# Android RIL

## Source code

Application note

**Abstract**

This document describes how to build and use the RIL library for u-blox cellular modules on the Android operating system.

# Document information

| Title | Android RIL | |
|---|---|---|
| Subtitle | Source code | |
| Document type | Application note | |
| Document number | UBX-13002041 | |
| Revision and date | R29 | 29-Jul-2021 |
| Disclosure restriction | C1-Public | |

This document applies to the following products:

| Product name |
|---|
| TOBY-L2 series |
| TOBY-L4 series |
| TOBY-R2 series |
| LARA-R2 series |
| SARA-R4 series |
| SARA-R5 series |
| LISA-U2 series |
| SARA-U2 series |
| LISA-C2 series |
| LEON-G1 series |
| SARA-G3 series |
| SARA-G4 series |

# Contents

# 1   Introduction

☞   An index finger points out key information pertaining to integration and performance.

⚠   A warning symbol indicates actions that could negatively impact performance or damage the device.

☞   This revision of the application note applies to the Android RIL v12.00 and onwards.

The software was developed for the following Android versions (AOSP) and platforms:

- Android 5.x (Lollipop): Nexus 5
- Android 6.x (Marshmallow): Nexus 5
- Android 7.x (Nougat): Nexus 5X
- Android 8.x (Oreo): Nexus 5X
- Android 9.x (Pie): Google Pixel 2
- Android 10.x: Google Pixel 2
- Android 11.x: Hikey960

See the RIL release note for the list of Android software deliveries and interfaces supported by u-blox cellular modules.

In the following chapters, <name> indicates a parameter that can:

- be customized
- be set corresponding to the system configuration
- provide software version

The following chapters describe the production packages delivered by u-blox.

An overview of the system setup is provided as well as the procedure to perform a log.

The following symbols are used to highlight important information within this document:

☞   The RIL source code provided by u-blox is for reference purposes only.

☞   u-blox RIL does not control or manage the device (for example, u-blox module or any other peripheral) GPIOs, such as to power off/on or reset the module. Power management is under the responsibility of the Android OS and the kernel subsystem.

⚠   u-blox assumes no responsibility for inappropriate use of RIL by customers.

# 2   Production delivery

## 2.1   Software release

The delivery consists of the RIL library source code.

⚠️   Android RIL does not offer a standard interface for powering off the module, so it is necessary to modify the Android Java framework application that manages system power-off and sends the AT+CPWROFF command.

## 2.2   Android delivery contents

☞   The Android delivery contents structure for all Android versions is described below. If the `RIL_sc_<version>.zip` does not include a component mentioned in the delivery contents, it means that the package has been updated for that release and the component is no longer required.

### 2.2.1   Android 5.x, 6.x, 7.x, 8.x, 9.x, 10.x, 11.x delivery contents

The RIL production delivery is provided in the `RIL_sc_<version>.zip` compressed file with the content structured as follows:

```
RIL_sc_<version>.zip
   Android_<version>/                  Files specific for Android version
      device/                         Platform specific device files

      system/                         system files

      external/ppp/
         chat/                        PPP chat files
         pppd/                        Data connection files for PPP

      packages/apps/Settings
         src/                         OEM Hook app src files
         res/                         OEM Hook app res files

   Common/                            Common files for all Android versions

      hardware/ril/ublox_ril/         RIL core directory

      system/core/rootdir
         init.ublox.rc                u-blox Init.rc file

      build/target/product            Device configuration folder files
         core_ublox.mk                Make file include all scripts
         rootdir/etc                  RIL data scripts for PPP and RNDIS modes
         rootdir/etc/ppp              Scripts for data connection
         rootdir/etc/uril             RIL configuration repository file
```

# 3   Build source code

☞   This section applies to the Linux operating system.

- Create the build directory using the following commands, where `<android_root>` is the directory name:
```
mkdir <android_root>
cd <android_root>
```

- Download an Android distribution for the platform. Follow the tutorial, manual, etc. related to the platform's distribution.

⚠   Android distribution must be one of the deliveries listed in section 1. Full functionality of software in other Android OS versions is not guaranteed.

- Create a directory and decompress the RIL file.

⚠   The password provided from u-blox to decrypt the compressed RIL file is needed.
```
mkdir <ril_dir>
cd <ril_dir>
cp <path_of_ril>/RIL_sc_<version>.zip .
unzip –P <ril_password> RIL_sc_<version>.zip
```

- Copy the extracted RIL source code into the Android distribution. From here on, this path will be referred to as `ril_sc_<version>`.

- Before beginning the RIL integration, run the following command inside the RIL source code directory `ril_sc_<version>`:
```
find . -type f -exec dos2unix {} \;
```

- For Android 5.x / 6.x / 7.x / 8.x / 9.x / 10.x / 11.x delivery:
```
cp –pvRf ril_sc_<version>/Android_<version>/external/ppp/* <android_root>/external/ppp

cp –pvRf ril_sc_<version>/Android_<version>/system/* <android_root>/system

cp –pvRf ril_sc_<version>/Common/build/target/product/*
<android_root>/build/target/product

cp –pvRf ril_sc_<version>/Common/system/core/rootdir/*
<android_root>/system/core/rootdir

cp –pvRf ril_sc_<version>/Common/hardware/ril/ublox_ril <android_root>/hardware/ril/
```

⚠   Files provided in folder "`ril_sc_<version>/Android_<version>/`" are for reference only and extracted/modified from AOSP of Google devices like Nexus5X and Pixel2. These Android files should be compared first and then replaced. If any other conflict is present, then avoid replacing it via "`cp –pvRf`" and instead manually add the changes tagged with "u-blox modifications". Otherwise, it can cause serious issues in compilation if code base is different.

☞   The delivery overwrites or adds some Android system files.

- Modify the kernel configuration as in appendix A and configure the script files module connection to the Android platform as described in:
  o   Appendix B: Android 5.x/6.x/7.x/8.x/9.x/10.x/11.x module configuration

☞   For module specific modifications, see appendix I.

☞   Set command line arguments of rild and MUX services in `repository.txt`.

☞   For information on MUX interfacing, see the Multiplexer implementation application note [5].

- Build the Android system for the device or for the emulator.

☞   Use the tutorial, manual, etc. related to the platform's distribution.

- Insert a microSD card (minimum 2 GB) into the PC. Determine the SD card device name. Prompt `dmesg` command on a system shell. An example of the result of this command is below
```
[85560.292608] sd 6:0:0:2: [sdd] Assuming drive cache: write through
[85560.292613] sd 6:0:0:2: [sdd] Attached SCSI removable dis
```

The device name is written in the brackets (for example, sdd)

- The next step partitions and populates the microSD card with the root file system, bootloader, etc.
- Use the command provided in the platform distribution to create an SD card for the platform

⚠ This step can erase the hard disk drive if the SD card device name entered after the script name is incorrect.

- Insert the SD card into the board's SD slot
- Connect the board with the u-blox cellular module using one of the supported interfaces
- Power on both devices

# 4 Debugging RIL

This section explains some debugging methods to test that RIL is working.

☞ If using RIL in MUX mode, see the Multiplexer implementation application note [5] for MUX related debugging information.

## 4.1 Host PC configuration

### 4.1.1 Windows configuration

- On a Windows host, download and extract the Android SDK [4]
- Update the Android SDK using the command:
  ```
  <home>\android-sdk-windows\tools\android.bat
  ```

- Download the USB driver for Android [3] and extract it into
  ```
  <home>\android-sdk-windows\usb-drivers\
  ```
- Modify the section [Google.NTx86] of the `android_winusb.inf` in the `<home>\android-sdk-windows\usb-drivers\usb_driver_r03-windows` directory with the following lines:
  ```
  ;
  ;TI EVM
  %SingleAdbInterface%        = USB_Install, USB\VID_18D1&PID_9018
  %CompositeAdbInterface%     = USB_Install, USB\VID_18D1&PID_9018&MI_01
  ```

- Run a shell and type the following command:
  ```
  echo 0x18D1 > "%USERPROFILE%\.android\adb_usb.ini"
  ```

- Install the downloaded USB drivers for the Android OS
- Connect the device to the PC
- The device is now ready to start communications

### 4.1.2 Linux configuration

- On a Linux host, download and extract the Android SDK [4]
- On Linux, type the following commands in a shell:
  ```
  sudo su <passwd>
  mkdir ~/.android
  vi ~/.android/adb_usb.ini
  echo "0x0451" > ~/.android/adb_usb.ini
  ```

- Connect the device to the PC
- The device is now ready to start communications

## 4.2   RIL log

Set up the Android Debugger (ADB) as mentioned above. By means of the ADB, it is possible to choose which RIL radio- and telephony-related log messages to receive by using the following command:

```
adb logcat  -v time -b radio | tee radio.log
```

The above command generates a log output like the following example:

```
02-24 01:55:22.034 D/RILD    (  512): **RIL Daemon Started**
02-24 01:55:22.034 D/RILD    (  512): **RILd param count=6**
02-24 01:55:22.068 W/RILD    (  512): RIL_SAP_Init not defined or exported in
/system/lib64/librapid-ril-core.so: undefined symbol: RIL_SAP_Init
02-24 01:55:22.069 D/RILD    (  512): RIL_Init argc = 5 clientId = 0
02-24 01:55:22.070 W/RILC    (  512): RIL_onUnsolicitedResponse called before RIL_register
02-24 01:55:22.081 I/use-Rlog/RLOG-RILR(  512): Log level [2]
02-24 01:55:22.081 I/use-Rlog/RLOG-RILR(  512): mainloop() - RIL Daemon Version: u-blox RIL 11.03
02-24 01:55:22.081 I/use-Rlog/RLOG-RILR(  512): mainLoop() - RIL Interface USB is selected.
02-24 01:55:22.092 I/use-Rlog/RLOG-RILR(  512): CSystemManager::CSystemManager() - Enter
02-24 01:55:22.092 I/use-Rlog/RLOG-RILR(  512): CSystemManager::CSystemManager() - Exit
02-24 01:55:22.092 I/use-Rlog/RLOG-RILR(  512): CSystemManager::InitializeSystem() - Enter
02-24 01:55:22.094 I/use-Rlog/RLOG-RILR(  512): CSystemManager::InitializeSystem() - Retrieved
Last CLIP Value: 0x0
02-24 01:55:22.099 I/use-Rlog/RLOG-RILR(  512): CSystemManager::InitializeSystem() - Retrieved
Last CLIR Value: 0x0
02-24 01:55:22.101 I/use-Rlog/RLOG-RILR(  512): CSystemManager::InitializeSystem() - Retrieved
Last COLP Value: 0x0
02-24 01:55:22.132 I/use-Rlog/RLOG-RILR(  512): CSystemManager::InitializeSystem() - Retrieved
Last COLR Value: 0x0
02-24 01:55:22.157 I/use-Rlog/RLOG-RILR(  512): CSystemManager::InitializeSystem() - Default DTE
Delay: 20
02-24 01:55:22.169 I/use-Rlog/RLOG-RILR(  512): CSystemManager::InitializeSystem() - Roaming
Network Improvements: [0]
02-24 01:55:22.228 I/use-Rlog/RLOG-RILR(  512): CChannelBase::OpenPort() - Opening COM
Port=[/dev/ttyACM2]  g_bIsSocket=[0]
02-24 01:55:28.072 I/use-Rlog/RLOG-RILR(  512): RIL_Init,init not finish:0
02-24 01:55:28.239 I/use-Rlog/RLOG-RILR(  512): CPort::OpenPort()  ATTEMPT NUMBER 1
02-24 01:55:28.240 I/use-Rlog/RLOG-RILR(  512): CFile::Open() - Enter
02-24 01:55:28.240 I/use-Rlog/RLOG-RILR(  512): CFile::Open() : pszFileName=[/dev/ttyACM2]
02-24 01:55:28.240 I/use-Rlog/RLOG-RILR(  512): CFile::Open() : fIsSocket=[0]
02-24 01:55:28.240 I/use-Rlog/RLOG-RILR(  512): CFile::Open() : fExists=[1]  fFile=[0]
02-24 01:55:28.240 E/use-Rlog/RLOG-RILR(  512): **********CFile::Open() : m_file=[7]
02-24 01:55:28.240 I/use-Rlog/RLOG-RILR(  512): CFile::Open() - Exit  m_fInitialized=[1]
02-24 01:55:28.240 I/use-Rlog/RLOG-RILR(  512): CChannelBase::OpenPort() - Opening COM Port:
SUCCESS
02-24 01:55:28.241 I/use-Rlog/RLOG-RILR(  512): CSystemManager::InitializeSystem() - VSPs were
opened successfully.
```

It is possible to provide the general Android log messages using the following command:

```
adb logcat  -v time | tee system.logs
```

The above command generates a log output like the following example:

```
12-21 11:42:22.453 I/DEBUG   (   31): debuggerd: Dec  2 2011 16:20:35
12-21 11:42:22.503 D/qemud   (   39): entering main loop
12-21 11:42:22.513 I/Netd    (   30): Netd 1.0 starting
12-21 11:42:22.564 I/Vold    (   29): Vold 2.1 (the revenge) firing up
12-21 11:42:22.564 D/Vold    (   29): USB mass storage support is not enabled in the kernel
12-21 11:42:22.564 D/Vold    (   29): usb_configuration switch is not enabled in the kernel
12-21 11:42:22.564 D/Vold    (   29): Volume sdcard state changing -1 (Initializing) -> 0 (No-
Media)
12-21 11:42:22.854 D/qemud   (   39): fdhandler_accept_event: accepting on fd 9
12-21 11:42:22.854 D/qemud   (   39): created client 0xe078 listening on fd 10
12-21 11:42:22.854 D/qemud   (   39): client_fd_receive: attempting registration for service
'boot-properties'
```

## 4.3   Logging commands

| Logs description | Commands |
|---|---|
| Radio logs | `$ adb logcat -b radio -v time | tee radio.log` |
| System logs | `$ adb logcat -v time | tee system.log` |
| System and radio logs | `$ adb logcat -b radio -b system -v time > sys_radio.log` |
| Kernel logs | `$ adb shell dmesg > dmesg.log` |
| Kernel and radio logs | `$ adb shell '(logcat -b radio -v time -f /dev/kmsg | cat /proc/kmsg)' > kernel_radio.log` |
| Clear previous logs | `$ adb logcat -b all -c` |
| Disable chatty service | `$ adb logcat -P " "` <br> //Execute this command at boot-up to avoid missing radio logs due to chatty service |
| Capture all logs | `$ adb logcat -b all -v time` |

## 4.4   Logger configuration

The logging level can be set in the **Group Logging** section of the `repository.txt` file by changing the value of the **LogLevel** field.

File path: `Common/build/target/product/rootdir/etc/uril/repository.txt`

**Allowed values**: Table 1 lists the various values that are currently supported.

| [LogLevel] | Logging levels |
|---|---|
| 1 | Verbose |
| 2 | Info |
| 3 | Warning |
| 4 | Critical |

**Table 1: Supported logging levels**

**Example**: The following lines are present in the `Repository.txt` file to define the log level.

```
// Log levels
Group Logging
LogLevel  2    // 4: Critical, 3: Warning, 2: Info, 1: Verbose
```

## 4.5   PPPD debug log

To enable the PPPD log, **debug** should be uncommented in the corresponding files.

File path: `Common/build/target/product/rootdir/etc/ppp/gprsX`

(where X is the PPPD interface used by RIL)

```
# uncomment the following line to debug pppd
debug
ttyACM1
crtscts
nodetach
usepeerdns
defaultroute
ipcp-max-failure 30
connect "chat -t 15 -f /system/etc/ppp/chat-isp1"
```

## 4.6 Verifying device ports permissions

To verify the permissions of the CDC-ACM or serial device ports, go to the adb shell and enter the commands:

```
$ls –l /dev/ttyACM*
$ls –l /dev/ttyUSB*
```

The permissions of the device should be set to radio for both the group and the user.

## 4.7 Start and stop RIL and MUX daemons

Use the following commands in the adb shell to start and stop the RIL and the MUX daemons.

If using Android 8.1 and onward, which uses vendor partition for RIL and MUX services, use these commands:

```
$start vendor.gsmmuxd
$start vendor.ril-daemon
$stop vendor.gsmmuxd
$stop vendor.ril-daemon
```

If using Android 5.x-8.0, which uses system partition for RIL and MUX services, use these commands:

```
$start gsmmuxd
$start ril-daemon
$stop gsmmuxd
$stop ril-daemon
```

## 4.8 Debugging the CDC-ACM or serial devices

After stopping RIL and MUX daemons, open the adb shell in two separate terminals. Use the following command in the first terminal:

```
$cat /dev/ttyUSB0       (to read the AT command response from the ttyUSB0 device)
```

Then enter the following command in the second terminal to send AT commands to the required device. For example:

```
$echo -en 'AT\r\n' > /dev/ttyUSB0     (where 'AT' is the AT command being sent and
                                        ttyUSB0 is the device being used)
```

The output of the AT command is displayed on the first terminal. For the CDC-ACM terminals just replace /dev/ttyUSB0 in the above commands with the CDC-ACM port to debug, e.g. /dev/ttyACM0.

# Appendix

# A Kernel configuration

Configure the kernel to support the u-blox cellular module and the applicable interface connection.

## A.1 USB kernel configuration

Enable the kernel configuration as follows:

```
Device Drivers
   USB Support
      <*>USB Modem (CDC ACM) support
```

## A.2 UART kernel configuration

This device is enabled in the default kernel configuration.

## A.3 SPI kernel configuration

Enable the kernel configuration as follows:

```
Device Drivers
   SPI Support
      <*>Debug support for SPI driver
      <*>GPIO-based bitbanging SPI host
      <*>McSPI driver for OMAP24xx/OMAP34xx
```

⚠ Implement the dedicated SPI protocol (see the SPI interface application note [1]) on the SPI kernel driver to correctly work with the cellular modules supporting the SPI interface.

## A.4 RNDIS / ECM kernel configuration

Enable the kernel configuration as follows:

```
Device Drivers
   Network device support
USB Network Adapters
   <*>Multi-purpose USB Networking Framework
   <*>CDC Ethernet support
   <*>Host for RNDIS and ActiveSync devices
```

⚠ This configuration is required only for the TOBY-L2 series modules configured with 1 CDC-ACM port and 1 RNDIS (factory-programmed configuration) or 1 ECM (AT+UUSBCONF=2).

## A.5 PPP kernel configuration

Enable the kernel configuration as follows:

```
Device Drivers
   Network device support
      <*>PPP (point-to-point protocol) support
      <*>PPP support for async serial ports
      <*>PPP support for sync tty ports
      <*>PPP Deflate compression
```

☞ Check if the "chat" package is included in the build. If not, add it using the following line in one of the .mk files included in the build.
```
PRODUCT_PACKAGES += chat
```

# B  Android 5.x/6.x/7.x/8.x/9.x/10.x/11.x module configuration

## B.1  `core.mk` configuration

The following modifications are required to include `core_ublox.mk` in android source.

```
File Path: build/target/product/core.mk or
           build/target/product/core_64_bit.mk
Modifications: Add below mentioned line at end of core.mk
   #***u-blox Modifications***#
   $(call inherit-product, $(SRC_TARGET_DIR)/product/core_ublox.mk)
   #***u-blox Modifications***#
```

## B.2  `ueventd.rc` configuration

This file is used to create or remove the device node (`/dev/xxx`) by receiving `uevent` messages from the kernel. To create the node for the u-blox modules, add the following configuration in `ueventd.[platform].rc` or `ueventd.rc`. Where `[platform]` labels depend on the Board Support Package (BSP), follow these permissions:

```
File Path [AOSP]: [ANDROID_SOURCE]/device/<platform>/ueventd.[platform].rc
```

Modifications:

**For USB/ttyACM mode:**
```
/dev/ttyACM[0-5]* 0660 radio  radio  #[if using CDC-ACM Interface]
```

**For UART/MUX mode:**
```
/dev/<serial device> 0660 radio radio #[if using VCP Interface e.g. ttyUSB, ttyS or ttymxc]
/dev/pts*   0660 radio  radio  #[if using pts Interface with MUX]
```

**For PPP mode:**
```
/dev/ppp 0660 radio radio
```

If PPP is already added with different permissions other than radio, then modify it to "radio radio".

## B.3  device.mk

```
File Path [AOSP]:      [ANDROID_SOURCE]/device/<platform>/device.mk
File Path [i-MX8/10]: [ANDROID_SOURCE]/device/fsl/imx8m/evk_8mq/evk_8mq.mk
```

Modifications:

- Set port path and ril lib path using following lines.
  o  **For Android 8.1/9.x/10.x**
  ```
  vendor.rild.libpath=/vendor/lib64/librapid-ril-core.so
  ```
  o  **For Android 5.x/6.x/7.x/8.0**
  ```
  rild.libpath=/system/lib64/librapid-ril-core.so   <For 64-bit architecture>
  rild.libpath=/system/lib/librapid-ril-core.so     <For 32-bit architecture>
  ```
☞  Android 9.x and onwards binaries path is changed from /system to /vendor.

☞  For TOBY-L4 series only use interface name ttyACM2.

- Update following property to set default network type to global (LTE, GSM, WCDMA).
  ```
  ro.telephony.default_network=9
  ```

## B.4 `init.rc` configuration

The following modifications are required to include `init.ublox.rc` in Android source.

```
File Path [AOSP]:  [ANDROID_SOURCE]/system/core/rootdir/init.rc
File Path [i-MX6]: [ANDROID_SOURCE]/device/fsl/sabresd_6dq/init.rc
```

Modifications: Add below mentioned line at top of `init.rc`.

```
#***u-blox Modifications***#
import /init.ublox.rc
#***u-blox Modifications***#
```

## B.5 `init.ublox.rc` configuration

The `init.ublox.rc` file contains all the necessary information used for the RIL integration. All the services, properties, permissions used for u-blox RIL are mentioned in this file.

This file is self-explanatory and contains all the steps used for the RIL initialization. Follow these steps to configure this file as per requirements.

☞ In the RIL delivery package the **init.ublox.rc** file is available at the following path:
`ril_sc_<version>/Common/system/core/rootdir`

☞ After the integration follow the steps mentioned in `init.ublox.rc`.

## B.6 SEPolicy configuration

To get started with SELinux:

1.  Enable SELinux in the kernel: CONFIG_SECURITY_SELINUX=y

    (By default it is enabled in the Android Open Source Project AOSP)

2.  Change the kernel_cmdline parameter (`device/<vendor>/<target>/BoardConfig.mk`) in the following way:
    `BOARD_KERNEL_CMDLINE := androidboot.selinux=permissive`

Permissive mode is only set for initial development phase. Once RIL integration is completed, it is recommended to set the device again to enforcing mode.

☞ For further details about the SEPolicy configuration, please contact the u-blox technical support.

# C AT pass through commands

The Android operating system does not allow sending commands to the module without using the RIL software layer, so two requests provided by RIL's API must be used:

- `RIL_REQUEST_OEM_HOOK_RAW`: it passes raw byte arrays back and forth
- `RIL_REQUEST_OEM_HOOK_STRINGS`: it passes strings back and forth

The following is sample code that could be used to send commands directly to the module:

⚠ Make a backup of the `<android_root>/packages/apps/Settings` folder before making any changes to it.

- Add the following lines in the bottom of `packages/apps/Settings/AndroidManifest.xml` before the `</application>` field.

```
<activity android:name="RilOemHookTest" android:label="@string/testing_RIL_OEMHook"
    android:process="com.android.phone">
    <intent-filter>
      <action android:name="android.intent.action.MAIN" />
      <category android:name="android.intent.category.DEVELOPMENT_PREFERENCE" />
    </intent-filter>
</activity>
```

- Add the following lines in the bottom of `packages/apps/Settings/res/values/strings.xml` file before the `</resources>` field.

```
<!-- For RIL OEM HOOK testing -->
<string name="testing_RIL_OEMHook">RIL OEM Hook Test</string>
<string name="ril_oem_choose">Choose a RIL OEM Hook API to test:</string>
<string name="foat_fls">Start FOAT (fls file=C800)</string>
<string name="foat_dffs">Start FOAT (dffs file=C801)</string>
<string name="radio_usb">Set Prop (modInf=USB)</string>
<string name="radio_uart">Set Prop (modInf=UART)</string>
<string name="radio_api1">API 1 (datalen=0)</string>
<string name="radio_api2">API 2 (datalen=1)</string>
<string name="radio_api3">API 3 (datalen=6)</string>
<string name="radio_api4">API 4 Type command :</string>
<string name="radio_run">RUN</string>
<string name="ril_oem_response">Module response:</string>
```

At the bottom of the `packages/apps/Settings/res/xml/testing_settings.xml` file before the `</PreferenceScreen>` field, add the following lines:

```
<PreferenceScreen
        android:title="@string/testing_RIL_OEMHook" >
    <intent
        android:action="android.intent.action.MAIN"
        android:targetPackage="com.android.settings"
        android:targetClass="com.android.settings.RilOemHookTest" />
</PreferenceScreen>
```

- Run the following command to merge the OEM Hook test application code into the Android source.

```
cp –pvRf ril_sc_<version>/Android_<version>/packages/apps/Settings/*
<android_root>/packages/apps/Settings/
```

- Add the following Java code to the relevant file depending on the Android version being used:

| Android version | File name and path | Java code |
|---|---|---|
| For Android 5.x - 6.x | `<android_root>/frameworks/opt/ telephony/src/java/com/android/ internal/telephony/PhoneProxy.java` | `public CommandsInterface getCommandsInterface() { return mCommandsInterface; }` |
| For Android 7.x - 11.x | `<android_root>/frameworks/opt/ telephony/src/java/com/android/ internal/telephony/Phone.java` | `public CommandsInterface getCommandsInterface() { return mCi; }` |

- It is possible to start this application from an Android shell using the following command or it can also be accessible using the Android secret code *#*#4636#*#*:

```
am start -a android.intent.action.MAIN -n
com.android.settings/com.android.settings.RilOemHookTest
```

or

```
am start -a android.intent.action.MAIN -n
com.android.settings/com.android.settings.TestingSettings
```

Now click on the "RIL OEM Hook Test" menu. A screen such as Figure 1 is displayed.
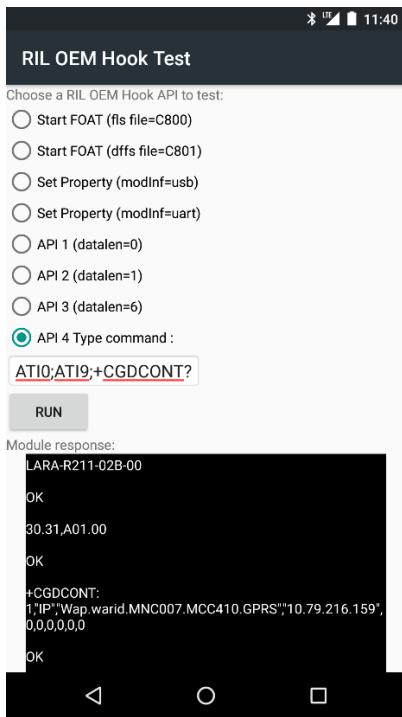


**Figure 1: RIL OEM Hook application**

The "Start FOAT" options are to be used for the module firmware update if the provided RIL and module are supported. See appendix F for more information on this. Apart from that, the first three APIs send bytes to the RIL layer to execute a set of commands.

The fourth API sends the command string to the module. Type the command into the white field and click the "RUN" button to execute the command. The module response is displayed in the bottom field.

⚠ RIL uses the numeric error value (AT+CMEE=1) as the error result codes format. A change in the error report setting causes unexpected errors in the RIL behavior.

# D   Default EPS bearer in LTE (initial PDP context)

## D.1   Default EPS bearer in LTE

An EPS bearer is established when the UE connects to a PDN and remains established throughout the lifetime of the PDN connection to provide the UE with always-on IP connectivity to that PDN. This bearer refers to the default EPS bearer. The UE can have additional default bearers as well. Each default bearer comes with an IP address and it has nominal QoS applied by the LTE network.

A default EPS bearer is allocated for an APN. To create multiple "Default EPS bearers" in an LTE network, more than one APN should be working in the LTE network.

## D.2   `apns-conf.xml` configurations (manually/during build process)

The `apns-conf.xml` file should have at least one corresponding MCC/MNC entry/record with **type** field "ia", where "ia" stands for "Initial Attach". Multiple values can be defined in the **type** field for example, default,supl,mms,ia

Example: during the source build process, the user can enter the corresponding network settings in the `apns-conf.xml` file. The sample values are given below.

```
<apn carrier="Cosmote Wireless Internet"
    mcc="202"
    mnc="01"
    apn=""
    type="ia"
/>
<apn carrier="Cosmote Wireless Internet"
    mcc="202"
    mnc="01"
    apn="internet"
    type="default,supl,mms"
/>
```

## D.3   `apns-conf.xml` configurations (Android UI)

The user can also set the APN according to the service provider manually through the Android UI.

- Go to Android's main settings.
- Under Wireless & Networks, go to **mobile networks** (Figure 2). Click **more settings** to get this option. (Figure 3).
- Tap on Access Point Names (that is what APN stands for).
- Press menu, and then click **New APN**.
- Enter in all of the required information in each field by clicking into it and then clicking **OK** to save (see Figure 4).
- After having finished entering all the settings, click menu again and then Save.
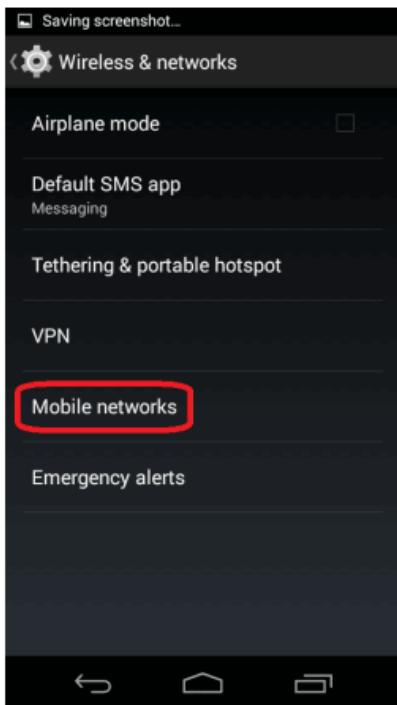
**Figure 2**
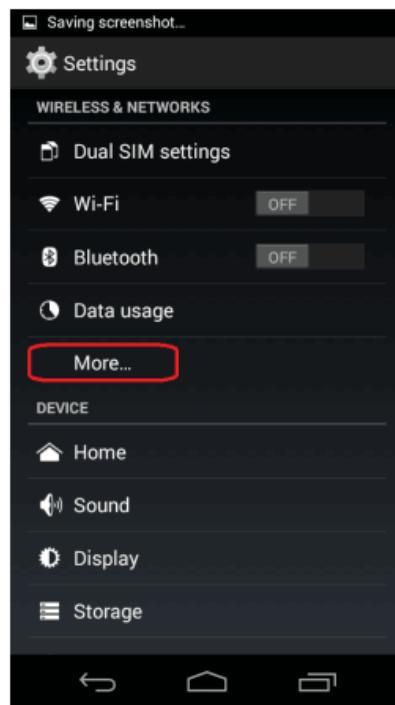


**Figure 3**
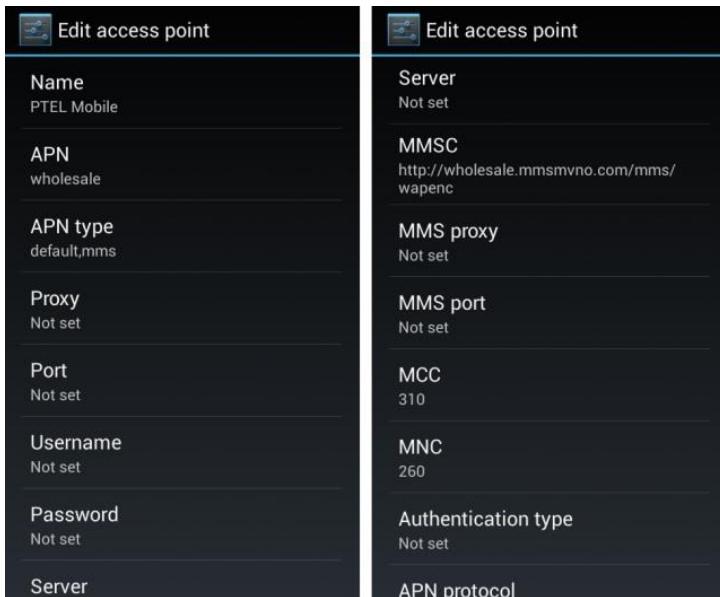


**Figure 4**

# D.4   Single default EPS bearer allowed – RIL handling

Some LTE networks allow only one default EPS bearer establishment: a single working APN is available for the user.

In the scenario where only a single working APN is available, the RIL tries to establish data on CID=1 (CID=1 is requested by OS) when `RIL_REQUEST_SETUP_DATA_CALL` is triggered by the OS. If a single default bearer case is encountered, RIL failed in its maximum tries while establishing data on CID=1.

Once the RIL has failed to establish the data on the requested CID by the OS, the RIL tries to establish data on CID=4, which is already established during the time of the LTE attach process, on the next try of `RIL_REQUEST_SETUP_DATA_CALL` by the OS.

# E Repository file configuration

The `repository.txt` file is a part of the u-blox RIL source, and it is used to set various user defined values for various fields. These fields are split into sections. Operational timeouts, TTY constants, RIL settings, MUX settings, last values, USB/UBM and log level etc. are some examples of the fields/values defined in this file.

## E.1 RIL TTY mode and port setting

The RIL TTY operating mode and port name setting can be configured by the "**TTY**" group of the `repository.txt` file.

**Example:** the following lines are present in the `Repository.txt` file to define the default TTY port setting.

```
// RIL TTY port settings
Group TTY
    Mode                    ttyACM          // Supported values: ttyACM, ttyUSB
                                            // (USB mode), pts (MUX mode),
                                            // multimode (Multiple module
                                            // support), remote (u-blox internal)
    CommandPort             /dev/ttyACM0
    CmdPortBaudRate         115200          // Supported values: 115200
    DataPortBaudRate        921600          // Supported values: 115200, 230400,
                                            // 460800, 921600 (default), 3000000
```

Configure the default interface of u-blox RIL by setting "Mode" field to one of the below values:

- **ttyACM (default)**
  For CDC-ACM or RNDIS mode. RIL will use modem usb ttyACM ports for AT and data channels. RIL will also set property `persist.ublox.uril.modInf` to `ttyACM`.
- **ttyUSB** (Only for SARA-R5 with dual UART configuration)
  For serial mode without MUX. RIL will use modem serial usb ttyUSB ports for AT and data channels. RIL will also set property `persist.ublox.uril.modInf` to `ttyUSB`.
- **pts**
  For serial mode with GSMMUX. RIL will use virtual pts ports created by MUX (using ttyS/ttyUSB serial interface) for AT and data channels. RIL will also set property `persist.ublox.uril.modInf` to `pts`.
- **multimode**
  For CDC-ACM/GSMMUX multi-mode. RIL will read property `persist.ublox.uril.modInf` and set mode automatically. For more details, see section G.
- **remote**
  For remote mode. Only for u-blox internal use. RIL will also set the `persist.ublox.uril.modInf` property to `remote`.

RIL will use the AT command port mentioned in the `CommandPort` field for communication with a modem if `Mode` is set to `ttyACM`, `ttyUSB` or `multimode`. The port could be a ttyACM or ttyS/ttyUSB serial port (in case of SARA-R5 dual UART mode).

☞ If `Mode` is set to `pts` then `CommandPort` is not applicable. In that case, the port specified by the `SerialPort` field in the `MUXPortSettings` group will be used by MUX for communication with a modem.

☞ The `CmdPortBaudRate` is the AT command port speed and currently fixed to 115200. This field will be applicable only if `CommandPort` is a ttyUSB/ttyS serial port. This parameter has no effect if `CommandPort` is a ttyACM usb port.

☞ The `DataPortBaudRate` is the data port speed for PPPD. This field will be applicable only for SARA-R5 / SARA-R42 series when [+USIO mode] is set to dual UART and `Mode` is set to `ttyUSB`. See the respective module's AT manual for officially supported bauds list. For RIL, currently supported baud rates are 115200, 230400, 460800, 921600 (default), 3000000 bit/s. For SARA-R422 / SARA-R422S / SARA-R422M8S, this value will also update the +UUARTCONF AT command setting if it is not already synchronized.

## E.2  GSMMUX setting

The GSMMUX port setting can be configured by the **MUXPortSettings** group of the `repository.txt` file.

**Example:** the following lines are present in the `repository.txt` file to define the default MUX port setting.

```
// Serial port settings when using the GSM MUX
Group MUXPortSettings
    SerialPort              /dev/ttyUSB0
    MUXFlowControl          1               // 1:Enabled, 0: Disabled
    MUXBaudRate             115200
    MUXNumOfVirtualPorts    3               // Max 5 supported
    MUXFrameSize            1509            // Any value from 7 - 1509
    MUXExitPowerOff         0               // 0:Disabled,1:Enabled Send CPWROFF
                                            // during MUX exit
    MUXVerbosity            5               // 0 (Silent) - 7 (Debug)
```

☞ This "`MUXPortSettings`" setting will be applicable only if the "Mode" field in the "TTY" group is set to "pts" or "multimode".

## E.3  Band selection bitmask +UBANDMASK

☞ The +UBANDMASK command is supported by the SARA-R4 / SARA-R5 series.

Sets the supported LTE-M / NB-IoT bands for different Radio Access Technologies (RATs).

The bands supported are set by means of a bitmask where each bit in an 8-byte integer corresponds to an LTE-M / NB-IoT band.

```
SetBandMaskFilter    0

Allowed values:
o   DISABLED (default) = 0
o   Max allowed value  = 0xFFFFFFFFFFFFFFFF
o   Test mode setting  = 18446744073709551615
```

## E.4  Serial interfaces configuration selection +USIO

☞ The +USIO AT command is supported by the SARA-R42 / SARA-R5 series.

Sets the serial interfaces configuration to single UART or dual UART mode.

The following configuration settings are available in the `repository.txt` file.

```
USIOConf    0
```

Allowed values:

- For SARA-R5 series:
```
0: Single UART(7-wire)      ==> Suggested RIL Mode = pts
1: Single UART(9-wire)      ==> Suggested RIL Mode = pts
2: Double UART(5-wire)      ==> Suggested RIL Mode = ttyUSB
3: Single UART(5-wire)      ==> Suggested RIL Mode = pts
```

- For SARA-R42 series:
  ```
  0: Single UART(9-wire)         ==> Suggested RIL Mode = pts
  1: Double UART(5-wire)         ==> Suggested RIL Mode = ttyUSB
  2: Single UART(5-wire)         ==> Suggested RIL Mode = pts
  ```

The RIL uses the +USIO AT command to set various UART configuration settings according to the value set in this field. For more details, see [Module specific configurations] for SARA-R5 / SARA-R4 series.

The recommended setting for SARA-R5 series +USIO is 2, which is double UART.

The recommended setting for SARA-R42 series +USIO is 1, which is double UART.

Android RIL supports both dual UART and single UART with MUX interface. For more details on the UART configuration of SARA-R5 and SARA-R42 series modules using +USIO AT command, see the corresponding system integration manual [7], [8].

## E.5   MNO profile setting

Automatically configures the module to be compliant to the requirements of various mobile network operators (MNO). Use the following settings to set the desired MNO profile. The 999 value is set as default value to disable this setting and the RIL will not set any MNO profile.

```
SetMobileNetworkOperator    999

DISABLE (default) = 999,
SW-default = 0,
ATT = 2,
Verizon = 3,
Telstra = 4,
T-Mobile US = 5,
China Telecom = 6,
Vodafone = 19,
Europe = 100,
```

☞   The +UMNOPROF AT command is supported by the SARA-R4 / SARA-R5 series modules.

☞   The +UMNOCONF AT command is supported by TOBY-L2 / LARA-R2 series modules.

## E.6   CDMA network operator name

The network operator names for CDMA are set in the "**Operator Settings**" group of the `repository.txt` file by setting the "OperatorNames" field to its appropriate name and the MCC and MNC values.

**Example:** the following lines are present in the `Repository.txt` file to define the network operator name.

```
//Operator Settings (only for CDMA)
Group OperatorNames
Supported Operator    Sprint
MCCMNC                310120
```

## E.7   Timeouts

The timeout values in the **Request timeouts** and **other timeouts** groups present in the repository file have already been tuned by u-blox for optimum functionality, but they can be modified according to need.

## E.8 Emergency numbers

From the developer's point of view, emergency numbers can be set in Android by assigning either the `ro.ril.ecclist` or `ril.ecclist` property, where the first one can be set at build-time only and is read-only, while the latter is dynamic and can be set/modified at runtime. When dialing emergency numbers, by default Android first checks the `ril.ecclist` property for the specific number, but if this property is not defined, then it checks the number in the `ro.ril.eccclist` property. The numbers currently present in the list are shown in Table 2. For more information on these numbers, kindly consult this Wikipedia link.

| Numbers | Listing |
|---|---|
| 111,113,115,117,122,125,127 | Common emergency numbers used in various countries |
| 112,911,000,08,110,999,118,119 | ETSI standard |

**Table 2: Supported emergency numbers**

# F   Module firmware update

☞ For further details about the module firmware update in an Android OS system, please contact the u-blox technical support.

# G Multi module support in Android RIL

⚠ This section only applies to RIL version 09.01 and above.

## G.1 Multi module

This method allows the u-blox RIL to switch between two modules connected on different interfaces. Only the USB and UART interfaces can be used to set up this feature.

When the module is configured in CDC-ACM or RNDIS mode, then it will be symbolized it as a USB interface. If the module is connected using a serial port or virtual COM port on a serial interface, then it is symbolized it as a UART interface.

The main purpose of this feature is to use two modules on both interfaces but activate one at a time; RIL has a functionality to switch between the two modules on a particular trigger.

### G.1.1 Prerequisites

Both modules should be connected on the respective interfaces according to need. By default, it is in "ttyACM[usb]" mode. The default mode can be configured through the `szInterfaceDefault` variable in the function named `mainloop` in the `rildmain.cpp` file.

### G.1.2 Mode switching

u-blox also provides a sample test application source for the user to trigger between both modules by setting a specific property. See appendix C to integrate the testing application in the Android source code. This application uses OEM_HOOK_API to communicate with RIL and perform specific operations.

The switching process is initialized by selecting the required interface property in the RIL OEM HOOK Test APK, for example, Set Property (modinf=ttyACM) for the USB interface and Set Property (modinf=pts) for the UART interface.

By setting the following property, mode switching operations can be performed.

> `persist.ublox.uril.modInf`

Possible values of this property are

- For CDC-ACM interface: "`ttyACM`"
- For UART interface with GSMMUX: "`pts`"

☞ The value of this property (`persist.ublox.uril.modInf`) can also be set from the `repository.txt` file by updating "Mode" field in "TTY" group setting to persist its desired set value.

Follow these steps to use this feature manually:

1. Set the "`Mode`" field of "`TTY`" group in `repository.txt` file to "`multimode`".
2. Set the property (`persist.ublox.uril.modInf`) for the desired interface.
3. Power off the module by means of the AT command.
4. Cut off the power IO to stop the module power.
5. Now enable the power of the second module.
6. The RIL will automatically connect to the set interface if it is available.

### G.1.3 Switching status

After the property has been successfully set, the power of the currently active module should be cut off, which can be done by issuing the AT+CPWROFF command from the RIL OEM HOOK test APK. Then turn the power back on to the new active module.

# H  Verizon network

⚠  This section only applies to RIL version 09.03 and onwards.

## H.1  init.ublox.rc

The following service needs to be added to the service section of `init.ublox.rc` file as follows:

```
# APN update service for Verizon Network
service static_apn /system/bin/static_apn
user root
group radio cache inet misc
class main
disabled
oneshot
```

## H.2  core_ublox.mk

Update the following file to copy the target directory on the device:

- `ANDROID_SOURCE/build/target/product/core_ublox.mk`

Add the following line to `PRODUCT_COPY_FILES`

```
$(LOCAL_PATH)/rootdir/etc/static_apn:system/bin/static_apn
```

# I  Module specific configurations

This section applies to modules requiring some customized modifications in the Android source or kernel to properly communicate with the u-blox RIL driver.

## I.1  TOBY-L2 series

TOBY-L2 are multi-radio Access Technology (RAT) modules capable of operating in 2G, 3G and 4G mobile networks. Logical interface supported are RNDIS, CDC-ECM and CDC-ACM. Regardless of the currently selected RAT, the packet switched connectivity over the USB virtual Ethernet interface may be established in two different networking modes:

*   **Router mode (default)**: the IP termination is placed on the module itself. In this configuration the data connectivity of the customer's application processor is provided through routing procedures. The module is operating as a mobile router.
*   **Bridge mode**: the IP termination of the data connectivity is on the customer's application processor and the module acts as a bridge device (similar to a USB dongle).

### I.1.1  TOBY-L2 USB profile configuration

The USB configurations can be set in the **Group USBUBM** section of the `repository.txt` file by setting the value of the **USBCONF** and **UBMCONF** fields, respectively. The various values that are currently supported are shown in Table 3 and Table 4. If these settings are not configured, then the module's default settings will be used.

| [USBCONF] | Configurations | RIL support |
| --- | --- | --- |
| 0 | CDC-ACM | Supported |
| 2 | CDC-ECM | Not Supported |
| 3 | RNDIS | Supported |

**Table 3: Supported USB configurations**

| [UBMCONF] | Configurations | RIL support |
| --- | --- | --- |
| 1 | Router | Supported |
| 2 | Bridge | Supported |

**Table 4: Supported u-blox modem configurations**

## I.2  SARA-R5 series

SARA-R5 are LTE-M / NB-IoT modules with secure cloud.

☞  Android RIL does not support NB-IoT mode for SARA-R5 series.

SARA-R5 series modules come with the dual UART feature, UART0 can be used as AT interface and UART1 can be used as data interface. For data services, PPP mode is supported. SARA-R5 series modules can also be used with single UART using GSMMUX to create multiple points channels for AT and data control ports.

### I.2.1  RIL configurations with dual UART interfaces

*   In the `repository.txt` file:
    Locate and set following fields in "`TTY`" group.
    ```
    Mode            ttyUSB
    CommandPort     /dev/ttyUSB0
    ```

    Locate and set following fields in "`CustomRILSettings`" group.
    ```
    USIOConf        2
    ```

### I.2.2    RIL configurations with single UART+MUX interface

- In the `repository.txt` file:
Locate and set following fields in "`TTY`" group.
```
Mode            pts
CommandPort     /dev/ttyUSB0
```

Locate and set following field in "MUXPortSettings" group.
```
SerialPort      /dev/ttyUSB0
```

Locate and set following fields in "`CustomRILSettings`" group.
```
USIOConf        0
```

- Enable the hardware flow control from command line parameters of GSMMUX service as mentioned in multiplexer implementation application note.

☞ For information on MUX interfacing, see the Multiplexer implementation application note [5].

## I.3    SARA-R4 series

SARA-R4 are LTE-M/NB-IoT/EGPRS modules.

☞ Android RIL does not support NB-IoT mode for SARA-R4 series modules.

SARA-R4 series modules can be used with a single UART using GSMMUX to create multiple channel points for AT and data control ports.

However, SARA-R42 series modules also come with the dual UART feature. UART0 can be used as an AT interface and UART1 can be used as a data interface. For data services, PPP mode is supported.

### I.3.1    SARA-R42 RIL configurations with dual UART interfaces

- In the `repository.txt` file:
Locate and set following fields in "`TTY`" group.
```
Mode            ttyUSB
CommandPort     /dev/ttyUSB0
```

For SARA-R42 series only, locate and set the following field in "`CustomRILSettings`" group.
```
USIOConf        1
```

### I.3.2    SARA-R4 RIL configurations with single UART+MUX interface

- In the `repository.txt` file:
Locate and set the following fields in "`TTY`" group.
```
Mode            pts
CommandPort     /dev/ttyUSB0
```

Locate and set the following field in "`MUXPortSettings`" group.
```
SerialPort      /dev/ttyUSB0
```

For SARA-R42 series only, locate and set the following field in "`CustomRILSettings`" group.
```
USIOConf        0
```

- Enable the hardware flow control via the command line parameters of GSMMUX service, as mentioned in the multiplexer implementation application note.

☞ For information on MUX interfacing, see the Multiplexer implementation application note [5].

# J   Audio codec

⚠️   LISA-C2 product series does not support external codec management by AT commands.

## J.1   Configuration

This section describes a set of u-blox proprietary AT commands to be used for the audio features configuration. Customers can configure this section for custom audio features according to their platform needs.

The section below in the RIL code is divided into two main sections:

- Settings to manage external codec or other external audio IC
- Setting the audio path and configure I2S interfaces

For further details, see the u-blox AT commands manual [6].

## J.2   Example

Below, a RIL code snapshot is provided to mark the location where the customer can update/add/manage their own custom audio codec.

```
(ublox_ril\core\nd\Systemmanager.cpp under ::InitializeModemFeatures() )

//Initialized digital audio features (I2C interface)
pCmd = new CCommand(RIL_CHANNEL_ATCMD, NULL,ND_REQ_ID_NONE,
"AT+UMCLK=2,1;+UI2CO=1,0,0,0x10,0;+UI2CW=\"00000000108F2024000010330025000008A\",18;+U
I2CW=\"049E\",2;+UI2CC\r", &CTE::ParseSupportedFeatures);

//Initialized digital audio features (I2S interface)
pCmd = new CCommand(RIL_CHANNEL_ATCMD, NULL,ND_REQ_ID_NONE,
"AT+USPM=1,1,0,0,2;+UI2S=1,1,0,3,0\r", &CTE::ParseSupportedFeatures);
```

# K  i.MX platform specific notes

Consider the following points during the integration on the i.MX 6 platform.

See section B for details about changes required in the following file.

## K.1  imx6.mk

Open the `imx6.mk` file for the path `device/fsl/imx6/imx6.mk` and apply the following changes.

```
#Remove following lines from imx6.mk file
   - chat
   - ip-up-vpn
   - ip-up-ppp0
   - ip-down-ppp0
   - device/fsl/imx6/etc/ppp/init.gprs-pppd:system/etc/ppp/init.gprs-pppd \
```

## K.2  apn-config.xml

File containing the APN information was not present in the i.MX6 platform. The following line was added to copy `apns-conf.xml` at the required path.

File path: `[ANDROID_SOURCE]/build/target/product/core_ublox.mk`

```
PRODUCT_COPY_FILES += \
    device/sample/etc/apns-full-conf.xml:system/etc/apns-conf.xml
```

## K.3  manifest.xml

Following configurations were added in manifest in order to bind RIL service

File path: `[ANDROID_SOURCE]/device/fsl/ather450/manifest.xml`

```
Lines added:
   <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>1.1</version>
   <interface>
         <name>IRadio</name>
         <instance>slot1</instance>
   </interface>
   <interface>
         <name>ISap</name>
         <instance>slot1</instance>
   </interface>
   </hal>
```

# L   Licenses and files

## L.1   Licenses

The following license can be found in the source code of the u-blox Android RIL driver package:

| License number | License |
|---|---|
| 1 | Copyright (c) 2020 u-blox AG, Thalwil, Switzerland<br><br>u-blox grants Customer a worldwide, non-exclusive, non-transferable, perpetual, without the right to sublicense, free of charge license to compile the RIL Software's source code ("Source Code") into object code format in Customer's platform containing the Product for the development, production (including production by Customer's subcontractors), sale, distribution, support and demonstration of Customer products ("RIL Software License"). Customer, except where explicitly stated otherwise, furthermore acknowledges and agrees:  not to use, copy, modify, port, translate, or otherwise reproduce or create derivative works of the RIL Software without the prior written permission of u-blox; (ii) not to decompile, disassemble, reverse engineer or attempt to reconstruct, identify, discover or disclose any Source Code, underlying user interface techniques or algorithms of the RIL Software; (iii) to reproduce the proprietary notices contained in the RIL Software and to include the applicable copyright notice each time the RIL Software is distributed; (iv) that it must have the appropriate licenses from operating system providers to develop and distribute products with the operating system; (v) to refrain from all illegal conduct and to fulfill all legal requirements placed upon Customer to perform its obligations hereunder; (vi) to not knowingly distribute the RIL Software to any person who infringes the copyright holder's rights to the RIL Software; (vii) to not distribute or provide access to the Source Code in the open source community or to provide it to general public libraries and to not knowingly take any action that would cause the Source Code to be placed in the public domain; (viii) to ensure the execution of reasonable measures to protect the confidentiality and security of the Source Code (including but not limited to: access only to a controlled group of individuals, no permission to work on development or modification of any competitive software, keep hard disk copies on password protected computers only, no transfer of soft copies across any Internet link except when password protected, keep all hard copies secured, immediately inform u-blox of any unauthorized use or disclosure, cooperate with u-blox to remedy such use or disclosure). In the event of a material violation of these provisions, which has not been cured within thirty (30) days of receipt of written notice from u-blox, u-blox shall be entitled to immediately terminate this license agreement. In such event, Customer shall immediately provide all Source Code to u-blox and certify compliance with these obligations. The RIL Software is delivered on an "as is" basis without warranty, representation or condition of any kind, including without limitation, the implied warranties or conditions of merchantable quality and fitness for a particular purpose. u-blox represents to Customer that, to the best of its knowledge, the RIL Software does not infringe any intellectual property rights of any third party. In no event shall u-blox or its licensors (including their respective licensors or suppliers) be liable for any loss of data, reprocurement costs, loss of revenues, loss of profits, loss of use or for any other consequential, indirect, exemplary, special or incidental damages arising under or in connection with this license, even if the other party has been advised of the possibility of such damages. Customer acknowledges that u-blox disclaims all liability of any nature to Customer's own customers, whether direct, indirect, incidental, consequential, arising out of the Customer's own customers' use of third party technology utilized with the RIL Software, and Customer agrees that Customer and its own customers shall have no claims against u-blox or its licensors (incl. their respective licensors or suppliers) whatsoever with respect thereto. This license agreement shall continue until terminated by u-blox in accordance with the aforementioned provisions or by Customer upon thirty (30) calendar days' prior written notice to u-blox. Each party shall return confidential information to the other within thirty (30) days from termination. Customer may dispose of RIL Software in Customer's inventory within a period of ninety (90) days. Termination of the General Terms and Conditions terminates all of Customer's license rights under this RIL Software License. Termination shall not affect Customer's right to develop, produce, sell, distribute, support and demonstrate its products containing the platform with the u-blox Product and the RIL Software in compiled object code form on the termination date. Customer acknowledges and agrees that the RIL Software and its documentation are confidential information of u-blox or its licensors (incl. their respective licensors or suppliers). |

| License number | License |
|---|---|
| 2 | Copyright 2020 u-blox AG, Thalwil, Switzerland<br><br>Licensed under the Apache License, Version 2.0 (the "License");<br>you may not use this file except in compliance with the License.<br>You may obtain a copy of the License at<br><br>http://www.apache.org/licenses/LICENSE-2.0<br><br>Unless required by applicable law or agreed to in writing, software<br>distributed under the License is distributed on an "AS IS" BASIS,<br>WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or    implied.<br>See the License for the specific language governing permissions and<br>limitations under the License. |
| 3 | Copyright 2005-2006, The Android Open Source Project<br>Author: Arve Hjønnevåg<br><br>This file is dual licensed.  It may be redistributed and/or modified<br>under the terms of the Apache 2.0 License OR version 2 of the GNU<br>General Public License. |
| 4 | Copyright 2006, The Android Open Source Project<br><br>Licensed under the Apache License, Version 2.0 (the "License");<br>you may not use this file except in compliance with the License.<br>You may obtain a copy of the License at<br><br>   http://www.apache.org/licenses/LICENSE-2.0<br><br>Unless required by applicable law or agreed to in writing, software<br>distributed under the License is distributed on an "AS IS" BASIS,<br>WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.<br>See the License for the specific language governing permissions and<br>limitations under the License. |
| 5 | Copyright (C) 2006 The Android Open Source Project<br><br>Licensed under the Apache License, Version 2.0 (the "License");<br>you may not use this file except in compliance with the License.<br>You may obtain a copy of the License at<br><br>   http://www.apache.org/licenses/LICENSE-2.0<br><br>Unless required by applicable law or agreed to in writing, software<br>distributed under the License is distributed on an "AS IS" BASIS,<br>WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.<br>See the License for the specific language governing permissions and<br>limitations under the License. |
| 6 | Copyright (C) 2007-2014 The Android Open Source Project<br><br> Licensed under the Apache License, Version 2.0 (the "License");<br> you may not use this file except in compliance with the License.<br> You may obtain a copy of the License at<br><br>   http://www.apache.org/licenses/LICENSE-2.0<br><br> Unless required by applicable law or agreed to in writing, software<br> distributed under the License is distributed on an "AS IS" BASIS,<br> WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.<br> See the License for the specific language governing permissions and<br> limitations under the License. |
| 7 | Copyright (C) 2007-2016 The Android Open Source Project<br><br>Licensed under the Apache License, Version 2.0 (the "License");<br>you may not use this file except in compliance with the License.<br>You may obtain a copy of the License at<br><br>   http://www.apache.org/licenses/LICENSE-2.0<br><br>Unless required by applicable law or agreed to in writing, software<br>distributed under the License is distributed on an "AS IS" BASIS,<br>WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. |

| License number | License |
|---|---|
| | See the License for the specific language governing permissions and limitations under the License. |
| 8 | Copyright (C) 2008 The Android Open Source Project<br><br>Licensed under the Apache License, Version 2.0 (the "License");<br>you may not use this file except in compliance with the License.<br>You may obtain a copy of the License at<br><br>    http://www.apache.org/licenses/LICENSE-2.0<br><br>Unless required by applicable law or agreed to in writing, software<br>distributed under the License is distributed on an "AS IS" BASIS,<br>WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.<br>See the License for the specific language governing permissions and<br>limitations under the License. |
| 9 | Copyright (C) 2011 The Android Open-Source Project<br><br>Licensed under the Apache License, Version 2.0 (the "License");<br>you may not use this file except in compliance with the License.<br>You may obtain a copy of the License at<br><br>    http://www.apache.org/licenses/LICENSE-2.0<br><br>Unless required by applicable law or agreed to in writing, software<br>distributed under the License is distributed on an "AS IS" BASIS,<br>WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.<br>See the License for the specific language governing permissions and<br>limitations under the License. |
| 10 | Copyright 2013 The Android Open Source Project<br><br>Licensed under the Apache License, Version 2.0 (the "License");<br>you may not use this file except in compliance with the License.<br>You may obtain a copy of the License at<br><br>    http://www.apache.org/licenses/LICENSE-2.0<br><br>Unless required by applicable law or agreed to in writing, software<br>distributed under the License is distributed on an "AS IS" BASIS,<br>WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.<br>See the License for the specific language governing permissions and<br>limitations under the License. |
| 11 | Copyright 2015 The Android Open Source Project<br><br>Licensed under the Apache License, Version 2.0 (the "License");<br>you may not use this file except in compliance with the License.<br>You may obtain a copy of the License at<br><br>    http://www.apache.org/licenses/LICENSE-2.0<br><br>Unless required by applicable law or agreed to in writing, software<br>distributed under the License is distributed on an "AS IS" BASIS,<br>WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.<br>See the License for the specific language governing permissions and<br>limitations under the License. |
| 12 | Copyright (C) 2015 The Android Open-Source Project<br><br>Licensed under the Apache License, Version 2.0 (the "License");<br>you may not use this file except in compliance with the License.<br>You may obtain a copy of the License at<br><br>    http://www.apache.org/licenses/LICENSE-2.0<br><br>Unless required by applicable law or agreed to in writing, software<br>distributed under the License is distributed on an "AS IS" BASIS,<br>WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.<br>See the License for the specific language governing permissions and<br>limitations under the License. |

| License number | License |
|---|---|
| 13 | Copyright (C) 2016 The Android Open-Source Project

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License. |
| 14 | Copyright (C) 2017 The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License. |
| 15 | Copyright (C) 2018 The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License. |
| 16 | Copyright (c) 1994-2004 Paul Mackerras. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. The name(s) of the authors of this software must not be used to endorse or promote products derived from this software without prior written permission.

3. Redistributions of any form whatsoever must retain the following acknowledgment:
"This product includes software developed by Paul Mackerras
 <paulus@samba.org>".

THE AUTHORS OF THIS SOFTWARE DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL THE AUTHORS BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Derived from main.c and pppd.h, which are:

Copyright (c) 1984-2000 Carnegie Mellon University. All rights reserved.

Redistribution and use in source and binary forms, with or without |

| License number | License |
|---|---|
| | modification, are permitted provided that the following conditions are met:<br><br>1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.<br><br>2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.<br><br>3. The name "Carnegie Mellon University" must not be used to endorse or promote products derived from this software without prior written permission. For permission or any legal details, please contact<br>  Office of Technology Transfer<br>  Carnegie Mellon University<br>  5000 Forbes Avenue<br>  Pittsburgh, PA  15213-3890<br>  (412) 268-4387, fax: (412) 268-7395<br>  tech-transfer@andrew.cmu.edu<br><br>4. Redistributions of any form whatsoever must retain the following acknowledgment:<br>"This product includes software developed by Computing Services at Carnegie Mellon University (http://www.cmu.edu/computing/)."<br><br>CARNEGIE MELLON UNIVERSITY DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL CARNEGIE MELLON UNIVERSITY BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. |
| 17 | Copyright (c) 1984-2000 Carnegie Mellon University. All rights reserved.<br><br>Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:<br><br>1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.<br><br>2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.<br><br>3. The name "Carnegie Mellon University" must not be used to endorse or promote products derived from this software without prior written permission. For permission or any legal details, please contact<br>  Office of Technology Transfer<br>  Carnegie Mellon University<br>  5000 Forbes Avenue<br>  Pittsburgh, PA  15213-3890<br>  (412) 268-4387, fax: (412) 268-7395<br>  tech-transfer@andrew.cmu.edu<br><br>4. Redistributions of any form whatsoever must retain the following acknowledgment:<br>"This product includes software developed by Computing Services at Carnegie Mellon University (http://www.cmu.edu/computing/)."<br><br>CARNEGIE MELLON UNIVERSITY DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL CARNEGIE MELLON UNIVERSITY BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. |

| License number | License |
|---|---|
| 18 | Chat -- a program for automatic session establishment (i.e. dial the phone and log in). |

Standard termination codes:
 0 - successful completion of the script
 1 - invalid argument, expect string too large, etc.
 2 - error on an I/O operation or fatal error condition.
 3 - timeout waiting for a simple string.
 4 - the first string declared as "ABORT"
 5 - the second string declared as "ABORT"
 6 - ... and so on for successive ABORT strings.

This software is in the public domain.

-----------------
12-May-99 added a feature to read data to be sent from a file,
if the send string starts with @.  Idea from gpk <gpk@onramp.net>.

added -T and -U option and \T and \U substitution to pass a phone
number into chat script. Two are needed for some ISDN TA applications.
Keith Dart <kdart@cisco.com>


Added SAY keyword to send output to stderr.
    This allows to turn ECHO OFF and to output specific, user selected,
    text to give progress messages. This best works when stderr
    exists (i.e.: pppd in nodetach mode).

Added HANGUP directives to allow for us to be called
    back. When HANGUP is set to NO, chat will not hangup at HUP signal.
    We rely on timeouts in that case.

    Added CLR_ABORT to clear previously set ABORT string. This has been
    dictated by the HANGUP above as "NO CARRIER" (for example) must be
    an ABORT condition until we know the other host is going to close
    the connection for call back. As soon as we have completed the
    first stage of the call back sequence, "NO CARRIER" is a valid, non
    fatal string. As soon as we got called back (probably get "CONNECT"),
    we should re-arm the ABORT "NO CARRIER". Hence the CLR_ABORT command.
    Note that CLR_ABORT packs the abort_strings[] array so that we do not
    have unused entries not being reclaimed.

    In the same vein as above, added CLR_REPORT keyword.

    Allow for comments. Line starting with '#' are comments and are
    ignored. If a '#' is to be expected as the first character, the
    expect string must be quoted.


Francis Demierre <Francis@SwissMail.Com>
Thu May 15 17:15:40 MET DST 1997


    Added -r "report file" switch & REPORT keyword.
        Robert Geer <bgeer@xmission.com>

    Added -s "use stderr" and -S "don't use syslog" switches.
        June 18, 1997
        Karl O. Pinc <kop@meme.com>


Added -e "echo" switch & ECHO keyword
Dick Streefland <dicks@tasking.nl>

| License number | License |
|---|---|

```
                    Considerable updates and modifications by
                    Al Longyear <longyear@pobox.com>
                    Paul Mackerras <paulus@cs.anu.edu.au>


                    The original author is:

                    Karl Fox <karl@MorningStar.Com>
                    Morning Star Technologies, Inc.
                    1760 Zollinger Road
                    Columbus, OH  43221
                    (614)451-1883
```

## L.2  Files

The following files are contained in the u-blox Android RIL driver package and are distributed under the mentioned licenses. Please refer to Licenses for the license terms.

| Files | License number | MD5 |
|---|---|---|
| Android_10.0.X\device\google\wahoo\system.prop | 15 | cc4e64ecc3c97144b5e029b00d63f2c5 |
| Android_10.0.X\device\google\wahoo\device.mk | 15 | 6b322eda215b3aaea153177c80e67caa |
| Android_10.0.X\external\ppp\chat\Android.mk | 18 | fdfcd78d9ff7723600629540bcd1e6a8 |
| Android_10.0.X\external\ppp\chat\chat.c | 18 | 193e91fed2a5802f4c1d725893889bb5 |
| Android_10.0.X\external\ppp\pppd\Android.bp | 15 | aca426c271b29bcb0f85f2a0f40accea |
| Android_10.0.X\external\ppp\pppd\main.c | 17 | 19970e6a895501633e3f40f36ec46ff5 |
| Android_10.0.X\external\ppp\pppd\pathnames.h | 17 | 6de33c24f24f71ff61c383a6aa0294d3 |
| Android_10.0.X\external\ppp\pppd\pppd.h | 17 | c2800ca084537644858b27b43a174673 |
| Android_10.0.X\external\ppp\pppd\sys-linux.c | 16 | a33b0c7a687029cfd2378c705042470c |
| Android_10.0.X\packages\apps\Settings\res\layout\riloemhook_layout.xml | 4 | 3cbba3695170872a1a211c3780fddb2e |
| Android_10.0.X\packages\apps\Settings\src\com\android\settings\RilOemHookTest.java | 5 | bffe153708a8a0d60797d4c155dd380d |
| Android_10.0.X\system\core\init\first_stage_init.cpp | 15 | 856641ea47e3b7f101e1595227b8b188 |
| Android_10.0.X\system\core\libcutils\properties.cpp | 5 | 0c548b2f926f59823e00180dec007c4d |
| Android_10.0.X\system\core\liblog\logger_write.cpp | 7 | f3974e341c80dc1208b41a7bb700712c |
| Android_10.0.X\system\core\libnetutils\Android.mk | 2 | 6327cdc3373aa0eaf2f175b4093ef272 |
| Android_10.0.X\system\netd\netutils_wrappers\NetUtilsWrapper-1.0.cpp | 14 | a70bdf04cd9107812a01e0208c8e3700 |
| Android_11.0.X\device\linaro\hikey\system.prop | 15 | fb0a70bfb4d15f1cdb76a3ee60949109 |
| Android_11.0.X\external\ppp\chat\Android.mk | 18 | fdfcd78d9ff7723600629540bcd1e6a8 |
| Android_11.0.X\external\ppp\chat\chat.c | 18 | 32196c4436a3de31dc9831034fc43503 |
| Android_11.0.X\external\ppp\pppd\Android.bp | 15 | aca426c271b29bcb0f85f2a0f40accea |
| Android_11.0.X\external\ppp\pppd\main.c | 17 | 408c3a94d910a6a50fc725f67aebf90d |
| Android_11.0.X\external\ppp\pppd\pathnames.h | 17 | 13089f7e853d46f134922dcb9bc6ae1f |
| Android_11.0.X\external\ppp\pppd\pppd.h | 17 | c2800ca084537644858b27b43a174673 |
| Android_11.0.X\external\ppp\pppd\sys-linux.c | 16 | ca93017c482404af657b93fe882ec29a |
| Android_11.0.X\packages\apps\Settings\res\layout\riloemhook_layout.xml | 4 | 3cbba3695170872a1a211c3780fddb2e |
| Android_11.0.X\packages\apps\Settings\src\com\android\settings\RilOemHookTest.java | 5 | c84f06e396baf70486444b532a42fd2b |

| Files | License number | MD5 |
|---|---|---|
| Android_11.0.X\system\core\init\first_stage_init.cpp | 15 | 7f1fbbc9daa43c1ca07ed412a3a21318 |
| Android_11.0.X\system\core\libcutils\properties.cpp | 5 | 4486827bb623f29faf55f67469292d6d |
| Android_11.0.X\system\core\liblog\logger_write.cpp | 7 | cde9dc181e24b5b85038473b09ab090c |
| Android_11.0.X\system\core\libnetutils\Android.mk | 2 | 338c67ed054a137cf686acfde5a8023a |
| Android_11.0.X\system\netd\netutils_wrappers\NetUtilsWrapper-1.0.cpp | 14 | a70bdf04cd9107812a01e0208c8e3700 |
| Android_5.0.X\device\lge\hammerhead\AndroidProducts.mk | 10 | e9ed37dba1d76d4ae3526c88a251e07e |
| Android_5.0.X\external\ppp\chat\Android.mk | 18 | cd8b1832697af9441bb76276d1b25f02 |
| Android_5.0.X\external\ppp\chat\chat.c | 18 | 193e91fed2a5802f4c1d725893889bb5 |
| Android_5.0.X\external\ppp\pppd\Android.mk | 9 | 7db01f89e851eb75f658760077f94909 |
| Android_5.0.X\external\ppp\pppd\main.c | 17 | b088c8d4b6ad67cd59835c4f725f2cc8 |
| Android_5.0.X\external\ppp\pppd\pathnames.h | 17 | 2fdf7d5853fd5a5c8082ec162b1ed3a0 |
| Android_5.0.X\external\ppp\pppd\pppd.h | 17 | 2ac7419cf1c71412432a39fb6aee2d9a |
| Android_5.0.X\external\ppp\pppd\sys-linux.c | 16 | 958bcb009d252449bba5b52152e99116 |
| Android_5.0.X\packages\apps\Settings\res\layout\riloemhook_layout.xml | 4 | 6a1414e0c7a85ffd1526689fbdececd1 |
| Android_5.0.X\packages\apps\Settings\src\com\android\settings\RilOemHookTest.java | 5 | 4da7d895b4d2c1d29f76341829bf40f7 |
| Android_5.0.X\system\core\init\init.c | 8 | 51f52003ad3382cea90bb5eb1d787b78 |
| Android_5.0.X\system\core\liblog\logd_write.c | 6 | e092a5004f6fd43c5d2776382f3dcc1e |
| Android_6.0.X\device\lge\hammerhead\AndroidProducts.mk | 10 | 389fe1f62ed2c0161531814773eef4cd |
| Android_6.0.X\external\ppp\chat\Android.mk | 18 | cd8b1832697af9441bb76276d1b25f02 |
| Android_6.0.X\external\ppp\chat\chat.c | 18 | 193e91fed2a5802f4c1d725893889bb5 |
| Android_6.0.X\external\ppp\pppd\Android.mk | 9 | c9526026b302fba33657033c39de08b7 |
| Android_6.0.X\external\ppp\pppd\main.c | 17 | e4df15ff11a3610a4db9d39cf7b6fe81 |
| Android_6.0.X\external\ppp\pppd\pathnames.h | 17 | 21449e4962967103c2b549fdb317c566 |
| Android_6.0.X\external\ppp\pppd\pppd.h | 17 | c2800ca084537644858b27b43a174673 |
| Android_6.0.X\external\ppp\pppd\sys-linux.c | 16 | a33b0c7a687029cfd2378c705042470c |
| Android_6.0.X\system\core\init\init.cpp | 8 | ea9639489038aff532679e74c5802095 |
| Android_6.0.X\system\core\liblog\logd_write.c | 6 | a8b1be0c992d2d2a2685e77ccac3adc1 |
| Android_7.0.X\device\lge\bullhead\AndroidProducts.mk | 11 | 22dac47cd9eefa7875de6eea359bdbcd |
| Android_7.0.X\device\lge\bullhead\device.mk | 12 | 0212c3ff88ae9340604e7cd1fc26256f |
| Android_7.0.X\external\ppp\chat\Android.mk | 18 | 5b686d5b6d8a7a75274c985af4101bca |
| Android_7.0.X\external\ppp\chat\chat.c | 18 | 193e91fed2a5802f4c1d725893889bb5 |
| Android_7.0.X\external\ppp\pppd\Android.mk | 9 | c9526026b302fba33657033c39de08b7 |
| Android_7.0.X\external\ppp\pppd\main.c | 17 | e4df15ff11a3610a4db9d39cf7b6fe81 |
| Android_7.0.X\external\ppp\pppd\pathnames.h | 17 | 21449e4962967103c2b549fdb317c566 |
| Android_7.0.X\external\ppp\pppd\pppd.h | 17 | c2800ca084537644858b27b43a174673 |
| Android_7.0.X\external\ppp\pppd\sys-linux.c | 16 | a33b0c7a687029cfd2378c705042470c |
| Android_7.0.X\packages\apps\Settings\res\layout\riloemhook_layout.xml | 4 | 3cbba3695170872a1a211c3780fddb2e |
| Android_7.0.X\packages\apps\Settings\src\com\android\settings\RilOemHookTest.java | 5 | bffe153708a8a0d60797d4c155dd380d |
| Android_7.0.X\system\core\init\init.cpp | 8 | 268d080077c142fe09d6b6817c5b9fa2 |
| Android_7.0.X\system\core\liblog\logger_write.c | 7 | 204127239dfe07dc34fb14e824e73e88 |
| Android_8.0.X\device\lge\bullhead\AndroidProducts.mk | 11 | b44fab2be7bab154fc4bf5b67d441105 |

| Files | License number | MD5 |
|---|---|---|
| Android_8.0.X\device\lge\bullhead\device.mk | 13 | 798beb0d957601da5bb7dfb50f5d3b3e |
| Android_8.0.X\external\ppp\chat\Android.mk | 18 | cd8b1832697af9441bb76276d1b25f02 |
| Android_8.0.X\external\ppp\chat\chat.c | 18 | 193e91fed2a5802f4c1d725893889bb5 |
| Android_8.0.X\external\ppp\pppd\Android.mk | 9 | c9526026b302fba33657033c39de08b7 |
| Android_8.0.X\external\ppp\pppd\main.c | 17 | e4df15ff11a3610a4db9d39cf7b6fe81 |
| Android_8.0.X\external\ppp\pppd\pathnames.h | 17 | 21449e4962967103c2b549fdb317c566 |
| Android_8.0.X\external\ppp\pppd\pppd.h | 17 | c2800ca084537644858b27b43a174673 |
| Android_8.0.X\external\ppp\pppd\sys-linux.c | 16 | a33b0c7a687029cfd2378c705042470c |
| Android_8.0.X\packages\apps\Settings\res\layout\riloemhook_layout.xml | 4 | 3cbba3695170872a1a211c3780fddb2e |
| Android_8.0.X\packages\apps\Settings\src\com\android\settings\RilOemHookTest.java | 5 | bffe153708a8a0d60797d4c155dd380d |
| Android_8.0.X\system\core\init\init.cpp | 8 | 6d442c0da18b6e739f8c29011a15a343 |
| Android_8.0.X\system\core\liblog\logger_write.c | 7 | 660c9945496624e1d315baec10253a67 |
| Android_8.1.X\device\google\wahoo\device.mk | 13 | de58379ce7b52dc0852320b685d10ba3 |
| Android_8.1.X\device\google\wahoo\system.prop | 9 | 6df55062f69ad28c2a889e0cf575b30f |
| Android_8.1.X\external\ppp\chat\Android.mk | 18 | fab46fb4b356246ccf0b8fdcd9192a4a |
| Android_8.1.X\external\ppp\chat\chat.c | 18 | 193e91fed2a5802f4c1d725893889bb5 |
| Android_8.1.X\external\ppp\pppd\Android.mk | 9 | 105fbecb6732a88aafd6435ffbc2bf89 |
| Android_8.1.X\external\ppp\pppd\main.c | 17 | e4df15ff11a3610a4db9d39cf7b6fe81 |
| Android_8.1.X\external\ppp\pppd\pathnames.h | 17 | 21449e4962967103c2b549fdb317c566 |
| Android_8.1.X\external\ppp\pppd\pppd.h | 17 | c2800ca084537644858b27b43a174673 |
| Android_8.1.X\external\ppp\pppd\sys-linux.c | 16 | a33b0c7a687029cfd2378c705042470c |
| Android_8.1.X\packages\apps\Settings\res\layout\riloemhook_layout.xml | 4 | 3cbba3695170872a1a211c3780fddb2e |
| Android_8.1.X\packages\apps\Settings\src\com\android\settings\RilOemHookTest.java | 5 | bffe153708a8a0d60797d4c155dd380d |
| Android_8.1.X\system\core\init\init.cpp | 8 | 6d442c0da18b6e739f8c29011a15a343 |
| Android_8.1.X\system\core\liblog\logger_write.c | 7 | 660c9945496624e1d315baec10253a67 |
| Android_8.1.X\system\netd\netutils_wrappers\NetUtilsWrapper-1.0.cpp | 14 | 092e3bcbfd64bba6cc028b30547aca75 |
| Android_9.0.X\device\google\wahoo\device.mk | 13 | 98480a3e21e17131bc938636f1d3d54c |
| Android_9.0.X\device\google\wahoo\system.prop | 9 | 5ab2562a562a5db8782704ffff7da538 |
| Android_9.0.X\external\ppp\chat\Android.mk | 18 | fdfcd78d9ff7723600629540bcd1e6a8 |
| Android_9.0.X\external\ppp\chat\chat.c | 18 | 193e91fed2a5802f4c1d725893889bb5 |
| Android_9.0.X\external\ppp\pppd\Android.mk | 9 | 105fbecb6732a88aafd6435ffbc2bf89 |
| Android_9.0.X\external\ppp\pppd\main.c | 17 | 19970e6a895501633e3f40f36ec46ff5 |
| Android_9.0.X\external\ppp\pppd\pathnames.h | 17 | 833238024f4d27f7a7c756a9b5f21907 |
| Android_9.0.X\external\ppp\pppd\pppd.h | 17 | c2800ca084537644858b27b43a174673 |
| Android_9.0.X\external\ppp\pppd\sys-linux.c | 16 | a33b0c7a687029cfd2378c705042470c |
| Android_9.0.X\packages\apps\Settings\res\layout\riloemhook_layout.xml | 4 | 3cbba3695170872a1a211c3780fddb2e |
| Android_9.0.X\packages\apps\Settings\src\com\android\settings\RilOemHookTest.java | 5 | bffe153708a8a0d60797d4c155dd380d |
| Android_9.0.X\system\core\init\init.cpp | 8 | efe50b36cd999dadb667e90fc5c52874 |
| Android_9.0.X\system\core\libcutils\properties.cpp | 5 | 0c548b2f926f59823e00180dec007c4d |

| Files | License number | MD5 |
|---|---|---|
| Android_9.0.X\system\core\liblog\logger_write.c | 7 | 67d04e2010a26e4cedd78f9717236be4 |
| Android_9.0.X\system\core\libnetutils\Android.mk | 2 | c83c6e05ffa19cf2649cca1bbd2b4ffc |
| Android_9.0.X\system\netd\netutils_wrappers\NetUtilsWrapper-1.0.cpp | 14 | 092e3bcbfd64bba6cc028b30547aca75 |
| Common\build\target\product\rootdir\etc\ppp\peers\cdma-gprs1 | NA | 2516e1089b330287b134751eea925649 |
| Common\build\target\product\rootdir\etc\ppp\peers\gprs1 | NA | 214f381b8e157bf88cee02298b3caade |
| Common\build\target\product\rootdir\etc\ppp\peers\gprs2 | NA | 13a73d2342ca27bebdade076818c77f0 |
| Common\build\target\product\rootdir\etc\ppp\peers\gprs3 | NA | 1e1742529ce5fe577e2c90795e5099c2 |
| Common\build\target\product\rootdir\etc\ppp\peers\gprs4 | NA | 8c1e9f8e5cbf42e830dda8a73a23587d |
| Common\build\target\product\rootdir\etc\ppp\cdma-chat-isp | NA | e41299885f1a62db61eb8564f9355ff6 |
| Common\build\target\product\rootdir\etc\ppp\chap-secrets | NA | 507862459b23157575bab12dd77df0db |
| Common\build\target\product\rootdir\etc\ppp\chat-isp1 | NA | e5a561b73a8dad83c371980dc2af021c |
| Common\build\target\product\rootdir\etc\ppp\chat-isp2 | NA | 4fe01e6277c35d4575f76306e0745f94 |
| Common\build\target\product\rootdir\etc\ppp\chat-isp3 | NA | 0fd81196de537971ded967a609b74598 |
| Common\build\target\product\rootdir\etc\ppp\chat-isp4 | NA | 09e2bfbfe1c11353098e07fdbe5e5ecd |
| Common\build\target\product\rootdir\etc\ppp\ip-down | 2 | fb19dab118e943d64cd14b651ac81023 |
| Common\build\target\product\rootdir\etc\ppp\ip-up | 2 | 13cbce5652ded39bf269fad6ae8917c0 |
| Common\build\target\product\rootdir\etc\ppp\ipv6-down | 2 | 3858e8bfe12f470ec0f96eba7776b2d3 |
| Common\build\target\product\rootdir\etc\ppp\ipv6-up | 2 | c339885549f6c030804264e5b6ce6491 |
| Common\build\target\product\rootdir\etc\ppp\pap-secrets | NA | 507862459b23157575bab12dd77df0db |
| Common\build\target\product\rootdir\etc\uril\repository.txt | 1 | c22b813afd3bddc6e2a4b35a827c5719 |
| Common\build\target\product\rootdir\etc\init.gprs-pppd | 2 | f6870b6f2919e5fc5a5bf04bf808ccb1 |
| Common\build\target\product\rootdir\etc\init_data | 2 | 49823eaf7c9e58012ba4b8343ef5b393 |
| Common\build\target\product\rootdir\etc\init_data_android_6 | 2 | 06b8902e39ff04687470fda4277f2cd1 |
| Common\build\target\product\rootdir\etc\static_apn | 2 | 9f4b5bfa66b49c3014f4112d98f66f7d |
| Common\build\target\product\rootdir\etc\stop_data | 2 | 92d0a2a09c4c75046f57d14d2a6fbe16 |
| Common\build\target\product\rootdir\etc\stop_data_android_6 | 2 | f36ad01c74ae6e4e43d640beca3d8420 |
| Common\build\target\product\rootdir\etc\stop_muxd | 2 | dff5c8e886cb5c6be3597fe6459dabbd |
| Common\build\target\product\rootdir\etc\stop_pppd | 2 | ba497597a6cfe7a8e7a20e5df3fc5dfe |
| Common\build\target\product\core_ublox.mk | 2 | b56372a09cfbb643bb848c41f393fde9 |
| Common\hardware\ril\ublox_ril\CORE\ND\MODEMS\silo_data_inf.cpp | 1 | 61edc32a088aa3d3ec3bfb96355667bd |
| Common\hardware\ril\ublox_ril\CORE\ND\MODEMS\silo_data_inf.h | 1 | 2b79898b359dc9f1d4dc608d7e88c616 |
| Common\hardware\ril\ublox_ril\CORE\ND\MODEMS\silo_network_inf.cpp | 1 | aac9225fd500473976b8a31998857cee |
| Common\hardware\ril\ublox_ril\CORE\ND\MODEMS\silo_network_inf.h | 1 | 222ff53af7e9d468f9a2442be77f54a7 |
| Common\hardware\ril\ublox_ril\CORE\ND\MODEMS\silo_phonebook_inf.cpp | 1 | e87726e0b90532bb3241dbc035c8df15 |
| Common\hardware\ril\ublox_ril\CORE\ND\MODEMS\silo_phonebook_inf.h | 1 | 38b8643c6b2f04afc5dd3f977f14e12f |
| Common\hardware\ril\ublox_ril\CORE\ND\MODEMS\silo_sim_inf.cpp | 1 | 9bc882bb046ab539fa09b8ba2e688107 |
| Common\hardware\ril\ublox_ril\CORE\ND\MODEMS\silo_sim_inf.h | 1 | 47b8e9c738f12527828d1cb31b32bcb9 |
| Common\hardware\ril\ublox_ril\CORE\ND\MODEMS\silo_sms_inf.cpp | 1 | ec7f66d43996a876d5fc2f9c34028a18 |

| Files | License number | MD5 |
|---|---|---|
| Common\hardware\ril\ublox_ril\CORE\ND\MODEMS\silo_sms_inf.h | 1 | 463fac92dbefe259e6cce65ac26f48f4 |
| Common\hardware\ril\ublox_ril\CORE\ND\MODEMS\silo_voice_inf.cpp | 1 | 349ddace338a11c000e3e3098f4f81a6 |
| Common\hardware\ril\ublox_ril\CORE\ND\MODEMS\silo_voice_inf.h | 1 | dc5fdc22ce1375e07a0e441fc0cd6942 |
| Common\hardware\ril\ublox_ril\CORE\ND\MODEMS\te_cdma.cpp | 2 | 2d3d5bc49573b7f9169435a2e3a54a9a |
| Common\hardware\ril\ublox_ril\CORE\ND\MODEMS\te_cdma.h | 2 | 1a3758ac2fcc3d3ae2bbff056c4462d2 |
| Common\hardware\ril\ublox_ril\CORE\ND\MODEMS\te_inf_u120.cpp | 1 | a863cb8a5d070e011120f0df6bc0d973 |
| Common\hardware\ril\ublox_ril\CORE\ND\MODEMS\te_inf_u120.h | 1 | 7c2f9afdc17a54c598010d704ff49899 |
| Common\hardware\ril\ublox_ril\CORE\ND\MODEMS\te_modem_toby_R2.cpp | 2 | 74438a3b465f3a1ac217a6c3c72fe102 |
| Common\hardware\ril\ublox_ril\CORE\ND\MODEMS\te_modem_toby_R2.h | 2 | d806da1c876d8c7930dc51ce768758a7 |
| Common\hardware\ril\ublox_ril\CORE\ND\MODEMS\ublox_rat.cpp | 2 | d2ea9c1c2b0dab6360f840d82fa1b7a2 |
| Common\hardware\ril\ublox_ril\CORE\ND\MODEMS\ublox_rat.h | 2 | a433a962301ef61f82bfe84264193a12 |
| Common\hardware\ril\ublox_ril\CORE\ND\callbacks.cpp | 1 | 6f52704cdd72158a676abeb570b0d0dc |
| Common\hardware\ril\ublox_ril\CORE\ND\callbacks.h | 1 | dc4c204b220d0690c6546fb45c4fb7f6 |
| Common\hardware\ril\ublox_ril\CORE\ND\channel_nd.cpp | 1 | 21b4db9cd586e181194efc350c79d746 |
| Common\hardware\ril\ublox_ril\CORE\ND\channel_nd.h | 1 | 4f715b89e353a02568f321c304bd1801 |
| Common\hardware\ril\ublox_ril\CORE\ND\file_ops.cpp | 1 | 16963b66bce012d65850034387a94e5f |
| Common\hardware\ril\ublox_ril\CORE\ND\misc.c | 4 | 9171676a551258a5ffd75ef4958307c9 |
| Common\hardware\ril\ublox_ril\CORE\ND\misc.h | 4 | e0671b1f37946637227ffdccb6380281 |
| Common\hardware\ril\ublox_ril\CORE\ND\nd_structs.h | 1 | 661d808ff8dd76fabc57f402b31ece03 |
| Common\hardware\ril\ublox_ril\CORE\ND\omap_csmi.h | 3 | bd6b43298e07a314b12055a978b771cd |
| Common\hardware\ril\ublox_ril\CORE\ND\radio_state.cpp | 1 | 8129101c2ef1c15bcacf595f51d395b6 |
| Common\hardware\ril\ublox_ril\CORE\ND\rildmain.cpp | 1 | 7b425709e0f8582acc4f201527be5d8a |
| Common\hardware\ril\ublox_ril\CORE\ND\rildmain.h | 1 | 7a2fd0aaf6b3a31920fab44d7a1921a6 |
| Common\hardware\ril\ublox_ril\CORE\ND\ril_result_nd.cpp | 1 | af1c0e2bcee16d8a837ebad9ce28c0af |
| Common\hardware\ril\ublox_ril\CORE\ND\ril_result_nd.h | 1 | c14db07ce68b24824f5286375d912445 |
| Common\hardware\ril\ublox_ril\CORE\ND\silo_data.cpp | 1 | cedf2041caf3fae444c46c295b8ac56b |
| Common\hardware\ril\ublox_ril\CORE\ND\silo_data.h | 1 | 41ffb1394089b70bb4ca327b91eaaa9b |
| Common\hardware\ril\ublox_ril\CORE\ND\silo_factory.cpp | 1 | 05d46211d1286791ad7f0f30af099933 |
| Common\hardware\ril\ublox_ril\CORE\ND\silo_factory.h | 1 | 4d2ad53e5b6067836e5d29a9597286f1 |
| Common\hardware\ril\ublox_ril\CORE\ND\silo_network.cpp | 1 | 49eccd369ccd234bc4b09148fd88b5b6 |
| Common\hardware\ril\ublox_ril\CORE\ND\silo_network.h | 1 | 20158924442e2f3ac4e866a59d6e2eb7 |
| Common\hardware\ril\ublox_ril\CORE\ND\silo_phonebook.cpp | 1 | 966c35de842f1aee0adfd322b3f06de8 |
| Common\hardware\ril\ublox_ril\CORE\ND\silo_phonebook.h | 1 | 8dad4636112326bbccbd251e2aca093e |
| Common\hardware\ril\ublox_ril\CORE\ND\silo_sim.cpp | 1 | 6b59831cb9e1265ec3565b052b05f9b0 |
| Common\hardware\ril\ublox_ril\CORE\ND\silo_sim.h | 1 | 8eed77972b135c6f7d89eb34e20218b1 |
| Common\hardware\ril\ublox_ril\CORE\ND\silo_sms.cpp | 1 | e472fbf81e75e532a49059f571d2dc82 |
| Common\hardware\ril\ublox_ril\CORE\ND\silo_sms.h | 1 | a1dcff1d63f9a0eac066714d938686c1 |
| Common\hardware\ril\ublox_ril\CORE\ND\silo_voice.cpp | 1 | 8b7d96a1d30470e5543c6f06cb702476 |
| Common\hardware\ril\ublox_ril\CORE\ND\silo_voice.h | 1 | cd72c76c7c4994e92a422ba7721a42f7 |

| Files | License number | MD5 |
|-------|----------------|-----|
| Common\hardware\ril\ublox_ril\CORE\ND\sync_ops.cpp | 1 | d735c8277cdf4b6bfe62ee6699a6db52 |
| Common\hardware\ril\ublox_ril\CORE\ND\systemmanager.cpp | 1 | 9e828662967a4e3577c227ee562d980a |
| Common\hardware\ril\ublox_ril\CORE\ND\systemmanager.h | 1 | ac1ef53ff4a46c054f849c3a257d7df7 |
| Common\hardware\ril\ublox_ril\CORE\ND\te.cpp | 1 | ebcb94794aa7bc0a366539f95cd90fc7 |
| Common\hardware\ril\ublox_ril\CORE\ND\te.h | 1 | 848e768d14170b4edcfded723ef4de4d |
| Common\hardware\ril\ublox_ril\CORE\ND\te_base.cpp | 1 | f2657829281d44c08afb7d4183280177 |
| Common\hardware\ril\ublox_ril\CORE\ND\te_base.h | 1 | edcc622b909b5788da33cbff8f5706ef |
| Common\hardware\ril\ublox_ril\CORE\ND\thread_ops.cpp | 1 | dcf5cf2c67fcb178a278d55c0d2495f0 |
| Common\hardware\ril\ublox_ril\CORE\Android.mk | 1 | 2ee7a92c4b94d6029fc1a42fcde8abab |
| Common\hardware\ril\ublox_ril\CORE\channelbase.cpp | 1 | 5db67b01a22202e39f2166b00ee5bb75 |
| Common\hardware\ril\ublox_ril\CORE\channelbase.h | 1 | 01ee7ee41c9e499774e5cb1ad0d29e90 |
| Common\hardware\ril\ublox_ril\CORE\channel_atcmd.cpp | 1 | 8c68ab931238bab3a4acf4f25a94ce1e |
| Common\hardware\ril\ublox_ril\CORE\channel_atcmd.h | 1 | cfe16d8ea2428fc95eb1895373cc361b |
| Common\hardware\ril\ublox_ril\CORE\channel_data.cpp | 1 | 542243d24e1dbe6af544b89c6e0ed143 |
| Common\hardware\ril\ublox_ril\CORE\channel_data.h | 1 | d03485c76a16e3abb9d0160c85b6fb29 |
| Common\hardware\ril\ublox_ril\CORE\channel_DLC2.cpp | 1 | 17ff06e9d95950125186f97e11e5fb3c8 |
| Common\hardware\ril\ublox_ril\CORE\channel_DLC2.h | 1 | ebe8f1207d77125683f7eabe94585540 |
| Common\hardware\ril\ublox_ril\CORE\channel_DLC6.cpp | 1 | 2c7f8e7244ee60ac903fb57c94520412 |
| Common\hardware\ril\ublox_ril\CORE\channel_DLC6.h | 1 | e98c0c954a1619591b58b7d63cce6a12 |
| Common\hardware\ril\ublox_ril\CORE\channel_DLC8.cpp | 1 | 0392e8411187fb38ca686108c0eee3cc |
| Common\hardware\ril\ublox_ril\CORE\channel_DLC8.h | 1 | 023d4b1c21c526db60ca07f87bb13ed0 |
| Common\hardware\ril\ublox_ril\CORE\cmdcontext.cpp | 1 | ab16574f2d02dc378e217025f46aafc0 |
| Common\hardware\ril\ublox_ril\CORE\cmdcontext.h | 1 | a7ab7596443fe2cb860fd5a653ca1d76 |
| Common\hardware\ril\ublox_ril\CORE\command.cpp | 1 | cb9014355fbf66ef0f9cf486a4cb86ba |
| Common\hardware\ril\ublox_ril\CORE\command.h | 1 | 96488cab4594a6b48776911e4ab47f4b |
| Common\hardware\ril\ublox_ril\CORE\com_init_index.h | 1 | 6bad9806e8d9bfd0e2602bd781da4a9d |
| Common\hardware\ril\ublox_ril\CORE\file_ops.h | 1 | 3c1b65c21b9a4242091f38d0dd17aac1 |
| Common\hardware\ril\ublox_ril\CORE\globals.cpp | 1 | 5754f1dd5b3790f2ba018fc64844b5bb |
| Common\hardware\ril\ublox_ril\CORE\globals.h | 1 | 12e7dc88b6ce00d95af5a7619f2c8762 |
| Common\hardware\ril\ublox_ril\CORE\mem_ops.h | 1 | 8ff5102aace1b2952a80890886f85abf |
| Common\hardware\ril\ublox_ril\CORE\notification.h | 1 | 7c18c7d299370361c4f3c10ea5bec27e |
| Common\hardware\ril\ublox_ril\CORE\oemhookids.h | 1 | 79ebcc76398c94dbee10330557e44bd6 |
| Common\hardware\ril\ublox_ril\CORE\port.cpp | 1 | 7826f322ad720ca49a49034638358700 |
| Common\hardware\ril\ublox_ril\CORE\port.h | 1 | e745bbfac81876b5a462af7436233f2b |
| Common\hardware\ril\ublox_ril\CORE\radio_state.h | 1 | da86e6109cf5439119c314f519c28399 |
| Common\hardware\ril\ublox_ril\CORE\repository.h | 1 | 31a8e8a12d7793213a276789f421f04a |
| Common\hardware\ril\ublox_ril\CORE\request_id.h | 1 | e3d7fa2662a301223f67915c62a8f08c |
| Common\hardware\ril\ublox_ril\CORE\request_info_table.cpp | 1 | 76e3816acb07bfe92bab75f9d7fcccd1 |
| Common\hardware\ril\ublox_ril\CORE\request_info_table.h | 1 | 70addbb5637d9f364301a108a9f756f6 |
| Common\hardware\ril\ublox_ril\CORE\response.cpp | 1 | f2c305a15d4c09794c2a67130b4d07c4 |
| Common\hardware\ril\ublox_ril\CORE\response.h | 1 | b2ec9071418d127e90bcd1076940a1a7 |
| Common\hardware\ril\ublox_ril\CORE\rilchannels.cpp | 1 | 8df1b653eea9178d4dff566c1c49dd27 |
| Common\hardware\ril\ublox_ril\CORE\rilchannels.h | 1 | 807945f75832b26b728f9913b146b0ff |
| Common\hardware\ril\ublox_ril\CORE\rillog.h | 1 | 5e207490d8cc167e9d6cc96e628c3dc8 |

| Files | License number | MD5 |
|---|---|---|
| Common\hardware\ril\ublox_ril\CORE\rilqueue.h | 1 | 409191694acea88b225a64f5bfd104af |
| Common\hardware\ril\ublox_ril\CORE\ril_result.h | 1 | 42273886a7fe5afdeb0900e34b0f5160 |
| Common\hardware\ril\ublox_ril\CORE\rril.h | 1 | fec032759339da5a8dc67e10ebd402cd |
| Common\hardware\ril\ublox_ril\CORE\silo.cpp | 1 | cb6e9f29b1e5cb1eeb7cd4c1e1c334da |
| Common\hardware\ril\ublox_ril\CORE\silo.h | 1 | ae4e1e8afdc77dd718af2cf299265219 |
| Common\hardware\ril\ublox_ril\CORE\sync_ops.h | 1 | 315856e62d490545a79d1d2bf568e1d3 |
| Common\hardware\ril\ublox_ril\CORE\thread_manager.cpp | 1 | e8f0115b9884430855a44f7110195a99 |
| Common\hardware\ril\ublox_ril\CORE\thread_manager.h | 1 | e97bdac0d3a1d39cc23d4c3e2fe6ae75 |
| Common\hardware\ril\ublox_ril\CORE\thread_ops.h | 1 | e7ac6de374f2e57293db8725ec45f600 |
| Common\hardware\ril\ublox_ril\CORE\types.h | 1 | b66068f84e0a74e6ea6b2ce6d81d21ff |
| Common\hardware\ril\ublox_ril\CORE\util.h | 1 | 35538f228de33892946d0d7a4ace30a1 |
| Common\hardware\ril\ublox_ril\INC\rril_OEM.h | 1 | 45847490ec374d6041f2b4352594c129 |
| Common\hardware\ril\ublox_ril\OEM\ND\Android.mk | 1 | 29ea386e54d44e2d79de40aef5ab2aaf |
| Common\hardware\ril\ublox_ril\OEM\ND\oem_android_init.cpp | 1 | 3e6d4e7d91bbd3eaebd1e41865945e18 |
| Common\hardware\ril\ublox_ril\UTIL\ND\Android.mk | 1 | 1481793687a9807641110380e3c7535a |
| Common\hardware\ril\ublox_ril\UTIL\ND\extract.cpp | 1 | 3a2ea01c3956149acaf7c3249a16ba0c |
| Common\hardware\ril\ublox_ril\UTIL\ND\extract.h | 1 | 9430d346b4a2b48840a9478c6985bd0a |
| Common\hardware\ril\ublox_ril\UTIL\ND\notification.cpp | 1 | 84d26280fe70fe6470593e44f20045ad |
| Common\hardware\ril\ublox_ril\UTIL\ND\repository.cpp | 1 | 2c3d20f4b41287a4c1292dea10e4f825 |
| Common\hardware\ril\ublox_ril\UTIL\ND\rillog.cpp | 1 | 1471e5de821ddcee9d87d42bc2797994 |
| Common\hardware\ril\ublox_ril\UTIL\ND\util.cpp | 1 | dc467412baa4f3b4f610baa75310f3ca |
| Common\system\core\rootdir\init.ublox.rc | 2 | 35e14b1ee8e8cc2fdeaa471aaed14f04 |

# M  Android RIL integration FAQ

**Q: RIL cannot open the COM port to connect ttyACMX or ttyUSBX.**

A: Make the following verification checks if the COM port is not opened:

- In the Android platform (while the u-blox module is connected), use adbshell and verify that the `ttyACMX` or `ttyUSBX` ports are present in the `/dev` path. If these are not present, see appendix A.
- Make sure that the necessary permissions (`ttyACMx`, `ttyUSBx`) are added in the `uevent.platform.rc` file.
- Verify the SEPolicies for the required port, see appendix B.6.

**Q: All the scripts are not working or RIL is not working after integration in Android.**

A: It is recommended that the RIL package is downloaded and opened on the Linux machine. If the zip file is opened on a Windows machine, it may result in script corruption because of illegal characters (^M).

**Q: Data in PPP mode is not working after integrating the u-blox RIL.**

A: In PPP mode, check that the pppd_data service is running properly. If not, check the permissions and SEPolicy for this script. The pppd_data service starts the PPP daemon and sets many properties, so it should be verified that the necessary SEPolicies for pppd should be set.

Also verify that the "chat" script is added and running properly in pppd.

**Q: Data in RNDIS is not working after integrating u-blox RIL.**

A: usb0 is used to connect data in RNDIS mode so it should be verified that usb0 is properly enumerated by RNDIS kernel driver when the u-blox module is attached in RNDIS mode. The "netd_ena" service should also be running properly.

**Q: How to increase log buffer size?**

A: There is a setting in the "Log buffer sizes" developer option that can be set from 256 k to 16 M. If this setting is not present in the developer options, then an alternative way to change the buffer size is to set following property at the start of the "boot" section in the `init.rc` file.

```
setprop persist.logd.size 16777216
```

**Q: How do you go back to 4G as there is no option in the preferred network type?**

A: This can be done by dialing "`*#*#4636#*#*`" on an Android device and a menu would pop up. Select the phone information menu option and then it is possible to set the preferred network type from there.

**Q: How to get the MDN info displayed in Android?**

A: The MDN is saved in a special memory area by the service provider, with the name "ON": Own number phonebook (read/write) and the content can be shown by means of +CNUM. Sometimes it is written/saved/available by the SIM provider and sometimes not. So, the +CNUM AT command can read this number (if it is available).

**Q: Why `gsm0710muxd` is used?**

A: `gsm0710muxd` is used for serial communications with the GSM modem. It uses the AT+CMUX command to enable the multiplexing protocol control to make the various channels available to RIL. These channels are virtually mapped with the /pts channel of `gsm0710muxd` and connect RIL with the /pts channel.

**Q: How to check the properties set in RIL?**

A: The properties set in RIL can be checked in the adb shell using the "`getprop`" command, and any property can be set manually using the "`setprop`" command.

**Q: pppd exits with error code 17; how to resolve this error?**

A: This error happens when the ppp server starts up too slowly. When the ppp daemon sends the link configuring packets, they get bounced from the remote modem, thereby giving a serial loopback error. To resolve this, in the gprs script, `build/target/product/rootdir/etc/ppp/peers`, add the following line.

```
"lcp-max-configure 50"
```

**Q: How to set the default preferred network settings in Android?**

A: To set the default preferred network in Android, set the **`ro.telephony.default.network`** property in `device.mk` to the desired network settings. For example,

```
ro.telephony.default.network=10 /* LTE, CDMA, EvDo, GSM/WCDMA */
```

Recompile the source after setting the property and verifying the change from the build.prop file in the out directory.

**Q: How to enable adb logging using Wi-Fi in RIL?**

A: Add the following line to the boot section of the `init.rc` file, that is:

```
setprop service.adb.tcp.port 5555
```

**Q: How to enable the USB serial driver (for example, FTDI) in the Android kernel?**

A: It can be configured by enabling the USB serial driver in the USB section using the "make menuconfig" command. Alternatively, it can also be enabled by adding the following lines into the defconfig file of the respective hardware.

```
CONFIG_USB_SERIAL=y
CONFIG_USB_SERIAL_CONSOLE=y
CONFIG_USB_SERIAL_FTDI_SIO=y
CONFIG_USB_SERIAL_QUALCOMM=y
```

# N  Glossary

| Abbreviation | Definition |
|---|---|
| ADB | Android Debugger |
| AOSP | Android Open Source Project |
| API | Application Programming Interface |
| APN | Access Point Name |
| AVC | Access Vector Cache |
| CDC-ACM | Abstract Control Model |
| CID | Context identifier |
| ECM | Ethernet Control Model |
| EPS | Evolved Packet System |
| GSM | Global System for Mobile Communications |
| IP | Internet Protocol |
| LTE | Long Term Evolution |
| MCC | Mobile Country Code |
| MNC | Mobile Network Code |
| MNO | Mobile Network Operator |
| MUX | Multiplexer |
| OS | Operating System |
| PDN | Packet Data Network |
| PDP | Packet Data Protocol |
| PPP | Point-to-Point Protocol |
| QoS | Quality of Service |
| RIL | Radio Interface Layer |
| RNDIS | Remote Network Driver Interface Specification |
| SPI | Serial Peripheral Interface |
| TTY | Virtual interface for communication purposes (originally from TeleTYpewriter) |
| UART | Universal Asynchronous Receiver-Transmitter |
| USB | Universal Serial Bus |
| WCDMA | Wideband Code Division Multiple Access |

**Table 5: Explanation of the abbreviations and terms used**

# Related documentation

[1]    u-blox SPI interface application note, UBX-13001919
[2]    Ubuntu download: http://www.ubuntu.com/desktop/get-ubuntu/download
[3]    USB driver Android: https://dl-ssl.google.com/android/repository/usb_driver_r03-windows.zip
[4]    Android SDK: http://developer.android.com/sdk/index.html
[5]    u-blox multiplexer implementation application note, UBX-13001887
[6]    u-blox AT commands manual, UBX-13002752
[7]    SARA-R5 series system integration manual, UBX-19041356
[8]    SARA-R4 series system integration manual, UBX-16029218
[9]    Implementing SELinux: https://source.android.com/security/selinux/implement

☞    For regular updates to u-blox documentation and to receive product change notifications, register on our homepage (www.u-blox.com).

# Revision history

| Revision | Date | Name | Comments |
|---|---|---|---|
| - | 11-Jul-2011 | fpic | Initial release |
| 1 | 05-Sep-2011 | fpic | Document aligned to RIL delivery 01.023 |
| 2 | 23-Sep-2011 | lpah | Added chapter Appendix 1: Module connection |
| 3 | 26-Jan-2012 | lpah | Extended to include LISA-U2 series |
| 4 | 26-Mar-2012 | fpic | Extended to include Android 4.0 delivery |
| 5 | 21-Jun-2012 | fpic | Extended to include LEON-G100 and LEON-G200 series<br>Added compatibility matrix for Android software deliveries and supported interface by u-blox wireless modules |
| 6 | 20-Nov-2012 | fpic / lpah | Android 4.1 supported<br>(Last revision with docu number 3G.G2-CS-11003)1 |
| A | 30-Aug-2012 | fpic | Extended to include SARA-G350 series<br>Insert minor changes and support for 2G modules into Android 4.113 |
| R08 | 08-Apr-2014 | fpic | Android 4.2 and 4.3 supported; Added note on power off handling |
| R09 | 31-Jul-2014 | fpic | Android 4.4.x supported<br>Extended the document applicability to SARA-U2 and TOBY-L2 series |
| R10 | 24-Oct-2014 | fpic | Included modification for latest TOBY-L2 delivery |
| R11 | 11-Sep-2015 | yasi | PPP support for TOBY-L2; Android 4.4.x supported with MUX interface for SARA-G310; Android 5.x supported |
| R12 | 05-Feb-2016 | msin | Extended to RIL version 08.01 |
| R13 | 03-May-2016 | bqam / bahm | Extended to RIL version 09.00; Android 6.x supported<br>Firmware update over AT (FOAT) section added for Android 2.3 (SARA-U270-00S) |
| R14 | 31-Mar-2017 | bkha / fdil | Extended to RIL version 09.02. Extended the document applicability to TOBY-R2 and LARA-R2 series. Updated TOBY-L2 series profile setting section. |
| R15 | 28-Jul-2017 | bahm / yasi | Android 7.x supported. Added section about Verizon network settings. |
| R16 | 30-Aug-2017 | uafz | Added section about module specific modifications |
| R17 | 10-Oct-2017 | bkha | Updated the init.rc configuration |
| R18 | 27-Nov-2017 | bkha | Updated multi module and OEM Hook App sections |
| R19 | 17-Jan-2018 | bkha | Android 8.x Oreo support and init.ublox.rc, Core_ublox.rc file addition. |
| R20 | 16-May-2018 | bkha | Extended document applicability to TOBY-L4 series. |
| R21 | 24-Oct-2018 | bkha / fdil | Added section regarding the integration on i.MX 6 platform. |
| R22 | 21-Dec-2018 | iazi | Added sections regarding Android 9.x Pie – New supporting Android version |
| R23 | 15-Apr-2018 | bkha | Added support for RIL v12.01 |
| R24 | 06-Sep-2019 | bkha | i.MX 8 Support, remove android 4.0 section, RILv12.03 |
| R25 | 08-Jan-2020 | bkha / fdil | Added support for Android 10.x. Extended the document applicability to SARA-R5 series. Inserted improvements in the handling of SARA-R4 features. |
| R26 | 19-May-2020 | bkha / fdil | Added new repository configurations for TOBY-L2/SARA-R5, MNO profile settings, RIL logging update, i.MX section improved |
| R27 | 27-Jul-2020 | uafz / fdil | Added new repository configurations, updated Multimode and SARA-R5 sections |
| R28 | 16-Feb-2021 | uafz/bkha/fdil | Added support for Android 11.x and removed Android 4.x support.<br>Added licenses headers and conditions. |
| R29 | 29-Jul-2021 | uafz | Added new repository settings, updated licenses and RIL debug sections. |

# Contact

For complete contact information, visit us at www.u-blox.com.

**u-blox Offices**

**North, Central and South America**

**u-blox America, Inc.**

Phone:    +1 703 483 3180
Email:    info_us@u-blox.com

**Regional Office West Coast:**

Phone:    +1 408 573 3640
Email:    info_us@u-blox.com

**Technical Support:**

Phone:    +1 703 483 3185
Email:    support@u-blox.com

**Headquarters
Europe, Middle East, Africa**

**u-blox AG**

Phone:    +41 44 722 74 44
Email:    info@u-blox.com
Support: support@u-blox.com

**Asia, Australia, Pacific**

**u-blox Singapore Pte. Ltd.**

Phone:    +65 6734 3811
Email:    info_ap@u-blox.com
Support: support_ap@u-blox.com

**Regional Office Australia:**

Phone:    +61 3 9566 7255
Email:    info_anz@u-blox.com
Support: support_ap@u-blox.com

**Regional Office China (Beijing):**

Phone:    +86 10 68 133 545
Email:    info_cn@u-blox.com
Support: support_cn@u-blox.com

**Regional Office China (Chongqing):**

Phone:    +86 23 6815 1588
Email:    info_cn@u-blox.com
Support: support_cn@u-blox.com

**Regional Office China (Shanghai):**

Phone:    +86 21 6090 4832
Email:    info_cn@u-blox.com
Support: support_cn@u-blox.com

**Regional Office China (Shenzhen):**

Phone:    +86 755 8627 1083
Email:    info_cn@u-blox.com
Support: support_cn@u-blox.com

**Regional Office India:**

Phone:    +91 80 405 092 00
Email:    info_in@u-blox.com
Support: support_in@u-blox.com

**Regional Office Japan (Osaka):**

Phone:    +81 6 6941 3660
Email:    info_jp@u-blox.com
Support: support_jp@u-blox.com

**Regional Office Japan (Tokyo):**

Phone:    +81 3 5775 3850
Email:    info_jp@u-blox.com
Support: support_jp@u-blox.com

**Regional Office Korea:**

Phone:    +82 2 542 0861
Email:    info_kr@u-blox.com
Support: support_kr@u-blox.com

**Regional Office Taiwan:**

Phone:    +886 2 2657 1090
Email:    info_tw@u-blox.com
Support: support_tw@u-blox.com