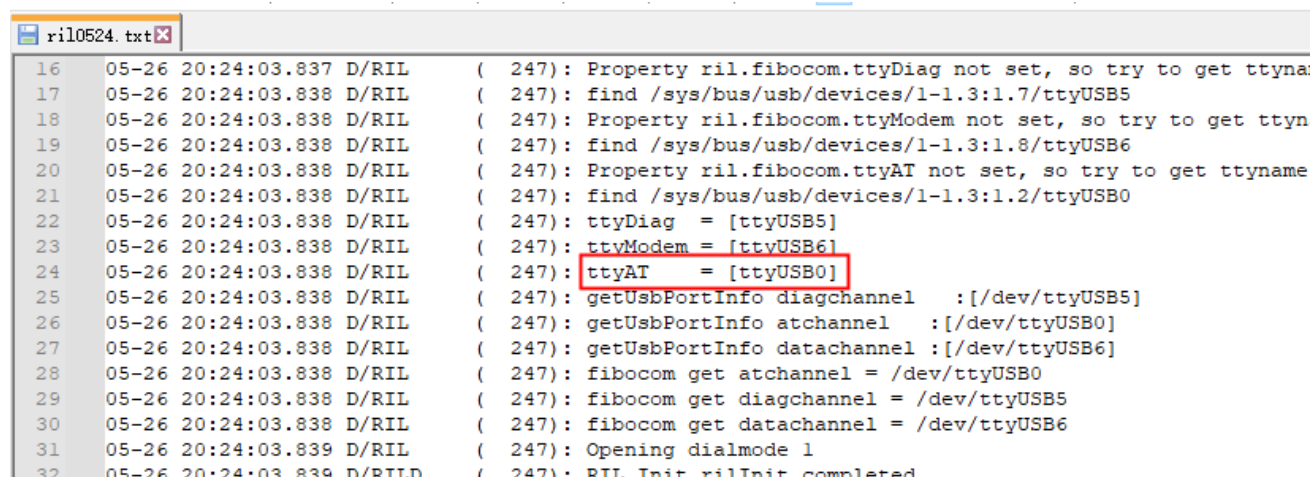
```

11304 01-10 17:47:52.804 359 387 D GHT_AT : *****Enter at handshake*****
11305 01-10 17:47:52.804 359 387 D GHT_AT : [at_send_command_full_nolock,937]
11306 01-10 17:47:52.804 359 387 D GHT_AT : Enter at_send_command_full_nolock: timeoutMsec=10000
11307 01-10 17:47:52.804 359 387 D GHT_AT : AT> ATE0Q0V1
11308 01-10 17:47:52.804 828 961 D RILJ : [UNSL]< UNSOL_RESPONSE_RADIO_STATE_CHANGED radioStat
11309 01-10 17:47:52.805 828 828 D GsmCdmaPhone: [0] EVENT_EVENT_RADIO_STATE_CHANGED
11310 01-10 17:47:52.805 828 828 D GsmCdmaPhone: handleRadioPowerStateChange, state= 0
11311 01-10 17:47:52.805 359 1091 D GHT_AT : AT< OK
11312 01-10 17:47:52.805 359 1091 D GHT_AT : isFinalResponseSuccess OK
11313 01-10 17:47:52.805 359 387 D GHT_AT : [at_send_command_full_nolock,1024]
11314 01-10 17:47:52.805 359 387 D GHT_AT : Start sleep for 3 seconds.
11315 01-10 17:47:52.806 828 828 D SST : [0] received event 1
11316 01-10 17:47:52.806 828 828 D SST : [0] mDeviceShuttingDown=false, mDesiredPowerState=tr

```

Figure 12. First AT command sent by RIL

If the above command is not successfully sent, or if an "OK" response is not received, it is an anomaly. In this situation, search for "ttyAT" in the RIL log and check the AT interface used by the RIL currently. The following figure shows that the RIL is currently using the "/dev/ttyUSB0" as the AT interface.



```
ril0524.txt
16 05-26 20:24:03.837 D/RIL ( 247): Property ril.fibocom.ttyDiag not set, so try to get ttyna
17 05-26 20:24:03.838 D/RIL ( 247): find /sys/bus/usb/devices/l-1.3:1.7/ttyUSB5
18 05-26 20:24:03.838 D/RIL ( 247): Property ril.fibocom.ttyModem not set, so try to get ttyn
19 05-26 20:24:03.838 D/RIL ( 247): find /sys/bus/usb/devices/l-1.3:1.8/ttyUSB6
20 05-26 20:24:03.838 D/RIL ( 247): Property ril.fibocom.ttyAT not set, so try to get ttynam
21 05-26 20:24:03.838 D/RIL ( 247): find /sys/bus/usb/devices/l-1.3:1.2/ttyUSB0
22 05-26 20:24:03.838 D/RIL ( 247): ttyDiag = [ttyUSB5]
23 05-26 20:24:03.838 D/RIL ( 247): ttyModem = [ttyUSB6]
24 05-26 20:24:03.838 D/RIL ( 247): ttyAT = [ttyUSB0]
25 05-26 20:24:03.838 D/RIL ( 247): getUsbPortInfo diagchannel :[/dev/ttyUSB5]
26 05-26 20:24:03.838 D/RIL ( 247): getUsbPortInfo atchannel :[/dev/ttyUSB0]
27 05-26 20:24:03.838 D/RIL ( 247): getUsbPortInfo datachannel :[/dev/ttyUSB6]
28 05-26 20:24:03.838 D/RIL ( 247): fibocom get atchannel = /dev/ttyUSB0
29 05-26 20:24:03.838 D/RIL ( 247): fibocom get diagchannel = /dev/ttyUSB5
30 05-26 20:24:03.838 D/RIL ( 247): fibocom get datachannel = /dev/ttyUSB6
31 05-26 20:24:03.839 D/RIL ( 247): Opening dialmode 1
32 05-26 20:24:03.839 D/RIL ( 247): RIL Init rilInit completed
```

Figure 13. AT interface used by the RIL

Run the following commands to manually send an AT command on the Android device.

```
adb shell
su
stop ril-daemon
echo -e -n "at\r" > /dev/ttyUSBx | cat /dev/ttyUSBx //Response from ttyUSBx
according to the previous query result
//The interface is /dev/ttyUSB0, as shown in the figure above.
```

If the AT response cannot be received on the Android device, the AT interface of the module may be abnormal. Restart the device and try again, or contact the RIL personnel to modify the AT interface used.

3 Abnormal Calling Service

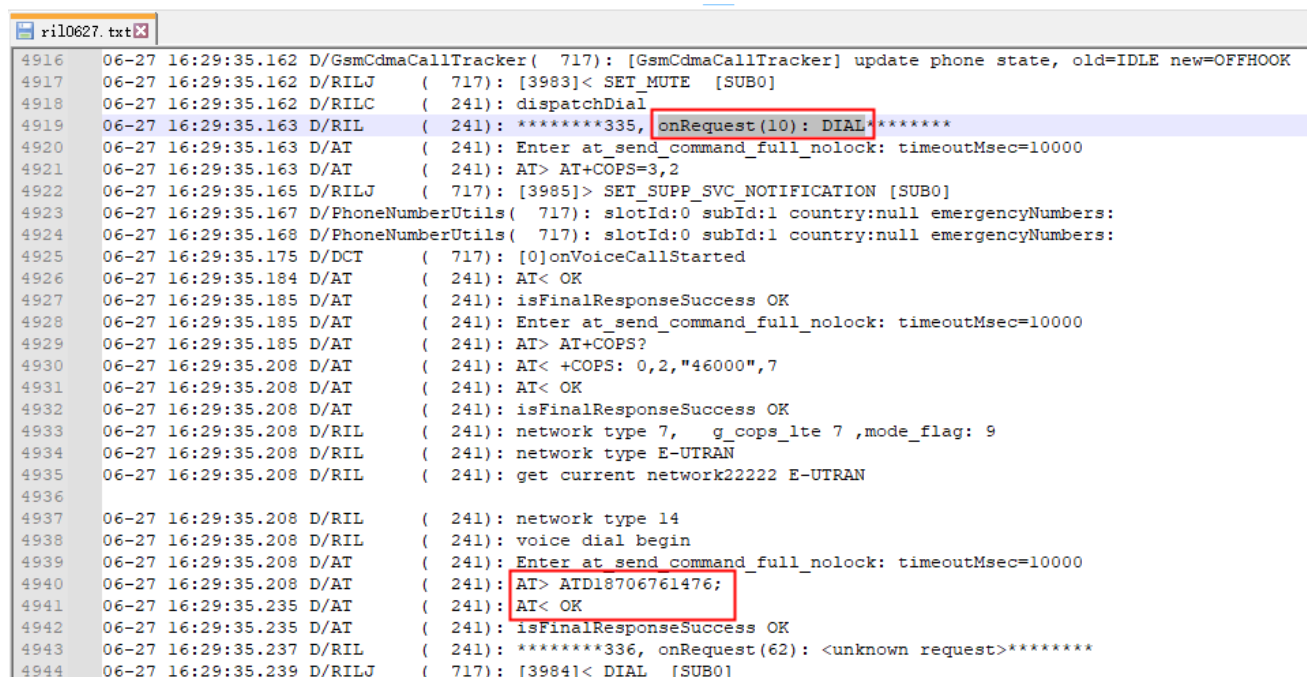
If a call cannot be initiated, it may be caused by the following reasons:

1. The module does not support the calling function.
2. The calling APP of the upper layer is abnormal, for example, insufficient permissions.

The following describes these situations in detail.

3.1 Checking Whether the Module Supports the Calling Service

Search for "DIAL" in the RIL log to find the content of "onRequest(xx): DIAL". The "ATDxxxxxxx" command sent in the request is the AT command for calling:



```

4916 06-27 16:29:35.162 D/GsmCdmaCallTracker( 717): [GsmCdmaCallTracker] update phone state, old=IDLE new=OFFHOOK
4917 06-27 16:29:35.162 D/RILJ ( 717): [3983]< SET MUTE [SUB0]
4918 06-27 16:29:35.162 D/RILC ( 241): dispatchDial
4919 06-27 16:29:35.163 D/RIL ( 241): *****335, onRequest(10): DIAL*****
4920 06-27 16:29:35.163 D/AT ( 241): Enter at_send_command_full_nolock: timeoutMsec=10000
4921 06-27 16:29:35.163 D/AT ( 241): AT> AT+COPS=3,2
4922 06-27 16:29:35.165 D/RILJ ( 717): [3985]> SET_SUPP_SVC_NOTIFICATION [SUB0]
4923 06-27 16:29:35.167 D/PhoneNumberUtils( 717): slotId:0 subId:1 country:null emergencyNumbers:
4924 06-27 16:29:35.168 D/PhoneNumberUtils( 717): slotId:0 subId:1 country:null emergencyNumbers:
4925 06-27 16:29:35.175 D/DCT ( 717): [0]onVoiceCallStarted
4926 06-27 16:29:35.184 D/AT ( 241): AT< OK
4927 06-27 16:29:35.185 D/AT ( 241): isFinalResponseSuccess OK
4928 06-27 16:29:35.185 D/AT ( 241): Enter at_send_command_full_nolock: timeoutMsec=10000
4929 06-27 16:29:35.185 D/AT ( 241): AT> AT+COPS?
4930 06-27 16:29:35.208 D/AT ( 241): AT< +COPS: 0,2,"46000",7
4931 06-27 16:29:35.208 D/AT ( 241): AT< OK
4932 06-27 16:29:35.208 D/AT ( 241): isFinalResponseSuccess OK
4933 06-27 16:29:35.208 D/RIL ( 241): network type 7, g_cops_lte 7 ,mode_flag: 9
4934 06-27 16:29:35.208 D/RIL ( 241): network type E-UTRAN
4935 06-27 16:29:35.208 D/RIL ( 241): get current network22222 E-UTRAN
4936
4937 06-27 16:29:35.208 D/RIL ( 241): network type 14
4938 06-27 16:29:35.208 D/RIL ( 241): voice dial begin
4939 06-27 16:29:35.208 D/AT ( 241): Enter at_send_command_full_nolock: timeoutMsec=10000
4940 06-27 16:29:35.208 D/AT ( 241): ATD18706761476;
4941 06-27 16:29:35.235 D/AT ( 241): AT< OK
4942 06-27 16:29:35.235 D/AT ( 241): isFinalResponseSuccess OK
4943 06-27 16:29:35.237 D/RIL ( 241): *****336, onRequest(62): <unknown request>*****
4944 06-27 16:29:35.239 D/RILJ ( 717): [3984]< DIAL [SUB0]
  
```

Figure 14. "DIAL" request

If you cannot find the "DIAL" request, the upper layer has not sent a calling request, and you need to check the upper layer according to the section [3.2](#).

If the "DIAL" request is found, but the ATD command in the request returns ERROR, check whether the currently used module or SIM card supports the calling function.

3.2 Checking the Calling APP Status

Generally, the upper-layer calling APP has two abnormal situations: There is no calling permission or the Android device does not support the calling function.

3.2.1 Checking Whether There Is a Calling Permission

Taking the Android 7 as an example, enable the calling permission in Settings > Apps > Phone > Permissions.

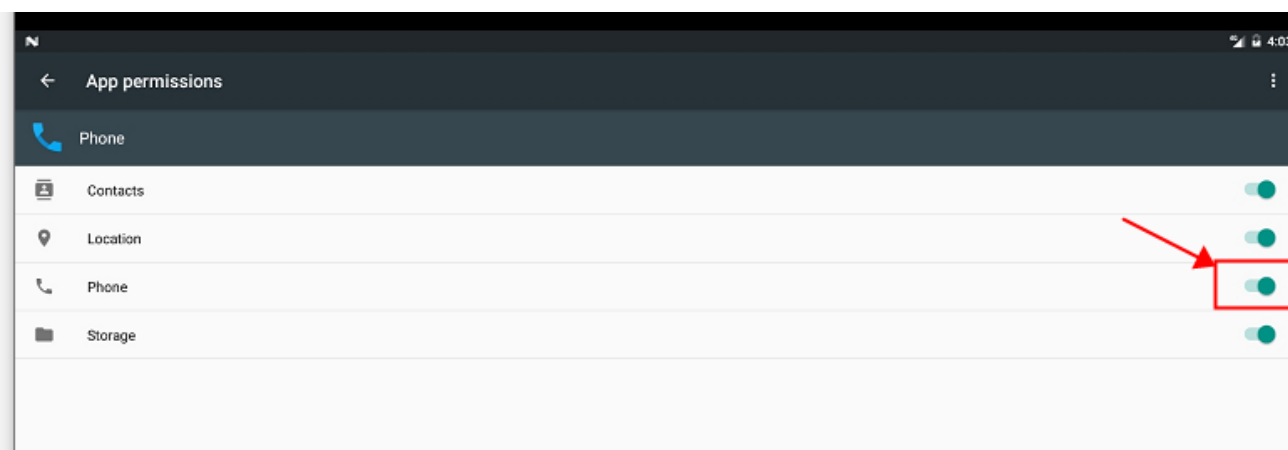


Figure 15. Calling permission of the APP

3.2.2 Checking Whether the Calling Function Is Enabled on the Device

If the customer uses the calling function for the first time and the calling fails, the calling function may not be enabled. If the customer has previously used the calling function, skip this step.

If the customer does not enable the calling function, enable the function according to section [6.1](#).

4 Abnormal SMS Service

Similar to the calling function, there are two abnormal situations of the SMS function.

1. The module does not support the SMS function.
2. The upper-layer SMS APP is abnormal.

The following describes these situations in detail.

4.1 Checking Whether the Module Supports the SMS Service

Search for "SEND_SMS" in the RIL log. If the log contains "onRequest(xx): SEND_SMS", it indicates that the RIL sends an SMS message. AT+CMGS is the AT command used for sending the SMS message. The details are shown in the following figure:

```

6373 06-27 16:51:12.362 V/GsmSMSDispatcher( 717): No carrier package.
6374 06-27 16:51:12.367 D/PhoneNumberUtils( 717): slotId:0 subId:1 country:cn emergencyNumbers:
6375 06-27 16:51:12.368 D/GsmSMSDispatcher( 717): sendSms: isIms()=false mRetryCount=0 mImsRetry=0
6376 06-27 16:51:12.369 D/RILJ ( 717): [40341]> SEND_SMS [SUB0]
6377 06-27 16:51:12.370 D/RIL ( 241): *****385, onRequest(25): SEND_SMS*****
6378 06-27 16:51:12.370 D/AT ( 241): Enter at_send_command_full_nolock: timeoutMsec=10000
6379 06-27 16:51:12.370 D/AT ( 241): AT> AT+COPS=3,2
6380 06-27 16:51:12.380 D/PhoneNumberUtils( 717): slotId:0 subId:1 country:CN emergencyNumbers:
6381 06-27 16:51:12.392 D/AT ( 241): AT< OK
6382 06-27 16:51:12.392 D/AT ( 241): isFinalResponseSuccess OK
6383 06-27 16:51:12.392 D/AT ( 241): Enter at_send_command_full_nolock: timeoutMsec=10000
6384 06-27 16:51:12.393 D/AT ( 241): AT> AT+COPS?
6385 06-27 16:51:12.416 D/AT ( 241): AT< +COPS: 0,2,"46000",7
6386 06-27 16:51:12.416 D/AT ( 241): AT< OK
6387 06-27 16:51:12.416 D/AT ( 241): isFinalResponseSuccess OK
6388 06-27 16:51:12.416 D/RIL ( 241): network type 7, g_cops_lte 7 ,mode_flag: 9
6389 06-27 16:51:12.416 D/RIL ( 241): network type E-UTRAN
6390 06-27 16:51:12.416 D/RIL ( 241): get current network22222 E-UTRAN
6391
6392 06-27 16:51:12.416 D/RIL ( 241): network type 14
6393 06-27 16:51:12.416 D/AT ( 241): Enter at_send_command_full_nolock: timeoutMsec=30000
6394 06-27 16:51:12.416 D/AT ( 241): AT> AT+CMGS=18
6395 06-27 16:51:12.437 D/AT ( 241): AT< >
6396 06-27 16:51:12.437 D/AT ( 241): AT> 0001000b818107761674f6000005e8329bfd06^Z
6397 06-27 16:51:12.500 D/SubscriptionController( 717): sim_state is :READY

```

Figure 16. "SEND_SMS" request

If you cannot find the "SEND_SMS" request, the upper layer has not sent an SMS request, and you need to check the upper layer according to the section [4.2](#).

If the "SEND_SMS" request is found, but AT+CMGS command in the request returns ERROR, check whether the currently used module or SIM card supports the SMS function.

4.2 Checking the SMS APP Status

Generally, the upper-layer SMS APP has two abnormal situations: There is no calling permission or the Android device does not support the SMS function.

4.2.1 Checking Whether There Is a SMS Sending Permission

Taking the Android 7 as an example, enable the SMS permission in Settings > Apps > Messaging > Permissions.

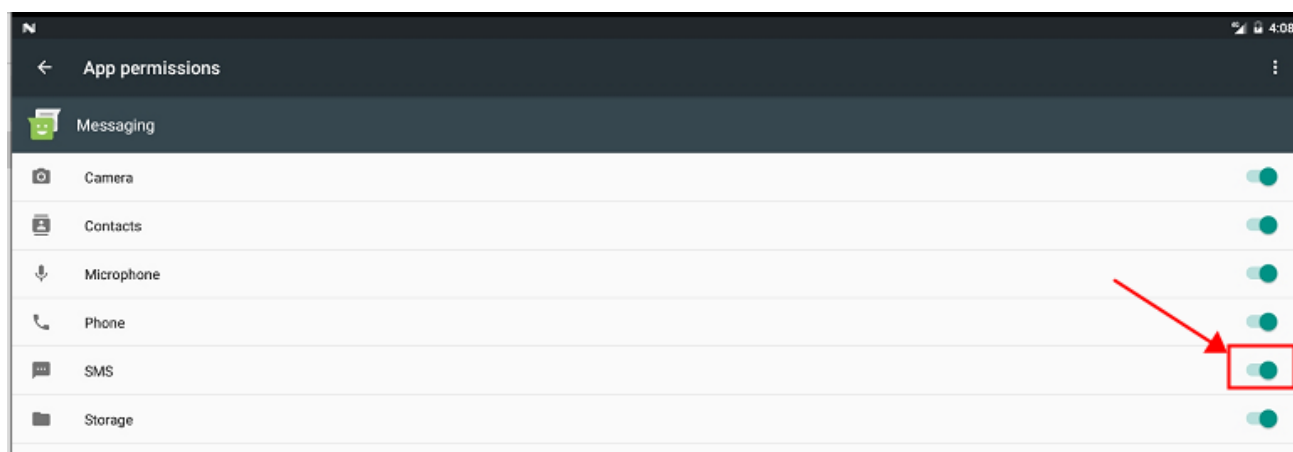


Figure 17. SMS permission of the APP

4.2.2 Checking Whether the SMS Function Is Enabled

If the SMS function is not enabled on the customer's device, enable this function according to section [6.1](#).

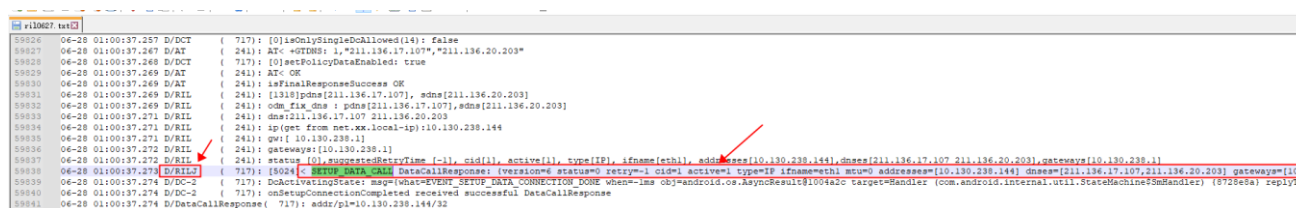
5 Abnormal Data Dial-up

There are many reasons for the data dial-up failure. This chapter describes the troubleshooting methods.

5.1 Checking Whether the Dial-up Request Is Successful

Check whether the upper layer initiates the dial-up and whether the dial-up is successful. In the RIL log, search for "SETUP_DATA_CALL". "onRequest(xx): SETUP_DATA_CALL" in the log indicates the RIL dial-up process. The log with the LOG TAG of RILJ and the content of "< SETUP_DATA_CALL" records the RIL dial-up result.

If the RIL log contains the RIL dial-up process and the result includes the IP and gateway addresses, the dial-up is successful. See Figure 18.



```

59926 06-20 01:00:37.257 D/DCT ( 717): [0]isOnlySingleDialAllowed(14): false
59927 06-20 01:00:37.267 D/AT ( 241): AT< +GTRNDIS: 1,"211.136.17.107","211.136.20.203"
59928 06-20 01:00:37.268 D/DCT ( 717): [0]setPolicyDataEnabled: true
59929 06-20 01:00:37.269 D/AT ( 241): AT< OK
59930 06-20 01:00:37.269 D/AT ( 241): isFinalResponseSuccess OK
59931 06-20 01:00:37.269 D/RIL ( 241): [1315]pdns[211.136.17.107], adns[211.136.20.203]
59932 06-20 01:00:37.269 D/RIL ( 241): odm_fix_dns : pdns[211.136.17.107],adns[211.136.20.203]
59933 06-20 01:00:37.271 D/RIL ( 241): dns=211.136.17.107 211.136.20.203
59934 06-20 01:00:37.271 D/RIL ( 241): ip(get from net.xml.local-ip):10.130.238.144
59935 06-20 01:00:37.271 D/RIL ( 241): gw:[ 10.130.238.1]
59936 06-20 01:00:37.272 D/RIL ( 241): gateways:[10.130.238.1]
59937 06-20 01:00:37.272 D/RIL ( 241): status [0],suppnetSetRegTime [-1],cid[0],active[0],type[0],ifname[eth0],addr=10.130.238.144,dns=211.136.17.107 211.136.20.203,gateways[10.130.238.1]
59938 06-20 01:00:37.274 D/RIL ( 717): [5024] < SETUP_DATA_CALL DataCallResponse: (version=0 status=0 regty=1 cid=1 active=1 type=0 ifname=eth0 scu=0 addresses=[10.130.238.144] dnses=[211.136.17.107,211.136.20.203] gateways=[10.130.238.1])
59939 06-20 01:00:37.274 D/DC-2 ( 717): DoActivatingState: msg=[what=EVENT_SETUP_DATA_CONNECTION_DONE when=-1ms obj=android.os.AsyncResult@81004a2c target=Handler (com.android.internal.util.StateMachine$StateMachine) (8728e8a) reply=
59940 06-20 01:00:37.274 D/DC-2 ( 717): onSetupConnectionCompleted received successful DataCallResponse
59941 06-20 01:00:37.274 D/DataCallResponse ( 717): addr/pl=10.130.238.144/32

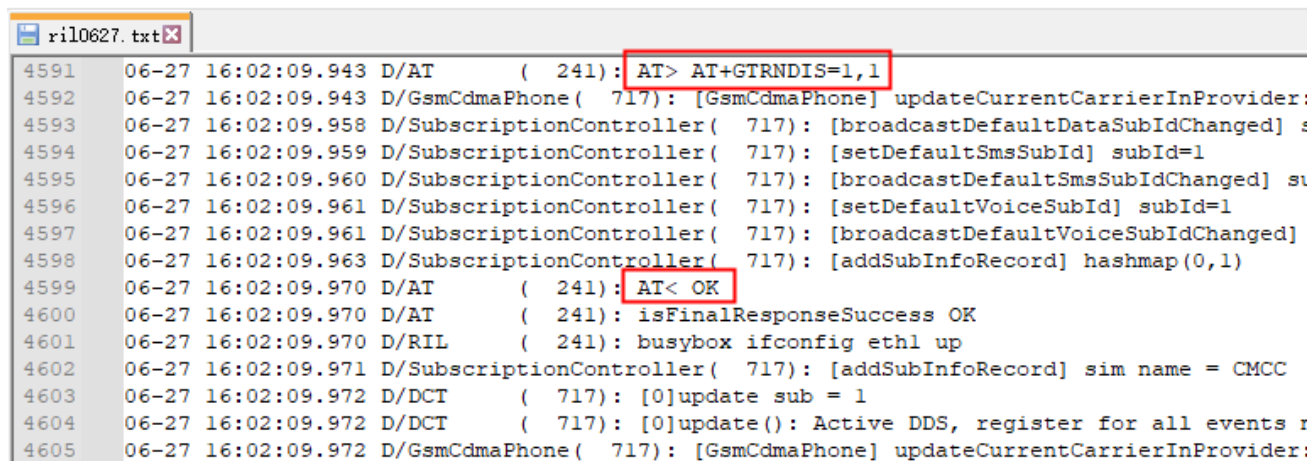
```

Figure 18. RIL dial-up result

If there is no RIL dial-up process, check whether the APN is normal according to section 5.2. If there is a dial-up process but the dial-up fails, proceed with the following steps.

5.1.1 ECM Dial-up

1. For the ECM dial-up, check whether OK is returned in response to the GTRNDIS command.



```

ril0627.txt
4591 06-27 16:02:09.943 D/AT ( 241): AT+GTRNDIS=1,1
4592 06-27 16:02:09.943 D/GsmCdmaPhone( 717): [GsmCdmaPhone] updateCurrentCarrierInProvider:
4593 06-27 16:02:09.958 D/SubscriptionController( 717): [broadcastDefaultDataSubIdChanged] :
4594 06-27 16:02:09.959 D/SubscriptionController( 717): [setDefaultSmsSubId] subId=1
4595 06-27 16:02:09.960 D/SubscriptionController( 717): [broadcastDefaultSmsSubIdChanged] s
4596 06-27 16:02:09.961 D/SubscriptionController( 717): [setDefaultVoiceSubId] subId=1
4597 06-27 16:02:09.961 D/SubscriptionController( 717): [broadcastDefaultVoiceSubIdChanged]
4598 06-27 16:02:09.963 D/SubscriptionController( 717): [addSubInfoRecord] hashmap(0,1)
4599 06-27 16:02:09.970 D/AT ( 241): AT< OK
4600 06-27 16:02:09.970 D/AT ( 241): isFinalResponseSuccess OK
4601 06-27 16:02:09.970 D/RIL ( 241): busybox ifconfig eth1 up
4602 06-27 16:02:09.971 D/SubscriptionController( 717): [addSubInfoRecord] sim name = CMCC
4603 06-27 16:02:09.972 D/DCT ( 717): [0]update sub = 1
4604 06-27 16:02:09.972 D/DCT ( 717): [0]update(): Active DDS, register for all events :
4605 06-27 16:02:09.972 D/GsmCdmaPhone( 717): [GsmCdmaPhone] updateCurrentCarrierInProvider:

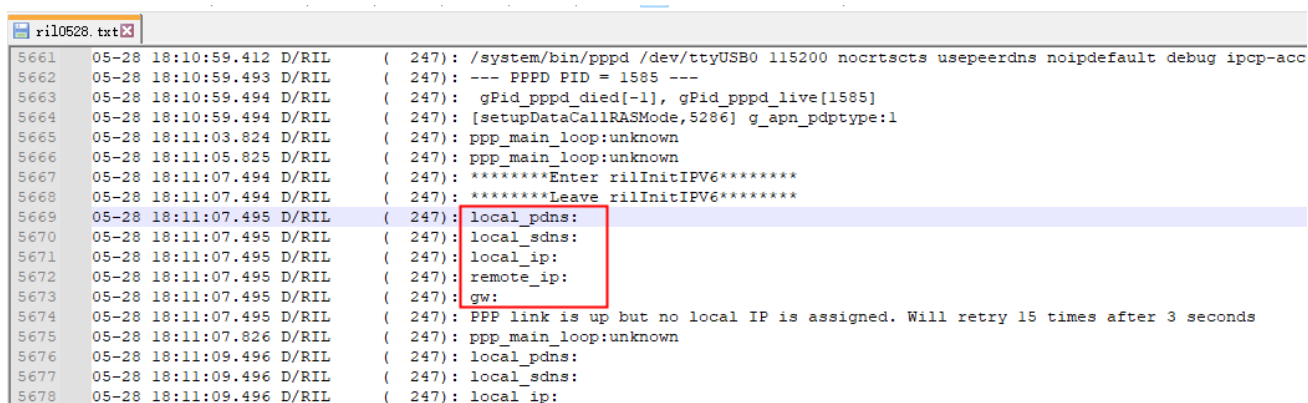
```

Figure 19. GTRNDIS command

If the command returns ERROR, check whether the USBMODE is correct. If USBMODE is correct, capture the log of the module and request the personnel responsible for the module to analyze.

5.1.2 PPP Dial-up

For the PPP dial-up, a common error is that the IP address is not obtained. The details are shown in the following figure:



```

ril0528.txt
5661 05-28 18:10:59.412 D/RIL ( 247): /system/bin/pppd /dev/ttyUSB0 115200 noctrlscts usepeerdns noipdefault debug ipcp-acc
5662 05-28 18:10:59.493 D/RIL ( 247): --- PPPD PID = 1585 ---
5663 05-28 18:10:59.494 D/RIL ( 247): gPid_pppd_died[-1], gPid_pppd_live[1585]
5664 05-28 18:10:59.494 D/RIL ( 247): [setupDataCallRASMode,5286] g_apn_pdptype:1
5665 05-28 18:11:03.824 D/RIL ( 247): ppp_main_loop:unknown
5666 05-28 18:11:05.825 D/RIL ( 247): ppp_main_loop:unknown
5667 05-28 18:11:07.494 D/RIL ( 247): *****Enter rilInitIPv6*****
5668 05-28 18:11:07.494 D/RIL ( 247): *****Leave rilInitIPv6*****
5669 05-28 18:11:07.495 D/RIL ( 247): local_pdns:
5670 05-28 18:11:07.495 D/RIL ( 247): local_sdns:
5671 05-28 18:11:07.495 D/RIL ( 247): local_ip:
5672 05-28 18:11:07.495 D/RIL ( 247): remote_ip:
5673 05-28 18:11:07.495 D/RIL ( 247): gw:
5674 05-28 18:11:07.495 D/RIL ( 247): PPP link is up but no local IP is assigned. Will retry 15 times after 3 seconds
5675 05-28 18:11:07.826 D/RIL ( 247): ppp_main_loop:unknown
5676 05-28 18:11:09.496 D/RIL ( 247): local_pdns:
5677 05-28 18:11:09.496 D/RIL ( 247): local_sdns:
5678 05-28 18:11:09.496 D/RIL ( 247): local_ip:

```

Figure 20. PPP dial-up without IP address

Analyze the RIL log (see section [1.1](#) for the capture method). If necessary, request the personnel responsible for the module to analyze. For example, identify the cause for the peer to deny the authentication, as shown in the following figure.


```

ppp0527.txt
4 05-27 11:36:31.641 1847 1847 D pppd : using channel 1
5 05-27 11:36:31.641 1847 1847 I pppd : Using interface ppp0
6 05-27 11:36:31.642 1847 1847 I pppd : Connect: ppp0 <-> /dev/ttyUSB0
7 05-27 11:36:31.642 1847 1847 D pppd : sent [LCP ConfReq id=0x1 <asynctest 0x0> <magic 0x46e16efc> <pcomp> <accomp>]
8 05-27 11:36:31.644 1847 1847 D pppd : rcvcd [LCP ConfReq id=0x1 <asynctest 0x0> <auth chap MD5> <magic 0x304256d8> <pcomp> <accomp>]
9 05-27 11:36:31.644 1847 1847 D pppd : No auth is possible
10 05-27 11:36:31.644 1847 1847 D pppd : sent [LCP ConfReq id=0x1 <auth chap MD5>]
11 05-27 11:36:31.644 1847 1847 D pppd : rcvcd [LCP ConfAck id=0x1 <asynctest 0x0> <magic 0x46e16efc> <pcomp> <accomp>]
12 05-27 11:36:31.645 1847 1847 D pppd : rcvcd [LCP ConfReq id=0x2 <asynctest 0x0> <magic 0x304256d8> <pcomp> <accomp>]
13 05-27 11:36:31.645 1847 1847 D pppd : sent [LCP ConfAck id=0x2 <asynctest 0x0> <magic 0x304256d8> <pcomp> <accomp>]
14 05-27 11:36:31.646 1847 1847 D pppd : sent [CCP ConfReq id=0x1 <deflate 15> <deflate(old#) 15> <bsd vl 15>]
15 05-27 11:36:31.647 1847 1847 D pppd : sent [IPCP ConfReq id=0x1 <compress VJ 0f 01> <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-c
16 05-27 11:36:31.647 1847 1847 D pppd : rcvcd [LCP TermReq id=0x3 "peer refused to authenticate"]
17 05-27 11:36:31.647 1847 1847 I pppd : LCP terminated by peer (peer refused to authenticate)
18 05-27 11:36:31.647 1847 1847 D pppd : sent [LCP TermAck id=0x3]
19 05-27 11:36:34.650 1847 1847 I pppd : Connection terminated.
20 05-27 11:36:35.664 1847 1847 I pppd : Modem hangup

```

Figure 21. Example PPP dial-up error

5.2 Checking Whether the APN Is Normal

Search for "ATTACH_APN" in the RIL log. If there is a "RIL_REQUEST_SET_INITIAL_ATTACH_APN" request and the "AT+CGDCONT" command in this request does not return ERROR, the APN is normal. The details are shown in the following figure:

```

ril0627.txt
4455 06-27 16:02:09.770 D/RILJ ( 717): Set RIL_REQUEST_SET_INITIAL_ATTACH_APN [SUB0]
4456 06-27 16:02:09.771 D/RILJ ( 717): [3970]> RIL_REQUEST_SET_INITIAL_ATTACH_APN, apn:cmnet, protocol:IP, authType:-1
4457 06-27 16:02:09.771 D/DCI ( 717): [0]onRecordsLoadedOrSubIdChanged: notifying data availability
4458 06-27 16:02:09.772 D/RIL ( 241): *****321, onRequest(111): RIL_REQUEST_SET_INITIAL_ATTACH_APN*****
4459 06-27 16:02:09.772 D/RIL ( 241): requestSetInitialAttachAPN E
4460 06-27 16:02:09.772 D/RIL ( 241): [requestSetInitialAttachAPN] apn:cmnet, pdp_type:IP
4461 06-27 16:02:09.772 D/RIL ( 241): [requestSetInitialAttachAPN] cur_oper:0, g_cops_lte:7
4462 06-27 16:02:09.772 D/AT ( 241): Enter at send command_full_nolock: timeoutMsec=10000
4463 06-27 16:02:09.772 D/AT ( 241): AT> AT+CGDCONT?
4464 06-27 16:02:09.792 D/DCI ( 717): [0]setupDataOnConnectableApns: simLoaded hipri:[state=IDLE,enabled=false] mms:[
4465 06-27 16:02:09.792 D/DCI ( 717): [0]isConnectable() call trySetupData
4466 06-27 16:02:09.792 D/DCI ( 717): [0]trySetupData for type:default due to simLoaded, mIsPpsRestricted=false
4467 06-27 16:02:09.794 D/DCI ( 717): [0]isEmergency: result=false
4468 06-27 16:02:09.794 D/DCI ( 717): [0]buildWaitingApns: E requestedApnType=default
4469 06-27 16:02:09.796 D/AT ( 241): AT+CGDCONT: 0,"IP","CMNET","10.254.182.213",0,0
4470 06-27 16:02:09.799 D/AT ( 241): AT< OK
4471 06-27 16:02:09.799 D/AT ( 241): isFinalResponseSuccess OK
4472 06-27 16:02:09.799 D/RIL ( 241): Current PDP_type:IP, Current APN_Name:cmnet, get cid:1
4473 06-27 16:02:09.799 D/RIL ( 241): [APNInfoChange,6453]Old_PDP_type:IP, New_Pdp_Type:IP
4474 06-27 16:02:09.800 D/RTT ( 241): [APNInfoChange,6454]Old APN Name:cmnet. New APN Name:cmnet

```

Figure 22. APN setting in RIL

If there is no "RIL_REQUEST_SET_INITIAL_ATTACH_APN" request, check whether the module is registered with the PS domain according to section 5.3. If this request exists but the AT+CGDCONT command returns ERROR, request the personnel responsible for the module to identify the cause for the error returned by the AT command.

5.3 Checking Whether the Module Registers with the PS Domain

The module cannot initiate the data dial-up if it is not registered with the PS domain.

In the RIL log, search for "DATA_REGISTRATION_STATE" and find the DATA_REGISTRATION_STATE request. Check whether the AT+CEREG? command in the request returns the correct information (as shown in the following figure).

```

ril0627.txt
3135 06-27 16:02:06.433 D/RIL ( 241): *****249, onRequest(21): DATA_REGISTRATION_STATE*****
3136 06-27 16:02:06.433 D/RIL ( 241): *****enter requestRegistrationState*****
3137 06-27 16:02:06.433 D/AT ( 241): Enter at_send_command_full_nolock: timeoutMsec=10000
3138 06-27 16:02:06.433 D/AT ( 241): AT> AT+COPS=3,2
3139 06-27 16:02:06.434 D/RIL ( 717): [3897]< OPERATOR {CHINA MOBILE, CMCC, 46000} [SUB0]
3140 06-27 16:02:06.454 D/AT ( 241): AT< OK
3141 06-27 16:02:06.455 D/AT ( 241): isFinalResponseSuccess OK
3142 06-27 16:02:06.455 D/AT ( 241): Enter at_send_command_full_nolock: timeoutMsec=10000
3143 06-27 16:02:06.455 D/AT ( 241): AT> AT+COPS?
3144 06-27 16:02:06.477 D/AT ( 241): AT< +COPS: 0,2,"46000",7
3145 06-27 16:02:06.477 D/AT ( 241): AT< OK
3146 06-27 16:02:06.477 D/AT ( 241): isFinalResponseSuccess OK
3147 06-27 16:02:06.477 D/RIL ( 241): network type 7, g_cops_lte 7 ,mode_flag: 9
3148 06-27 16:02:06.477 D/RIL ( 241): network type E-UTRAN
3149 06-27 16:02:06.477 D/RIL ( 241): get current network22222 E-UTRAN
3150
3151 06-27 16:02:06.478 D/RIL ( 241): network type 14
3152 06-27 16:02:06.478 D/AT ( 241): Enter at_send_command_full_nolock: timeoutMsec=10000
3153 06-27 16:02:06.478 D/AT ( 241): AT> AT+CEREG=2
3154 06-27 16:02:06.499 D/AT ( 241): AT< OK
3155 06-27 16:02:06.499 D/AT ( 241): isFinalResponseSuccess OK
3156 06-27 16:02:06.499 D/AT ( 241): Enter at_send_command_full_nolock: timeoutMsec=10000
3157 06-27 16:02:06.499 D/AT ( 241): AT> AT+CGREG=2
3158 06-27 16:02:06.521 D/AT ( 241): AT< OK
3159 06-27 16:02:06.521 D/AT ( 241): isFinalResponseSuccess OK
3160 06-27 16:02:06.521 D/AT ( 241): Enter at_send_command_full_nolock: timeoutMsec=10000
3161 06-27 16:02:06.521 D/AT ( 241): AT> AT+CEREG=2
3162 06-27 16:02:06.542 D/AT ( 241): AT< OK
3163 06-27 16:02:06.542 D/AT ( 241): isFinalResponseSuccess OK
3164 06-27 16:02:06.543 D/AT ( 241): Enter at_send_command_full_nolock: timeoutMsec=10000
3165 06-27 16:02:06.543 D/AT ( 241): AT> AT+CEREG?
3166 06-27 16:02:06.568 D/AT ( 241): AT< +CEREG: 2,1,"90F3", "07828A01",7
3167 06-27 16:02:06.568 D/AT ( 241): AT< OK
3168 06-27 16:02:06.569 D/AT ( 241): isFinalResponseSuccess OK
3169 06-27 16:02:06.569 D/RIL ( 241): parseRegistrationState. Parsing: +CEREG: 2,1,"90F3", "07828A01",7
3170 06-27 16:02:06.569 D/RIL ( 241): commas = 4
3171 06-27 16:02:06.569 D/RIL ( 241): resp[] = 1, 37107, 125995521, 7
3172 06-27 16:02:06.569 D/RIL ( 241): resp[] = 1, 37107, 125995521, 7
3173 06-27 16:02:06.569 D/RIL ( 241): [3837]registration[0][1][2][3] == 1,37107,125995521,7
3174
3175 06-27 16:02:06.569 D/RIL ( 241): registration[3] == 14
3176
3177 06-27 16:02:06.569 D/RIL ( 241): registration[0] == 1
3178
3179 06-27 16:02:06.569 D/RIL ( 241): *****250, onRequest(20): VOICE_REGISTRATION_STATE*****
3180 06-27 16:02:06.570 D/RIL ( 241): *****enter requestRegistrationState*****
3181 06-27 16:02:06.570 D/AT ( 241): Enter at_send_command_full_nolock: timeoutMsec=10000
3182 06-27 16:02:06.570 D/AT ( 241): AT> AT+COPS=3,2
3183 06-27 16:02:06.570 D/RIL ( 717): [3898]< DATA_REGISTRATION_STATE {1, 90f3, 7828a01, 14} [SUB0]
3184 06-27 16:02:06.591 D/AT ( 241): AT< OK

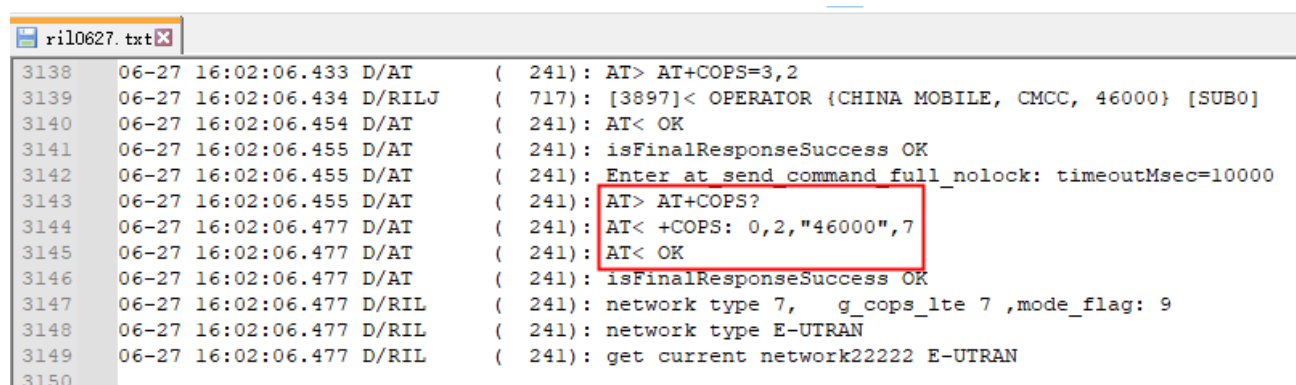
```

Figure 23. DATA_REGISTRATION_STATE request

If the AT+CEREG? command does not return the available cell information (for example, the return value is 2, 0), check whether the module is correctly registered with the network according to section [5.4](#).

5.4 Checking Whether the Network Registration Is Successful

In the RIL log, run COPS to find the AT+COPS command. Then, you can identify whether the module is successfully registered with the network. The following figure shows the log when the module is successfully registered with the network.



```
ril0627.txt
3138 06-27 16:02:06.433 D/AT ( 241): AT> AT+COPS=3,2
3139 06-27 16:02:06.434 D/RILJ ( 717): [3897]< OPERATOR {CHINA MOBILE, CMCC, 46000} [SUB0]
3140 06-27 16:02:06.454 D/AT ( 241): AT< OK
3141 06-27 16:02:06.455 D/AT ( 241): isFinalResponseSuccess OK
3142 06-27 16:02:06.455 D/AT ( 241): Enter at send command full_nolock: timeoutMsec=10000
3143 06-27 16:02:06.455 D/AT ( 241): AT> AT+COPS?
3144 06-27 16:02:06.477 D/AT ( 241): AT< +COPS: 0,2,"46000",7
3145 06-27 16:02:06.477 D/AT ( 241): AT< OK
3146 06-27 16:02:06.477 D/AT ( 241): isFinalResponseSuccess OK
3147 06-27 16:02:06.477 D/RIL ( 241): network type 7, g_cops_lte 7 ,mode_flag: 9
3148 06-27 16:02:06.477 D/RIL ( 241): network type E-UTRAN
3149 06-27 16:02:06.477 D/RIL ( 241): get current network22222 E-UTRAN
3150
```

Figure 24. AT+COPS command

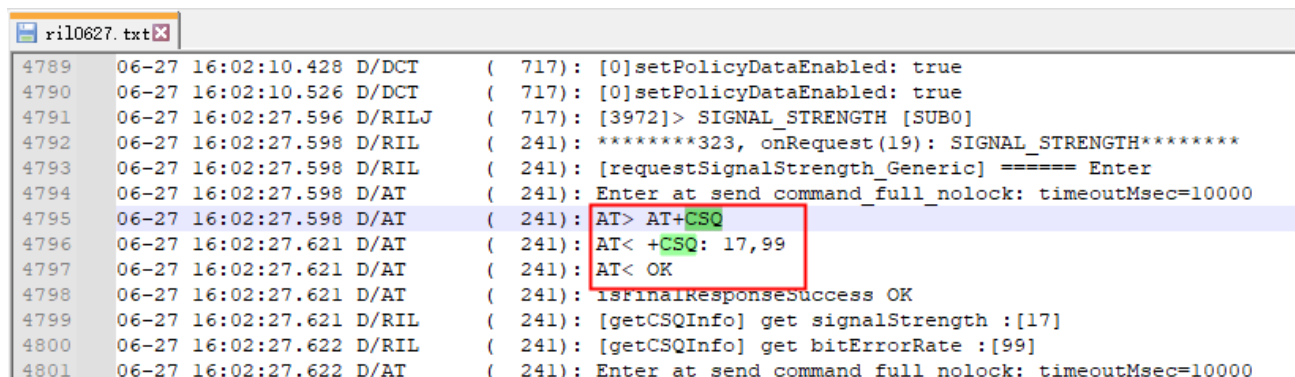
Unsuccessful network registration may be caused by the following reasons, listed based on the order of probability from high to low:

1. The signal strength is poor.
2. The SIM card is not detected.
3. The network system is incorrect.

The following section respectively describes these situations in detail.

5.4.1 Signal Strength

In the RIL log, search for "CSQ" to obtain the current signal strength. The details are shown in the following figure:



```

4789 06-27 16:02:10.428 D/DCT ( 717): [0]setPolicyDataEnabled: true
4790 06-27 16:02:10.526 D/DCT ( 717): [0]setPolicyDataEnabled: true
4791 06-27 16:02:27.596 D/RILJ ( 717): [3972]> SIGNAL_STRENGTH [SUB0]
4792 06-27 16:02:27.598 D/RIL ( 241): *****323, onRequest(19): SIGNAL_STRENGTH*****
4793 06-27 16:02:27.598 D/RIL ( 241): [requestSignalStrength_Generic] ===== Enter
4794 06-27 16:02:27.598 D/AT ( 241): Enter at send command_full_nolock: timeoutMsec=10000
4795 06-27 16:02:27.598 D/AT ( 241): AT> AT+CSQ
4796 06-27 16:02:27.621 D/AT ( 241): AT< +CSQ: 17,99
4797 06-27 16:02:27.621 D/AT ( 241): AT< OK
4798 06-27 16:02:27.621 D/AT ( 241): isFinalResponseSuccess OK
4799 06-27 16:02:27.621 D/RIL ( 241): [getCSQInfo] get signalStrength :[17]
4800 06-27 16:02:27.622 D/RIL ( 241): [getCSQInfo] get bitErrorRate :[99]
4801 06-27 16:02:27.622 D/AT ( 241): Enter at send command_full_nolock: timeoutMsec=10000

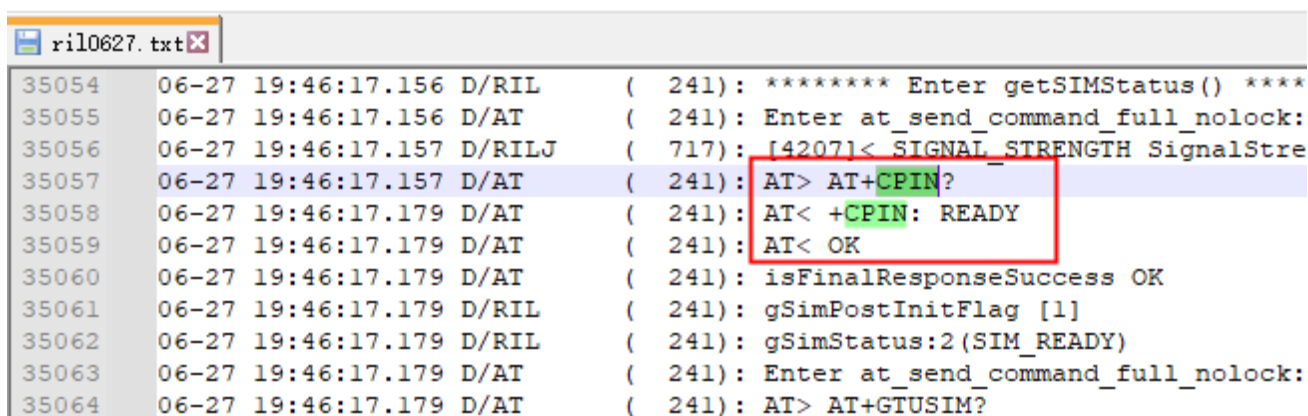
```

Figure 25. AT+CSQ

If the signal strength value is low, check the antenna and the signal coverage in the corresponding area.

5.4.2 SIM Card Detection

In the RIL log, search for "CPIN" to check whether the SIM card is correctly identified. The details are shown in the following figure:



```

35054 06-27 19:46:17.156 D/RIL ( 241): ***** Enter getSIMStatus() *****
35055 06-27 19:46:17.156 D/AT ( 241): Enter at_send_command_full_nolock:
35056 06-27 19:46:17.157 D/RILJ ( 717): [42071]< SIGNAL_STRENGTH SignalStre
35057 06-27 19:46:17.157 D/AT ( 241): AT> AT+CPIN?
35058 06-27 19:46:17.179 D/AT ( 241): AT< +CPIN: READY
35059 06-27 19:46:17.179 D/AT ( 241): AT< OK
35060 06-27 19:46:17.179 D/AT ( 241): isFinalResponseSuccess OK
35061 06-27 19:46:17.179 D/RIL ( 241): gSimPostInitFlag [1]
35062 06-27 19:46:17.179 D/RIL ( 241): gSimStatus:2 (SIM_READY)
35063 06-27 19:46:17.179 D/AT ( 241): Enter at_send_command_full_nolock:
35064 06-27 19:46:17.179 D/AT ( 241): AT> AT+GTUSIM?

```

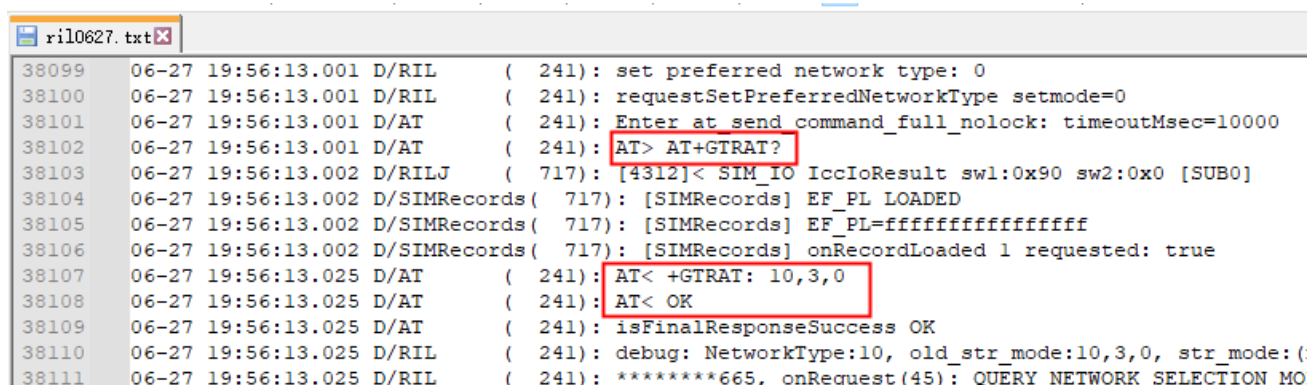
Figure 26. AT+CPIN

If the SIM card is not correctly loaded, check whether the SIM card is inserted properly or whether the hardware circuit is normal.

5.4.3 Network System

If the module uses the 4G+2G or 4G system and the current network system is 2G, the module may slowly register with the network or fail to register with the network.

In the RIL log, search for "GTRAT" to obtain the current network system. The details are shown in the following figure:



```
38099 06-27 19:56:13.001 D/RIL ( 241): set preferred network type: 0
38100 06-27 19:56:13.001 D/RIL ( 241): requestSetPreferredNetworkType setmode=0
38101 06-27 19:56:13.001 D/AT ( 241): Enter at send command_full_nolock: timeoutMsec=10000
38102 06-27 19:56:13.001 D/AT ( 241): AT> AT+GTRAT?
38103 06-27 19:56:13.002 D/RILJ ( 717): [4312]< SIM_IO IccIoResult sw1:0x90 sw2:0x0 [SUB0]
38104 06-27 19:56:13.002 D/SIMRecords( 717): [SIMRecords] EF_PL LOADED
38105 06-27 19:56:13.002 D/SIMRecords( 717): [SIMRecords] EF_PL=ffffffffffffffff
38106 06-27 19:56:13.002 D/SIMRecords( 717): [SIMRecords] onRecordLoaded 1 requested: true
38107 06-27 19:56:13.025 D/AT ( 241): AT< +GTRAT: 10,3,0
38108 06-27 19:56:13.025 D/AT ( 241): AT< OK
38109 06-27 19:56:13.025 D/AT ( 241): isFinalResponseSuccess OK
38110 06-27 19:56:13.025 D/RIL ( 241): debug: NetworkType:10, old_str_mode:10,3,0, str_mode:(
38111 06-27 19:56:13.025 D/RIL ( 241): *****665, onRequest(45): QUERY_NETWORK_SELECTION_MO
```

Figure 27. AT+GTRAT

If the module adopts the 2G network system, change the network system to 4G in Settings > More > Cellular networks > Preferred network type (taking Android 7 as an example).

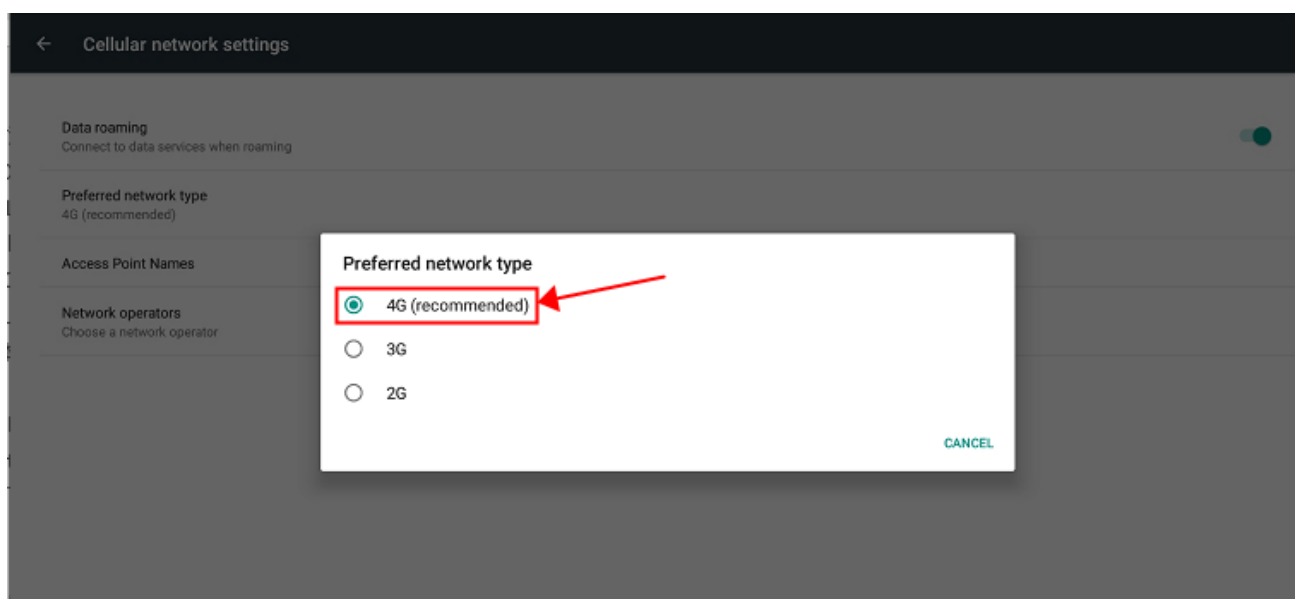


Figure 28. Setting the preferred network

6 Accessories

6.1 Modification on the AP Side

```
./device/rockchip/common/overlay/frameworks/base/core/res/res/values/config.xml
diff --git
a/device/rockchip/common/overlay/frameworks/base/core/res/res/values/config.xml
b/device/rockchip/common/overlay/frameworks/base/core/res/res/values/config.xml
index 5663d84..e88088b 100644
--- a/device/rockchip/common/overlay/frameworks/base/core/res/res/values/config.xml
+++ b/device/rockchip/common/overlay/frameworks/base/core/res/res/values/config.xml
@@ -20,10 +20,10 @@
<resources>

    <!-- This device is not "voice capable"; it's data-only. -->
-    <bool name="config_voice_capable">false</bool>
+    <bool name="config_voice_capable">true</bool>

    <!-- This device does not allow sms service. -->
-    <bool name="config_sms_capable">false</bool>
+    <bool name="config_sms_capable">true</bool>

    <!-- If this is true, the screen will come on when you unplug usb/power/whatever.
-->
    <bool name="config_unplugTurnsOnScreen">true</bool>

./device/rockchip/common/tv/overlay/frameworks/base/core/res/res/values/config.xml
diff --git
a/device/rockchip/common/tv/overlay/frameworks/base/core/res/res/values/config.xml
b/device/rockchip/common/tv/overlay/frameworks/base/core/res/res/values/config.xml
index cbd8423..67d4dda 100755
```

```
---
a/device/rockchip/common/tv/overlay/frameworks/base/core/res/res/values/config.xml
+++
b/device/rockchip/common/tv/overlay/frameworks/base/core/res/res/values/config.xml
@@ -24,10 +24,10 @@
    <bool name="config_sendAudioBecomingNoisy">false</bool>

    <!-- This device is data-only. -->
-   <bool name="config_voice_capable">false</bool>
+   <bool name="config_voice_capable">true</bool>

    <!-- This device does not allow sms service. -->
-   <bool name="config_sms_capable">false</bool>
+   <bool name="config_sms_capable">true</bool>

    <!-- Control the default UI mode type to use when there is no other type override
        happening. One of the following values (See Configuration.java):
```