

Android RIL Driver User Guide

GSM/GPRS/UMTS/HSPA/LTE Module Series

Rev. Android_RIL_Driver_User_Guide_V1.7

Date: 2018-07-13

Status: Released



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

Quectel Wireless Solutions Co., Ltd.

7th Floor, Hongye Building, No.1801 Hongmei Road, Xuhui District, Shanghai 200233, China

Tel: +86 21 5108 6236

Email: info@quectel.com

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

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

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

<http://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.

COPYRIGHT

THE INFORMATION CONTAINED HERE IS PROPRIETARY TECHNICAL INFORMATION OF QUECTEL WIRELESS SOLUTIONS CO., LTD. TRANSMITTING, REPRODUCTION, DISSEMINATION AND EDITING OF THIS DOCUMENT AS WELL AS UTILIZATION OF THE CONTENT ARE FORBIDDEN WITHOUT PERMISSION. OFFENDERS WILL BE HELD LIABLE FOR PAYMENT OF DAMAGES. ALL RIGHTS ARE RESERVED IN THE EVENT OF A PATENT GRANT OR REGISTRATION OF A UTILITY MODEL OR DESIGN.

Copyright © Quectel Wireless Solutions Co., Ltd. 2018. All rights reserved.

About the Document

History

Revision	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

Contents

About the Document.....	2
Contents	3
Table Index.....	4
Figure Index	5
1 Introduction	6
1.1. Applicable Products	6
2 Overview of Android RIL Driver	7
2.1. Directory Structure	7
2.2. Supported Functions	7
2.3. Supported Android Versions	8
3 RIL Integration.....	9
3.1. RIL Driver Structure.....	9
3.2. PPP Configuration in Linux Kernel	10
3.3. Driver Installation	10
3.4. RIL Driver Integration by Library	10
3.5. System Configuration.....	11
3.5.1. Configure "init.rc".....	11
3.5.2. Configure Device Permission.....	13
3.5.3. Enable SELinux (Only 5.0 or Later Versions are Supported).....	13
4 Debugging Method	14
4.1. Catch Logs	14
4.2. Common Log Tags	14
5 FAQs.....	16
5.1. How to Set the APN	16
5.2. Why Quectel RIL Driver Does not Work	17
5.3. Why Short Messages cannot be Sent or Received.....	17
6 Appendix A References.....	18

Table Index

TABLE 1: APPLICABLE MODULES.....	6
TABLE 2: SUPPORTED FUNCTIONS	7
TABLE 3: SUPPORTED ANDROID VERSIONS	8
TABLE 4: COMMON LOG TAGS.....	14
TABLE 5: RELATED DOCUMENTS.....	18
TABLE 6: TERMS AND ABBREVIATIONS	18

Figure Index

FIGURE 1: STRUCTURE OF RIL DRIVER PACKAGE	7
FIGURE 2: RIL DRIVER STRUCTURE.....	9
FIGURE 3: EDIT ACCESS POINT	16

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 Products

The document is applicable to the following Quectel LTE modules.

Table 1: Applicable Modules

LTE Module Series	EC2x: includes EC20/EC25/EC20 R2.0/EC20 R2.1 modules
	EG9x: includes EG91/EG95 modules
	Ex06: includes EP06/EG06 modules
	AG35 module
UMTS/HSPA(+) Module Series	UCxx: includes UC15/UC20 modules
	UGxx: includes UG95/UG96 modules
GSM/GPRS Module Series	Mxx: includes M66/M72/M80/M85/M95/MC60 modules

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

Table 2: Supported Functions

Function	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

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 illustrates the RIL in the context of Android telephony architecture.

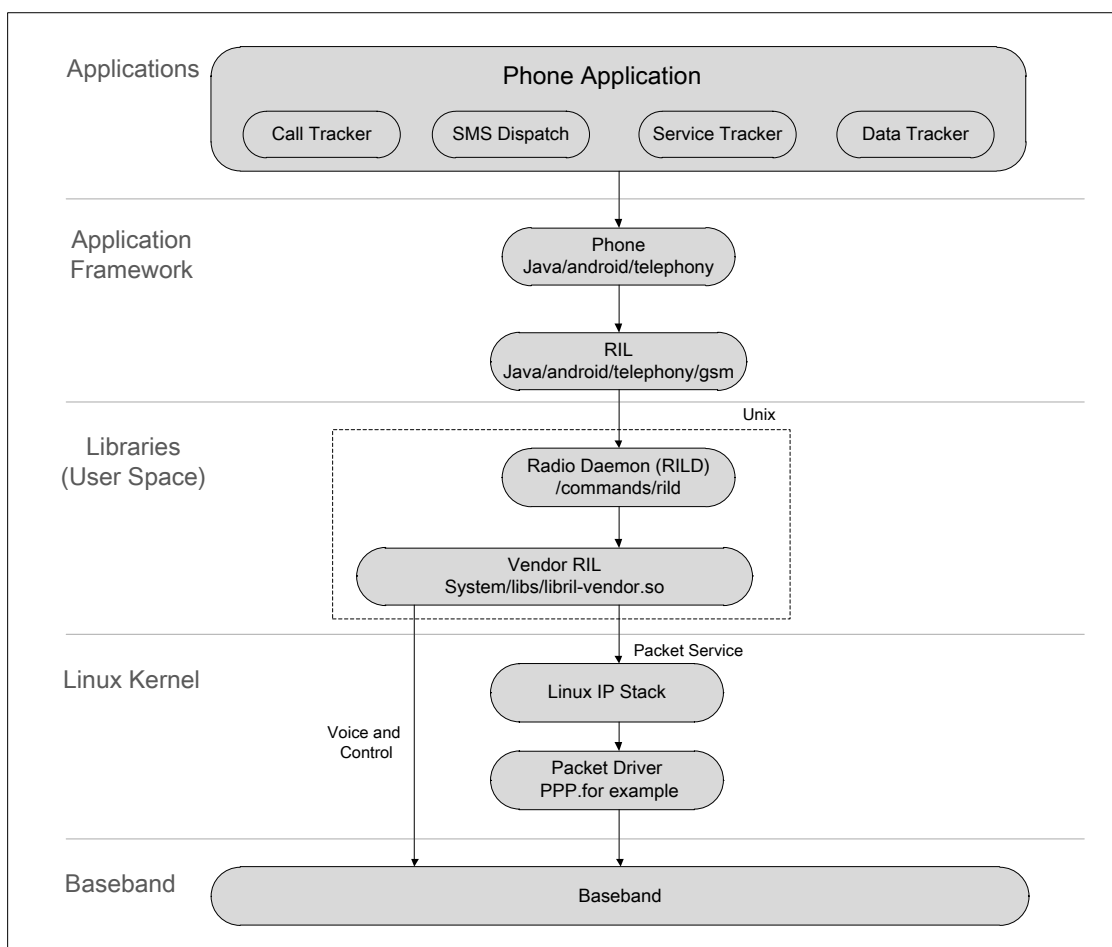


Figure 2: RIL Driver Structure

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 with packet data channel (PDCH).

The java framework of RIL consists of two parts as well. One is RIL module and the other is Phone module. The RIL module is used to communicate with the lower RILD, and the phone module directly provides phone function interfaces to application.

3.2. PPP Configuration in Linux Kernel

The kernel needs to be configured to support PPP dial up. For detailed operation procedures, please refer to **Chapter 3.6** in document *Quectel_WCDMA<E_Linux_USB_Driver_User_Guide*.

3.3. Driver Installation

If customers are using EC2x/EG9x/Ex06/AG35/UCxx/UGxx modules, 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 **Chapter 3.2, Chapter 3.3, Chapter 3.4 or Chapter 3.5** in *Quectel_WCDMA<E_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.

- For Android 8.0 version and below

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

- For Android 8.1 version and above

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

NOTE

If customers are using Android 8.1 version and above, Quectel RIL driver version V1.42.08 or above 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 "init.rc"

Add "ril-daemon" service in "init.rc":

If customers are using EC2x/EG9x/Ex06/AG35/UCxx/UGxx modules which are accessed by USB interface, please add the following service in "init.rc":

- For Android 8.0 version and below

```
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 radio  
    group radio cache inet misc audio sdcard_rw log
```

- For Android 8.1 version and above

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

If customers are using GSM modules which are accessed by UART interface, please add the following service in "init.rc":

- **For Android 8.0 version and below**

```
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 radio
    group radio cache inet misc audio sdcard_rw log
```

- **For Android 8.1 version and above**

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

Meanwhile, the following parameter needs to be configured accordingly:

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

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. For example: 115200bps, 230400bps or 460800bps. The default baud rate is 115200bps.

- **-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 with different modules, the paths may be one of the followings:

- system/core/rootdir/init.rc
- device/fsl/imx6/init.rc
- device/ti/am335xevm_sk/init.am335xevm.rc
- device/generic/x86/init.rc
- device/samsung/smdkv210/init.smdkv210_sdmmc.rc

3.5.2. Configure Device Permission

RILD (ril-daemon) requires radio privilege, so the file "(\$Android_src)/system/core/rootdir/ueventd.rc" needs to be modified accordingly as follows:

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

3.5.3. Enable SELinux (Only 5.0 or Later Versions are Supported)

If SELinux need to be operated, then it needs to be enabled by modifying "(\$Android_src)/external/sepolicy/file_contexts" and "(\$Android_src)/external/sepolicy/rild.te" files.

Add the following codes in "(\$Android_src)/external/sepolicy/file_contexts" file:

```
/dev/ttyUSB[0-9]*      u:object_r:tty_device:s0
/dev/ttyACM[0-9]*      u:object_r:tty_device:s0
/system/bin/rild        u:object_r:rild_exec:s0
/system/socket/rild     u:object_r:rild_socket:s0
/system/socket/rild-debug u:object_r:rild_debug_socket:s0
/system/bin/pppd        u:object_r:pppd_exec:s0
/dev/ppp               u:object_r:ppp_device:s0
```

Add the following codes in "(\$Android_src)/external/sepolicy/rild.te" file:

```
allow rild default_prop:property_service set;
allow rild device:chr_file { read write ioctl open getattr };
allow rild kernel:system module_request;
allow rild net_radio_prop:property_service set;
allow rild ppp_device:chr_file { read write ioctl open };
allow rild ppp_exec:file { read execute open execute_no_trans };
allow rild radio_prop:property_service set;
allow rild self:capability { net_admin setuid };
allow rild shell_exec:file { read execute open execute_no_trans };
allow rild sysfs_wake_lock:file { open read write };
allow rild system_file:file execute_no_trans;
allow rild system_prop:property_service set;
```

4 Debugging Method

4.1. Catch Logs

- 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
```

- 3) Sometimes, customers may want to perform tests on lots of devices or for a long time, and it is not convenient to connect all devices with PC via USB cables. In such case, customers can catch the log files by following command:

```
adb shell  
logcat -b radio -v time -f <filename> &
```

The character "&" makes the "logcat" process run in the background, thus customers' devices can be disconnected.

- 4) When customers' tests are finished, the log files can be fetched from devices to a local directory by the following command:

```
adb pull <filename> <local directory>
```

4.2. Common Log Tags

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

Table 4: 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

5 FAQs

5.1. How to Set the APN

If the dialling process is interrupted, it is quite possible that the APN has not been set yet. Please 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 case, customers need to add a new APN to the system. The following is an example shows the access point editing interface. Please note that the Access Point Name varies with the operators and (U)SIM cards.

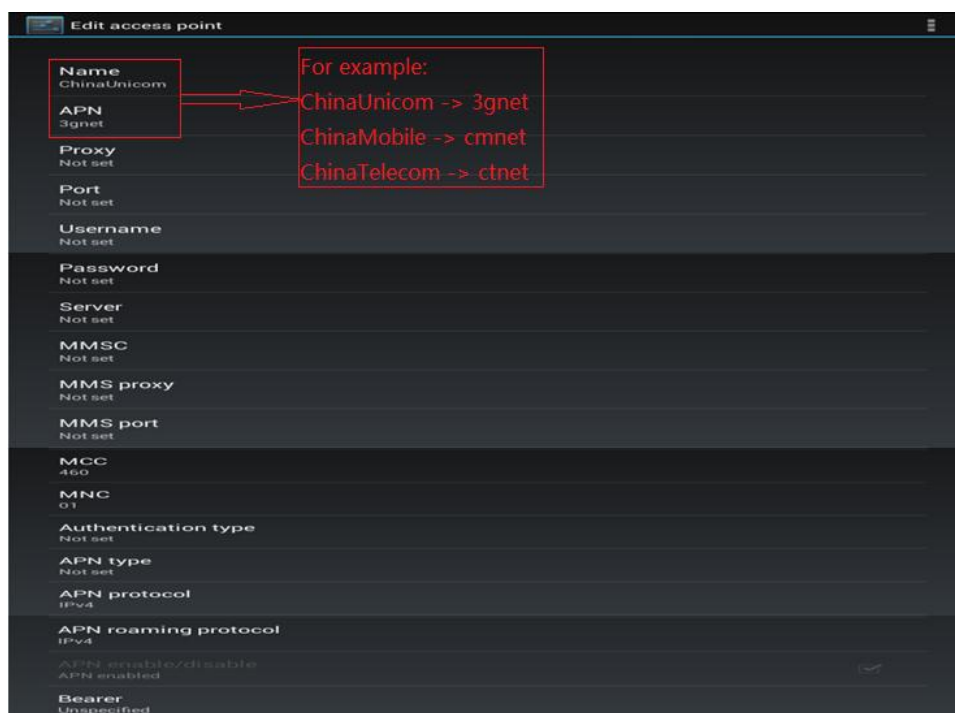


Figure 3: Edit Access Point

5.2. 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 non-value, **Stopped** or **Restarting** is returned, rather than **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 “**service ril-daemon /system/bin/rild -l /system/lib/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 return value should be started with “**Quectel_Android_RIL_SR**”. If not, then it indicates the RIL library is not provided by Quectel.

5.3. Why Short Messages cannot be Sent or Received

Use “**getprop telephony.sms.receive**” command or “**telephony.sms.send**” command to check whether the system has disabled SMS receive/send function.

- If the return value is false, it indicates that the SMS receive/send function has been disabled.
- If non-value is returned, please check *frameworks\base\core\res\res\values\config.xml* (the manufacturer override configure file) and find *config_sms_capable*. If the *config_sms_capable* configuration in the *config.xml* is false (e.g. `<bool name="config_sms_capable">false</bool>`), please change it to true.

6 Appendix A References

Table 5: Related Documents

SN	Document name	Remark
[1]	Quectel_WCDMA<E_Linux_USB_Driver_User_Guide	Linux USB driver user guide for WCDMA and LTE series modules

Table 6: Terms and Abbreviations

Abbreviation	Description
GSM	Global System for Mobile Communications
MS	Mobile Station
PID	Product ID
RIL	Radio Interface Layer
VID	Vendor ID
WCDMA	Wideband Code Division Multiple Access