Android RIL Driver
User Guide

Rev. Android_RIL_Driver_User_Guide_V1.9

Date: 2020-05-11

Status: Released

www.quectel.com

Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company headquarters:

Quectel Wireless Solutions Co., Ltd.
Building 5, Shanghai Business Park Phase III (Area B), No.1016 Tianlin Road, Minhang District, Shanghai 200233, China
Tel: +86 21 5108 6236
Email: info@quectel.com

Or our local office. For more information, please visit:
http://www.quectel.com/support/sales.htm

For technical support, or to report documentation errors, please visit:
http://www.quectel.com/support/technical.htm
Or email to: support@quectel.com

GENERAL NOTES

QUECTEL OFFERS THE INFORMATION AS A SERVICE TO ITS CUSTOMERS. THE INFORMATION PROVIDED IS BASED UPON CUSTOMERS' REQUIREMENTS. QUECTEL MAKES EVERY EFFORT TO ENSURE THE QUALITY OF THE INFORMATION IT MAKES AVAILABLE. QUECTEL DOES NOT MAKE ANY WARRANTY AS TO THE INFORMATION CONTAINED HEREIN, AND DOES NOT ACCEPT ANY LIABILITY FOR ANY INJURY, LOSS OR DAMAGE OF ANY KIND INCURRED BY USE OF OR RELIANCE UPON THE INFORMATION. ALL INFORMATION SUPPLIED HEREIN IS SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.

About the Document

Revision History

Version	Date	Author	Description
1.0	2015-02-27	Carl YIN	Initial
1.1	2015-03-25	Carl YIN	Updated supported products
1.2	2015-04-07	Kent XU	Added zero packet feature in Section 3.3.3.
1.3	2015-07-10	Kent XU	1. Added GSM modules in supported products 2. Added Android 5.x in supported Android versions
1.4	2016-06-21	Carl YIN/ Neo HOU	1. Added EC21&EC25 in supported products 2. Added FAQ
1.5	2016-08-23	Kent XU	Added EC20 R2.0 in supported products
1.6	2017-01-05	Macro GAO	1. Added RIL driver integration by library in Section 3.5. 2. Deleted how to modify the right of RILD. 3. Added how to configure device permissions in Section 3.6.2. 4. Added how to enable SELinux in Section 3.6.3.
1.7	2018-07-13	Macro GAO	Added Android 8.x in supported Android versions
1.8	2019-03-07	Macro GAO	1. Added Android 9.x in supported Android versions. 2. Added EC21, EC200T, EM06, EM12, EG12, AG36, UC200T& MC90 in supported products. 3. Updated the description of system configuration in Chapter 3. 4. Added how to configure SElinux in Chapter 4.
1.9	2020-05-11	Marco GAO	1. Updated the applicable modules in Chapter 1.1. 2. Updated the supported Android versions in

			Chapter 2.3. 3. Added the supported IRadio versions in Chapter 2.4. 4. Updated the description in Chapter 3.2. 5. Updated reference documents for USB driver installation of different modules in Chapter 3.3. 6. Updated Quectel RIL driver version for customers using Android 8.0 or later versions in Chapter 3.4. 7. Updated the description of service ril-daemon configuration in Chapter 3.5.1. 8. Updated HIDL description in Chapter 3.5.3. 9. Updated how to configure SElinux in Chapter 4. 10. Updated how to catch logs in Chapter 5. 11. Removed the section of Why Short Messages cannot be Sent or Received and added Why Phone Process Does not Work in Chapter 6.

Contents

Table Index

Figure Index

1 Introduction

This document mainly introduces how to integrate RIL (Radio Interface Layer) driver into Android OS of customers' target devices as well as how to modify the configuration files for starting RIL service.

1.1. Applicable Modules

The document is applicable to the following Quectel modules.

Table 1: Applicable Modules

LTE Standard Module Series	ECxx: EC25/ EC20 R2.1/ EC21/ EC200T
	EG9x: EG91/ EG95
	EG2x-G: EG21-G/ EG25-G
	EM05
LTE-A Module Series	Ex06: EM06/ EP06/ EG06
	Ex12: EM12/ EG12
Automotive Module Series	AGxx: AG35
LPWA Module Series	BGxx: BG95/ BG96
UMTS/HSPA(+) Module Series	UCxx: UC15/ UC200T
	UGxx: UG95/ UG96
GSM/GPRS/GNSS Module Series	Mxx: M66/ M72/ M95
	MCxx: MC60/ MC90
5G Module Series	RMxx: RM500Q-GL/ RM510Q-NA
	RGxx: RG500Q

NOTE

Quectel modules listed above may include multiple models. Please refer to the corresponding module specifications for details.

2 Overview of Android RIL Driver

2.1. Directory Structure

The file structure of Quectel RIL driver package is shown as below.

```
|-- arm64-v8a
|    |-- chat
|    |-- ip-down
|    |-- ip-up
|    `-- libreference-ril.so
|-- armeabi
|    |-- chat
|    |-- ip-down
|    |-- ip-up
|    `-- libreference-ril.so
|-- ql-ril.conf
`-- x86
     |-- chat
     |-- ip-down
     |-- ip-up
     `-- libreference-ril.so
```

Figure 1: Structure of RIL Driver Package

2.2. Supported Functions

Quectel RIL driver supports the following functions.

Table 2: Supported Functions

Functions	Support or Not
SMS	YES
Voice Call	YES
Data Service	YES
(U)SIM Tool Kit	NO
Phonebook	YES

2.3. Supported Android Versions

Presently, Quectel RIL driver supports the following Android versions.

Table 3: Supported Android Versions

Versions	Support or Not
Android 4.x	YES
Android 5.x	YES
Android 6.0	YES
Android 7.x	YES
Android 8.x	YES
Android 9.x	YES
Android 10.x	YES

2.4. Supported IRadio Versions

Quectel RIL driver supports the following IRadio versions.

Table 4: Supported IRadio Versions

Versions	Support or Not
IRadio 1.0	YES
IRadio 1.1	YES
IRadio 1.2	YES
IRadio 1.3	NO
IRadio 1.4	NO

3 RIL Integration

The chapter mainly describes the RIL driver structure and procedures of setting up an Android system with the RIL driver.

3.1. RIL Driver Structure

Android RIL provides the abstract layer between Android telephony service and radio hardware.

The following figure illustrates the RIL in the context of Android telephony architecture.

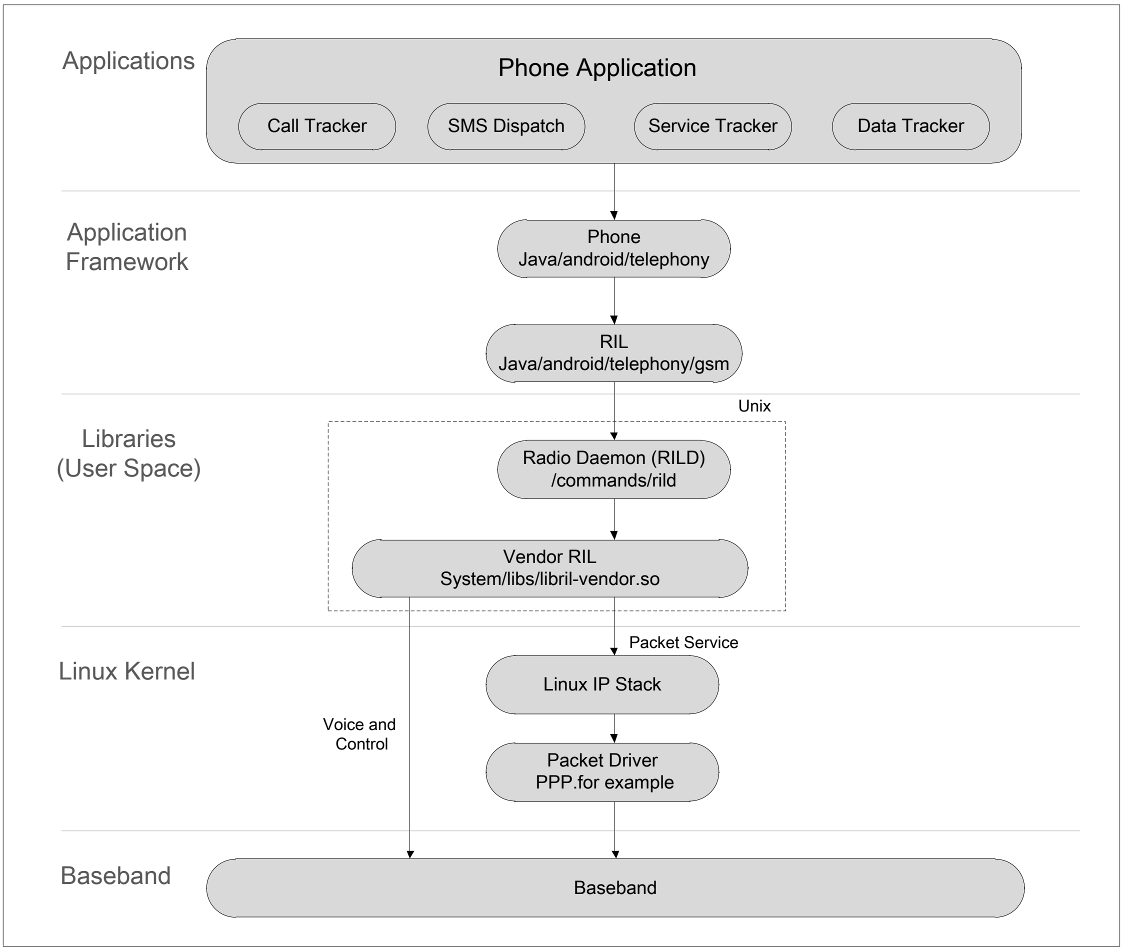

Figure 2: RIL Driver Structure

The RIL in Android is located between kernel and application framework. It is divided into two parts: RILD and Vendor RIL.

RILD is responsible for communications between socket and application framework.

Vendor RIL is responsible for radio communication via AT command channel and data communication via packet data channel (PDCH).

The java framework of RIL consists of two parts as well: RIL module and phone module. The RIL module communicates with the lower RILD while the phone module directly provides phone function interfaces to applications.

3.2. PPP Configuration in Linux Kernel

For modules accessed by UART interface, the kernel must be configured to support PPP dial-up. For modules accessed by USB interface, the kernel needs to be configured to support PPP dial-up only when the PPP function is used. For detailed operation procedures, please refer to ***Chapter 3.6*** in *Quectel_WCDMA&LTE_Linux_USB_Driver_User_Guide*.

3.3. USB Driver Installation in Linux Kernel

For modules accessed by USB interface, please integrate USB serial driver, CDC ACM driver, GobiNet driver or QMI_WWAN driver in Linux kernel. Otherwise, please skip this section.

For detailed operation procedures, please refer to the following documents：

UCxx/ UGxx:	*Quectel_WCDMA&LTE_Linux_USB_Driver_User_Guide*
EC2x/ EG9x/ EG2x-G/ EM05/ Ex06/ Ex12/ AGxx/ BGxx/ RMxx/ RGxx:	*Quectel_LTE&5G_Linux_USB_Driver_User_Guide*
EC200T:	*Quectel_EC200T_Linux_USB_Driver_User_Guide*
UC200T:	*Quectel_UC200T_Linux_USB_Driver_User_Guide*

3.4. RIL Driver Integration by Library

Please put the following RIL library files provided by Quectel into the Android system.

1. For Android versions prior to Android 8.0

- For 32-bit Android system

```
system/bin/chat
system/etc/ppp/ip-down
system/etc/ppp/ip-up
system/lib/libreference-ril.so
```

- For 64-bit Android system

```
system/bin/chat
system/etc/ppp/ip-down
system/etc/ppp/ip-up
system/lib64/libreference-ril.so
```

2. For Android 8.0 or later versions

- For 32-bit Android system

```
system/bin/chat
system/etc/ppp/ip-down
system/etc/ppp/ip-up
vendor/lib/libreference-ril.so
```

- For 64-bit Android system

```
system/bin/chat
system/etc/ppp/ip-down
system/etc/ppp/ip-up
vendor/lib64/libreference-ril.so
```

NOTE

For customers using Android 8.0 or later versions, Quectel RIL driver V2.0.0 or later versions should be used.

3.5. System Configuration

In order to use the RIL driver normally, some configuration files in Android system should be modified.

3.5.1. Configure Service ril-daemon

Service ril-daemon can be configured by adding the following lines to *init.rc* or *rild.rc*. The relevant lines will vary depending on the accessing interface and the android version being used.

3.5.1.1. Modules Accessed by USB Interface

For modules accessed by USB interface, add the following lines to *init.rc or rild.rc*:

- **For Android versions prior to Android 8.0**

```
service ril-daemon /system/bin/rild -l /system/lib/libreference-ril.so
    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
```

- **For Android 8.0 or later versions**

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

3.5.1.2. Modules Accessed by UART Interface

For modules accessed by UART interface, add the following lines to *init.rc or rild.rc*:

- **For Android versions prior to Android 8.0**

```
service ril-daemon /system/bin/rild -l /system/lib/libreference-ril.so -- -d <UART port name> -B <baud rate> -C <hardware flow control>
    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
```

- **For Android 8.0 or later versions**

```
service ril-daemon /vendor/bin/hw/rild -l /vendor/lib64/libreference-ril.so -- -d <UART port name> -B
<baud rate> -C <hardware flow control>
    class main
    user root
    group radio cache inet misc audio sdcard_rw log
    capabilities BLOCK_SUSPEND NET_ADMIN NET_RAW
```

Meanwhile, the following parameters need to be configured accordingly:

- **-d <UART port name>**

The UART port which is currently used. For example: /dev/ttyS1.

It is optional to configure the following two parameters:

- **-B <baud rate>**

The speed of UART port. Unit: bps. For example: 115200, 230400 or 460800. The default baud rate is 115200.

- **-C <hardware flow control>**

1: Enable the hardware flow control function
0: Disable the hardware flow control function (default setting)

The location of *init.rc* file varies according to customers' project settings. The following gives a non-exhaustive list of file paths which may contain the *init.rc* file.

- *device/fsl/imx6dq/sabresd_6dq/init.rc*
- *device/ti/am335xevm_sk/init.am335xevm.rc*
- *device/rockchip/rk3399/init.rk3399.rc*
- *device/samsung/smdkv210/init.smdkv210_sdmmc.rc*

3.5.2. Modify rild.c (Applicable to Android Versions Prior to Android 8.0)

For Android versions prior to Android 8.0, RILD (ril-daemon) requires root privileges, which can be achieved by commenting the function of *switchUser()* in the file *($Android_src)/hardware/ril/rild/rild.c*.

```
OpenLib:
#endif
    //switchUser();

    dlHandle = dlopen(rilLibPath, RTLD_NOW);
```

3.5.3. Add HIDL Description (Applicable to Android 8.0 or Later Versions)

On Android 8.0 or later versions, the communication interface between Android phone framework and ril-daemon service has changed from socket to HIDL. Therefore the following HIDL description needs to be added to *manifest.xml*.

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

The location of *manifest.xml* file varies according to customers' project settings. For example:

- *device/rockchip/rk3399/manifest.xml*
- *device/fsl/imx6dq/sabresd_6dq/manifest.xml*

4 SELinux Configuration

If the SELinux installed in customers' Android devices is enabled and runs in enforcing mode, follow the procedures below to make sure that Quectel RIL has full access to SELinux privileges. Otherwise, customers can skip this chapter.

4.1. Modify Service ril-daemon Configuration

To make sure Quectel RIL has full access to SELinux priviledges, the user of service ril-daemon should be radio. Therefore the user of service ril-daemon in the lines illustrated in ***Chapter 3.5.1*** should be changed from root to radio. An example with Android 8.0 for modules accessed by USB interface is shown below.

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

4.2. Uncomment the Function of switchUser()

To make sure Quectel RIL has full access to SELinux priviledges, the function of *switchUser()* mentioned in ***Chapter 3.5.2*** should be uncommented.

4.3. Configure SELinux Rule for RIL

The following definition should be added to *ueventd.rc*:

```
#quectel port
/dev/ttyUSB*        0660    radio   radio
/dev/ttyACM*        0660    radio   radio
/dev/cdc-wdm*       0660    radio   radio
/dev/qcqmi*         0660    radio   radio
/dev/cdc-acm*       0660    radio   radio
```

The following definition should be added to *file_contexts*:

```
/dev/ttyUSB[0-9]            u:object_r:radio_device:s0
/dev/ttyACM[0-9]            u:object_r:radio_device:s0
/dev/cdc-wdm[0-9]           u:object_r:radio_device:s0
/dev/qcqmi[0-9]             u:object_r:radio_device:s0
/vendor/bin/hw/rild         u:object_r:rild_exec:s0
/dev/socket/rildOemHook     u:object_r:rild_socket:s0
```

The following definition should be added to *rild.te*:

```
allow rild self:packet_socket { create bind write read };
```

The locations of *ueventd.rc, rild.te* and *file_contexts* may vary according to customers' project settings, but in general, these files are located in the following paths respectively by default.

- *device/fsl/imx6dq/sabresd_6dq/ueventd.freescale.rc*
- *device/fsl/imx6dq/sabresd_6dq/sepolicy/rild.te*
- *device/fsl/imx6dq/sabresd_6dq/sepolicy/file_contexts*
- *device/rockchip/common/ueventd.rockchip.rc*
- *device/rockchip/common/sepolicy/file_contexts*

5 Debugging Method

5.1. Catch Logs Automatically (Recommended)

Catching logs automatically is recommended for Quectel RIL driver and it can be achieved by fulfilling two prerequisites: disabling SELinux and applying the library files provided by Quectel. The detailed procedures are as below:

1) Set SELinux to permissive if it is in enforcing mode.

```
adb root
adb shell setenforce 0
```

2) Create folder */data/quectel_debug_log* on Android device and then restart it.

```
adb shell mkdir /data/quectel_debug_log
adb shell chmod 777 /data/quectel_debug_log
adb reboot
```

3) Get logs to local.

```
adb pull /data/quectel_debug_log
```

5.2. Catch Logs Manually

Quectel RIL driver also supports catching logs manually. The detailed procedures are as below:

1) Catch the logs of RIL module by typing the following command in Window's CMD tool:

```
adb logcat -b radio -v time
```

2) Catch the logs of Android system by typing the following command in Window's CMD tool:

```
adb logcat -v time
```

5.3. Common Log Tags

The following table lists some log tags that are commonly applied.

Table 5: Common Log Tags

RIL	*/hardware/ril/reference-ril/refereince-ril.c*
AT	*/hardware/ril/reference-ril/atchannel.c*
RILD	*/hardware/ril/rild/rild.c*
RILC	*/hardware/ril/libril/ril.cpp*
RILB	*/frameworks/base/telephony/java/com/android/internal/telephony/BaseCommands.java*
RILJ	*/frameworks/base/telephony/java/com/android/internal/telephony/gsm/RIL.java*
GSM	*/frameworks/base/telephony/java/com/android/internal/telephony/gsm/GSMPhone.java*

6 FAQs

6.1. How to Set the APN

If the dialling process is interrupted, it is quite possible that the APN has not been set yet. Set the APN in Android UI: "**Settings**" → "**WIRELESS & NETWORKS**" → "**…**" → "**Mobile Networks**" → "**Access Point Names**".

If nothing is found in "**Access Point Names**", it indicates that the APN has not been set. In such a case, customers need to add a new APN to the system. The following figure shows an example of the access point editing interface. Please note that the Access Point Name varies according to the operators and (U)SIM cards.

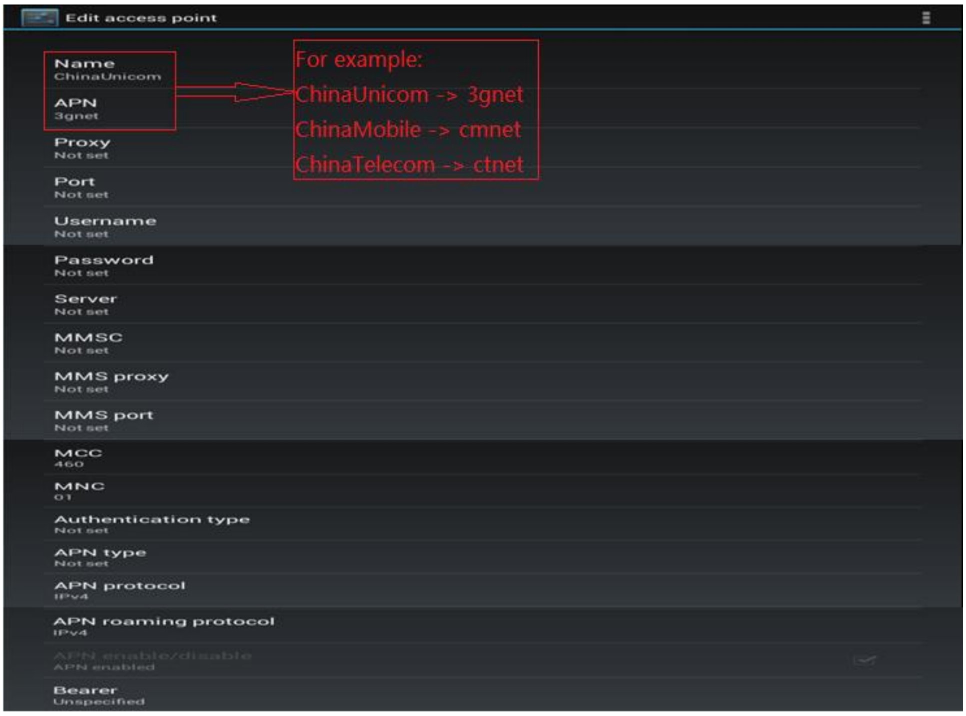

Figure 3: Edit Access Point

6.2. Why Quectel RIL Driver Does not Work

There are many reasons that may cause the failure of Quectel RIL operation. Some common causes are listed as below for troubleshooting.

1. RIL daemon is not running

Use command **getprop init.svc.ril-daemon** to check RIL daemon status. If no value is returned or **Stopped**/ **Restarting** is returned instead of **Running**, it indicates that RIL daemon is not running.

2. RIL library is not loaded correctly

Check the "ril-daemon" service definition in *init.rc* through command **cat /init*.rc | grep ril-daemon**. The expected result should be one of the following:

- **service ril-daemon /system/bin/rild -l /system/lib/libreference-ril.so**
- **service ril-daemon /system/bin/rild -l /system/lib64/libreference-ril.so**
- **service ril-daemon /system/bin/rild -l /vendor/lib/libreference-ril.so**
- **service ril-daemon /system/bin/rild -l /vendor/lib64/libreference-ril.so**.

Check the arguments, word spelling, blank space, etc. to make sure the RIL library is loaded correctly.

3. Cannot access USB serial port device file

- Use **ls -l /dev/ttyUSB*** command to check the access right of the device file.
- Use **getenforce** command to check whether the SELinux has been enabled. If yes, please use **setenforce 0** command to disable the SELinux.

4. The RIL library is not provided by Quectel

Use **getprop gsm.version.ril-impl** command to check the Quectel RIL version, and the returned value should start with **Quectel_Android_RIL_Driver_V**. If not, it indicates the RIL library is not provided by Quectel.

6.3. Why Phone Process Does not Work

Android system determines whether the system supports data access, phone, SMS and other features by configuring items. The configuration items and APK names are listed below. Please check whether the configuration is correct. Take imx6q as an example:

- If data access is required, the attribute *networkAttributes* must contain the following items:

```
<string-array translatable="false" name="networkAttributes">
    <item>"mobile,0,0,0,-1,true"</item>
    <item>"mobile_mms,2,0,4,60000,true"</item>
    <item>"mobile_supl,3,0,2,60000,true"</item>
    <item>"mobile_dun,4,0,2,60000,true"</item>
```

- If the phone feature is required, configure it as below:

```
<bool name="config_voice_capable">true</bool>
```

- If SMS is required, configure it as below:

```
<bool name="config_sms_capable">true</bool>
```

The above three configurations are in the file *config.xml.* The path can be, for example, *device/fsl/imx6dq/sabresd_6dq/overlay/frameworks/base/core/res/res/values/config.xml.*

- The following files must be installed on Android devices.

  /vendor/bin/hw/rild
  /vendor/lib/libril.so
  /system/priv-app/TeleService/TeleService.apk
  /system/priv-app/TelephonyProvider/TelephonyProvider.apk
  /system/framework/telephony-common.jar

7 Appendix A References

Table 6: Related Documents

SN	Document name	Remark
[1]	Quectel_WCDMA&LTE_Linux_USB_Driver_User_Guide	Linux USB driver user guide for WCDMA and LTE series modules
[2]	Quectel_LTE&5G_Linux_USB_Driver_User_Guide	Linux USB driver user guide for LTE and 5G series modules
[3]	Quectel_EC200T_Linux_USB_Driver_User_Guide	Linux USB driver user guide for EC200T series modules
[4]	Quectel_UC200T_Linux_USB_Driver_User_Guide	Linux USB driver user guide for UC200T

Table 7: Terms and Abbreviations

Abbreviation	Description
APN	Access Point Name
bps	Bit Per Second
HIDL	Hardware Interface Definition Language
PDCH	Packet Data Channel
PPP	Point-to-Point Protocol
RIL	Radio Interface Layer
RILD	Radio Interface Layer Daemon
SELinux	Security-Enhanced Linux
SMS	Short Message Service
UI	User Interface