

Android RIL Driver User Guide

GSM/GPRS/UMTS/HSPA/LTE Module Series

Rev. Android_RIL_Driver_User_Guide_V1.6

Date: 2017-11-10



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. 2017. 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	<ol style="list-style-type: none">1. Added GSM modules in Supported Products2. Added Android 5.x in Supported Android Versions
1.4	2016-06-21	Carl YIN/ Neo HOU	<ol style="list-style-type: none">1. Added EC21&EC25 in Supported Products2. Added FAQ
1.5	2016-08-23	Kent XU	Added EC20 R2.0 in Supported Products
1.6	2017-11-10	Kent XU	<ol style="list-style-type: none">1. Deleted Chapter 1.1 which was used to show the Directory Structure of Quectel RIL Driver Package2. Updated description of Supported Products, and added EG91, EG95, EG06, EM06, EP06, BG96, and AG35 in Supported Products3. Added Android 6.x/7.x in Supported Android Versions

Contents

About the Document	2
Contents	3
Table Index.....	4
Figure Index	5
1 Introduction	6
2 Overview of Android RIL Driver	7
2.1. Supported Products	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. USB Drivers Integration	10
3.4. RIL Driver Integration	10
3.5. System Configuration.....	11
3.5.1. Configure init.rc	11
3.5.2. Modify the Right of RILD	12
4 Debugging Method	13
4.1. Method of Catching LOG	13
4.2. Some Common Log Tags	13
5 FAQs.....	15
5.1. How to Set APN	15
5.2. Quectel RIL Does Not Work.....	16
5.3. Cannot Send or Receive Short Message	16
6 Appendix.....	17

Table Index

TABLE 1: SUPPORTED PRODUCTS.....	7
TABLE 2: SUPPORTED FUNCTIONS.....	7
TABLE 3: SUPPORTED ANDROID VERSIONS.....	8
TABLE 4: SOME COMMON LOG TAGS.....	13
TABLE 5: TERMS AND ABBREVIATIONS.....	17

Quectel
Confidential

Figure Index

FIGURE 1: RIL DRIVER STRUCTURE.....	9
FIGURE 2: RIL FILES.....	11
FIGURE 3: EDIT ACCESS POINT	15

Quectel
Confidential

1 Introduction

This document mainly introduces how to integrate RIL Driver into Android OS of your target machine and how to modify the configuration files for starting RIL service and PPP dialling.

This document is applicable to Quectel GSM, UMTS and LTE modules. More details are provided in *Table 1*.

Quectel
Confidential

2 Overview of Android RIL Driver

2.1. Supported Products

Table 1: Supported Products

Product	Description	
GSM	All Quectel GSM modules	
UMTS	UCxx	UC15/UC20
	UGxx	UG95/UG96
LTE	EC2x	EC20/EC21/EC25
	EGxx	EG91/EG95/EG06
	Others	EP06/EM06/BG96/AG35

2.2. Supported Functions

Table 2: Supported Functions

Function	Support or Not
SMS	YES
Voice Call	YES
Data Service	YES
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 2.x	YES
Android 3.x	YES
Android 4.x	YES
Android 5.x	YES
Android 6.x	YES
Android 7.x	YES

Quectel
Confidential

3 RIL Integration

This chapter mainly introduces the RIL driver structure and how to set up Android system with the RIL driver.

3.1. RIL Driver Structure

Android RIL (Radio Interface Layer) provides the abstract layer between Telephony service and Radio hardware.

The following illustration describes the RIL's position in the Android architecture.

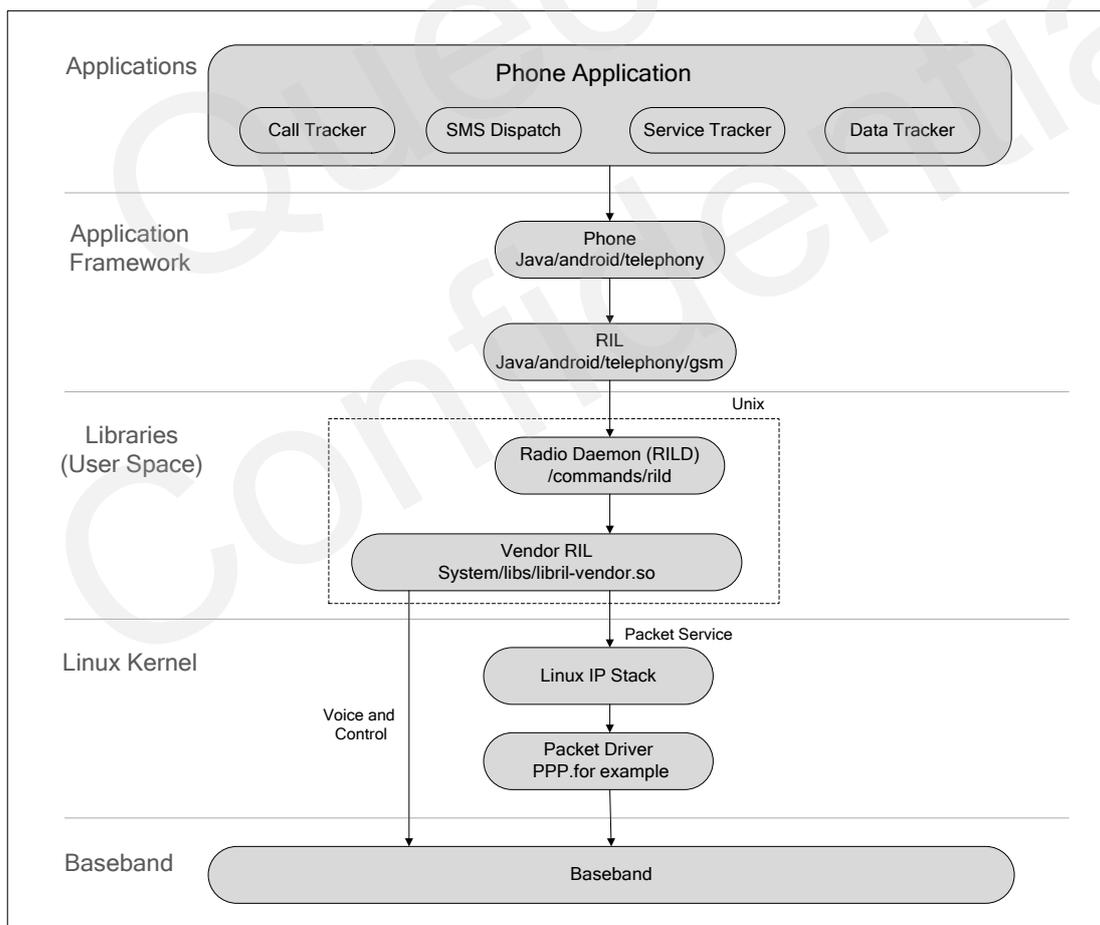


Figure 1: RIL Driver Structure

The RIL in Android is located between Kernel and Application Framework. It is divided into two parts, one is RILD and the other is Vendor RIL.

RILD is responsible for the communication between Socket and Application Framework.

Vendor RIL is responsible for communication with Radio via AT command channel and Packet data channel (PDCH). AT command channel is used for communicating with Radio directly and PDCH is used for data service.

The java framework of RIL is divided into two parts too. One is RIL module and the other is Phone module. The RIL module is used to communicate with the lower RILD. The Phone module directly provides phone function interfaces to application through which customers can call to realize the phone functions.

3.2. PPP Configuration in Linux Kernel

The kernel needs to be configured to support PPP dial up.

For detailed kernel configuration (to support PPP) steps, please refer to **Chapter 3.6** of *Quectel_WCDMA<E_Linux_USB_Driver_User_Guide*.

3.3. USB Drivers Integration

If Quectel GSM modules are used, please skip this chapter.

The kernel should be configured to support UCxx/UGxx/EC2x/EGxx/EP06/EM06/BG96/AG35 modules. For detailed operation procedures, please refer to **Chapter 3.2~3.5** in *Quectel_WCDMA<E_Linux_USB_Driver_User_Guide*:

3.4. RIL Driver Integration

Presently, Quectel provides RIL driver in the form of source code. There is only a need to copy the RIL driver source code files to the correct path on your project directory, and then recompile the Android system.

The source path of the RIL driver files in RIL Driver package is: *Driver package/reference-ril*

The destination path in Android system is: *(\$Android_src)/hardware/ril/*

Please use the following command to update these files' date stamp to make sure they will be compiled.

```
touch hardware/ril/*
```

After recompiling the Android system successfully, please make sure the following files exists in folder `out/target/product/($your_board_name)/system`.

```
-rwxr-xr-x 1 root root 62884 2017-11-06 10:13:44.579951421 +0800 /system/bin/chat
-rwxr-xr-x 1 root root 8716 2017-11-06 10:13:03.039951426 +0800 /system/etc/ppp/ip-down
-rwxr-xr-x 1 root root 9404 2017-11-06 10:13:03.039951426 +0800 /system/etc/ppp/ip-up
-rw-r--r-- 1 root root 726192 2017-11-06 10:13:31.107951423 +0800 /system/lib/libreference-ril.so
```

Figure 2: RIL Files

NOTE

Quectel would not give the PPP script, because we have integrated the PPP script in the source code of the RIL driver. And the chat, ip-up and ip-down would be generated when the RIL driver has been compiled.

3.5. System Configuration

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

3.5.1. Configure `init.rc`

Add service `rild-daemon` in `init.rc`:

If UCxx/UGxx/EC2x/EGxx/EP06/EM06/BG96/AG35 modules are used and they are accessed by USB interface, please add the following service in `init.rc`:

```
service rild-daemon /system/bin/rild -l /system/lib/libreference-ril.so
    class main
    socket rild stream 660 root radio
    socket rild-debug stream 666 radio system
    user root
    group radio cache inet misc audio sdcard_rw log
```

If GSM modules are used and they are accessed by UART interface, please add the following service in `init.rc`:

```
service rild-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 666 radio system
user root
group radio cache inet misc audio sdcard_rw log
```

The following parameter needs to be configured:

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

The name of UART port being used. For example: `/dev/ttyS1`

It is optional to configure the following two parameters:

- **-B <baud rate>**

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

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

1: open hardware flow control

0: close hardware flow control

Hardware flow control is closed by default.

The `init.rc` file may be located in different paths according to different projects, for example:

- `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. Modify the Right of RILD

RILD (ril-daemon) requires root privilege. So there is a need to comment out the function `switchUser()` in the file `($Android_src)/hardware/ril/rild/rild.c`:

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

dlHandle = dlopen(rilLibPath, RTLD_NOW)
```

4 Debugging Method

4.1. Method of Catching LOG

- 1) Catch the log of RIL module by typing the following command in window's CMD tool:

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

- 2) Catch the log of Android system by typing the following command in window's CMD tool:

```
adb logcat -v time
```

- 3) Sometimes, customers may want to make 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 this case, they can log file by the following command:

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

The character '&' makes the 'logcat' process running in background, so the devices can be disconnected.

- 4) When tests are finished, the following command can be used to fetch log files from devices to local directory:

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

4.2. Some Common Log Tags

Table 4: Some 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

Quectel
Confidential

5 FAQs

5.1. How to Set APN

If the dialling process cannot go on, it is quite possible that the APN has not been set yet. The APN can be set in Android UI:

Settings → WIRELESS & NETWORKS → More... → Mobile Networks → Access Point Names

If nothing is seen in **Access Point Names**, it indicates that the APN has not been set. In such case, a new APN should be added to the system. The following is an example which shows the access point editing interface. Please note that the Access Point Name varies with the operator and (U)SIM card.

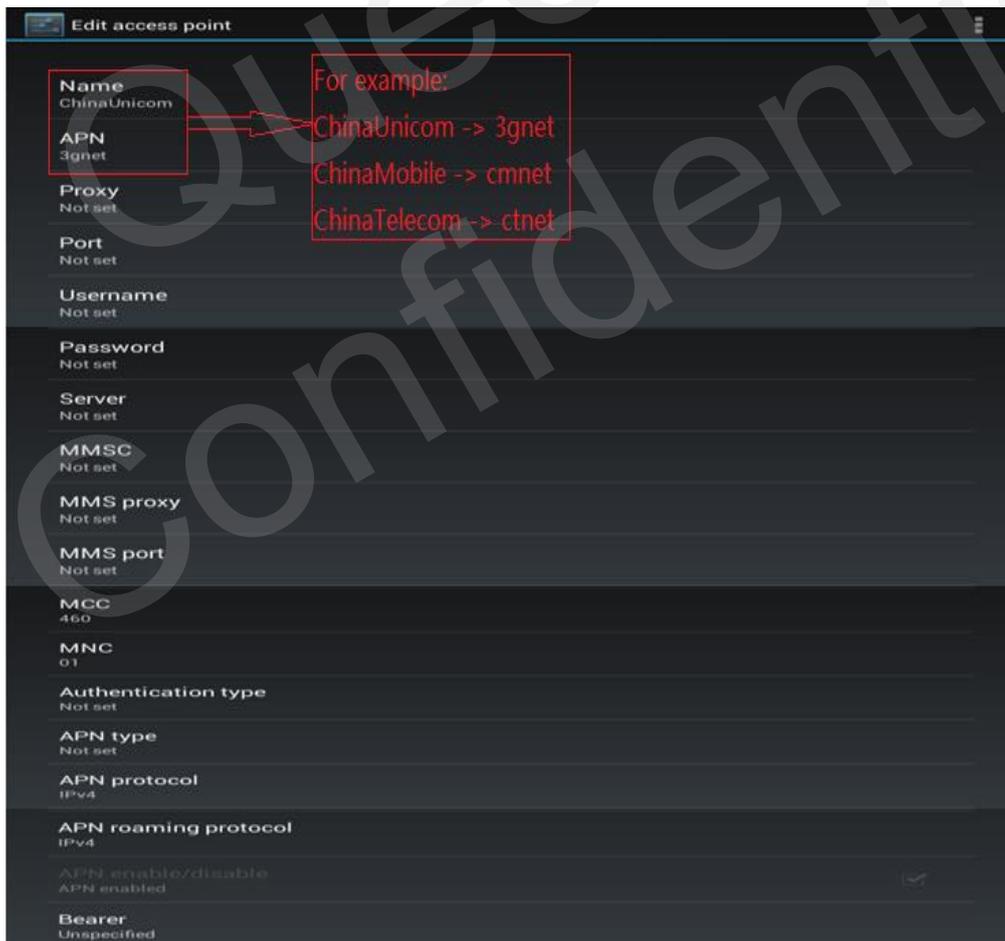


Figure 3: Edit Access Point

5.2. Quectel RIL Does Not Work

There are many reasons that may cause Quectel RIL not working. The following are some common causes.

1. RIL daemon is not running

Use the command below to check RIL daemon status. If non-value, “Stopped” or “Restarting” is returned, it indicates that RIL daemon is not running. The expected return value should be “Running”.

```
getprop init.svc.ril-daemon
```

2. RIL library is not loaded correctly

Check the RIL-daemon service definition in *init.rc* through the command below. 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.

```
cat /init.rc | grep ril-daemon
```

3. Cannot access USB serial port device file

- Use command “ls -l /dev/ttyUSB*” to check the access right of the device file.
- Use command “getenforce” to check whether the SELinux has been enabled.
- Command “setenforce 0” can disable the SELinux.

4. The RIL library is not provided by Quectel

Use the command below to check Quectel RIL version, and the return value should start with *Quectel_Android_RIL_SR*. If not, the RIL library is not provided by Quectel.

```
getprop gsm.version.ril-impl
```

5.3. Cannot Send or Receive Short Message

Use the following command to check whether the system has disabled SMS receive/send function.

```
getprop telephony.sms.receive
```

Or

```
telephony.sms.send
```

- 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* (first check the manufacturer override configuration file) to see *config_sms_capable*. If the *config_sms_capable* configuration file in *config.xml* is false just like `<bool name="config_sms_capable">false</bool>`, then change it to true.

6 Appendix

Table 5: Terms and Abbreviations

Abbreviation	Description
GSM	Global System for Mobile Communications
MS	Mobile Station
PID	Product ID
RIL	Radio Interface Layer
TA	Terminal Adapter
VID	Vendor ID
WCDMA	Wideband Code Division Multiple Access
LTE	Long Term Evolution