HUAWEI Module

# Android RIL Integration Guide

| | |
|---|---|
| Version | 3.4 |
| Date | 2014-04-26 |

## Copyright © Huawei Technologies Co., Ltd. 2014. All rights reserved.

No part of this manual may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd. and its affiliates ("Huawei").

The product described in this manual may include copyrighted software of Huawei and possible licensors. Customers shall not in any manner reproduce, distribute, modify, decompile, disassemble, decrypt, extract, reverse engineer, lease, assign, or sublicense the said software, unless such restrictions are prohibited by applicable laws or such actions are approved by respective copyright holders.

## Trademarks and Permissions

HUAWEI, HUAWEI, and are trademarks or registered trademarks of Huawei Technologies Co., Ltd.

LTE is a trade mark of ETSI.

Other trademarks, product, service and company names mentioned may be the property of their respective owners.

## Notice

Some features of the product and its accessories described herein rely on the software installed, capacities and settings of local network, and therefore may not be activated or may be limited by local network operators or network service providers.

Thus, the descriptions herein may not exactly match the product or its accessories which you purchase.

Huawei reserves the right to change or modify any information or specifications contained in this manual without prior notice and without any liability.

## DISCLAIMER

ALL CONTENTS OF THIS MANUAL ARE PROVIDED “AS IS”. EXCEPT AS REQUIRED BY APPLICABLE LAWS, NO WARRANTIES OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE MADE IN RELATION TO THE ACCURACY, RELIABILITY OR CONTENTS OF THIS MANUAL.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL HUAWEI BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT, OR CONSEQUENTIAL DAMAGES, OR LOSS OF PROFITS, BUSINESS, REVENUE, DATA, GOODWILL SAVINGS OR ANTICIPATED SAVINGS REGARDLESS OF WHETHER SUCH LOSSES ARE FORSEEABLE OR NOT.

THE MAXIMUM LIABILITY (THIS LIMITATION SHALL NOT APPLY TO LIABILITY FOR PERSONAL INJURY TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH A LIMITATION) OF HUAWEI ARISING FROM THE USE OF THE PRODUCT DESCRIBED IN THIS MANUAL SHALL BE LIMITED TO THE AMOUNT PAID BY CUSTOMERS FOR THE PURCHASE OF THIS PRODUCT.

## Import and Export Regulations

Customers shall comply with all applicable export or import laws and regulations and be responsible to obtain all necessary governmental permits and licenses in order to export, re-export or import the product mentioned in this manual including the software and technical data therein.

## Privacy Policy

To better understand how we protect your personal information, please see the privacy policy at http://consumer.huawei.com/privacy-policy.

# About This Document

## Revision History

| Document Version | Date | Chapter | Description |
|---|---|---|---|
| 1.0 | 2011-06-23 | | Created |
| 2.0 | 2011-09-09 | 3 | Added guide for TD-SCDMA |
| | | 4.2 | Upgraded guide for UpdateWizard V100R005. |
| | | 4.4 | Added GPS guide for Android 2.3. |
| 2.1 | 2011-12-30 | 3.2 | Added 3.2.7, usage of OEM-HOOK. |
| 2.2 | 2012-02-03 | 2.3 | Added support for Android 4.0 |
| | | 3.2 | Modified 3.2.6 for CDMA2000 SMS |
| | | 4.4 | Added GPS guide for Android 3.x, 4.0 |
| 2.3 | 2012-02-10 | 4.4 | Added interfaces supported by GPS for Android 3.x, 4.0 |
| 2.4 | 2012-03-13 | 3.3 | Deleted the description on how to integrate drivers |
| 2.5 | 2012-03-13 | 4.4 | modified the description on how to integrate GPS |
| 2.6 | 2012-05-02 | 4.4 | Added XTRA server in gps.conf |
| | | 3.2 | Added XTRA server in gps.conf |
| 2.7 | 2012-07-20 | 4.4.5 | Deleted SONY GPS customization |
| 2.8 | 2013-01-14 | 3.2 | Modified the document to be sure that it can be used for Huawei Module and data card |

| Document Version | Date | Chapter | Description |
|---|---|---|---|
| 2.9 | 2013-01-18 | 4.5.5 | Added a description of how to Launch gps.default.so |
| 3.0 | 2013-02-08 | 2.4 | Added support for MU609 |
| | | 4.3 | Added support for MU609 |
| 3.1 | 2013-05-30 | 3.2.2 | Added ports instructions |
| 3.2 | 2013-10-12 | 2.4 | Added GPS support for MU736 |
| | | 3.2.8 | Added support for LTE |
| 3.3 | 2014-01-08 | 2.2 | Added Support for GPS+GLONASS Position |
| | | 2.4 | Added support for ME906E |
| | | 4.5.5 | Modified Android 2.3 System for GPS and deleted GPS Integration for Android 2.2 |
| 3.4 | 2014-04-26 | Copyright | Updated Copyright description |
| | | 2.4 | Deleted Supported Products |

# Contents

# 1 Overview

## 1.1 About This Chapter

This chapter describes the short introduction and deliverables of the Huawei RIL.

- Introduction to the Huawei RIL
- Huawei RIL Deliverables

## 1.2 Introduction to the Huawei RIL

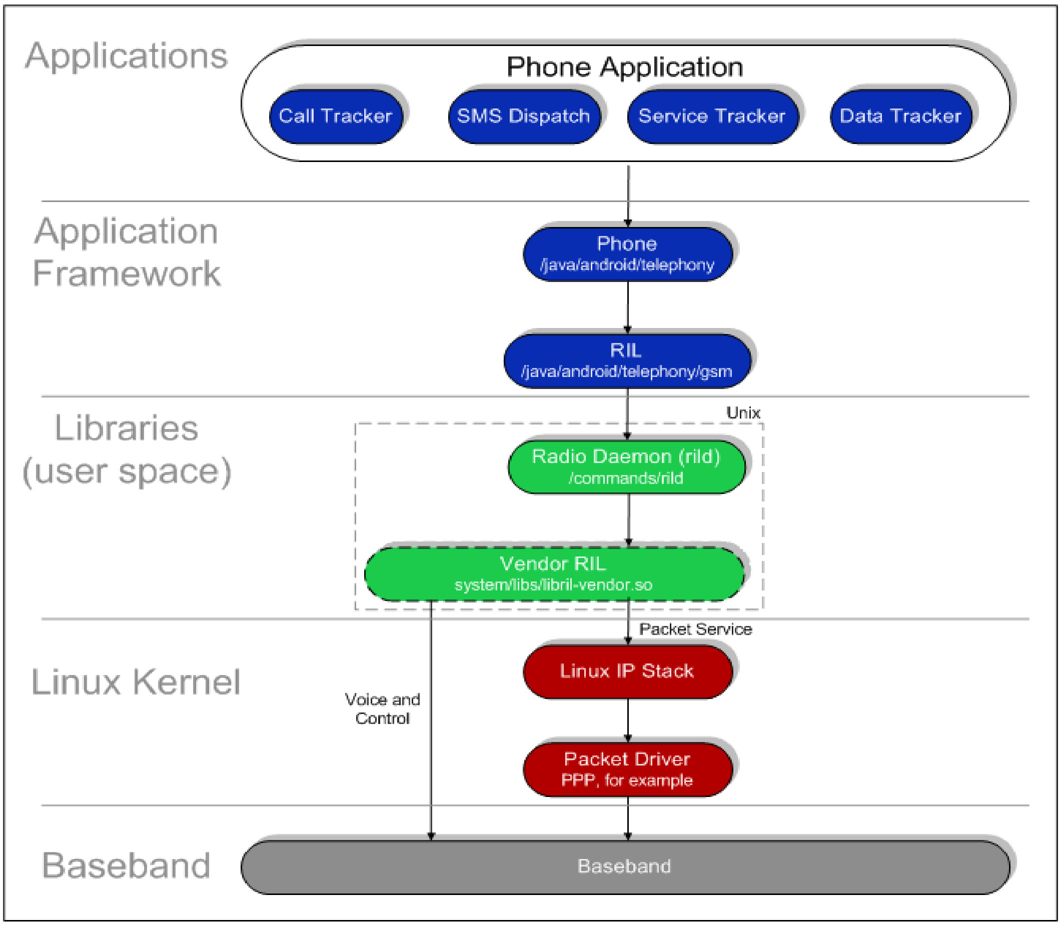
RIL is short for Radio Interface Layer.

When an Android operating system (OS) works with a module/modem to provide functions such as data, voice and Global Positioning System (GPS) services, the Android OS must communicate with the module/modem. The RIL is used to implement the communication between an Android OS and a module/modem.

To enable Android to support different communications methods (such as using AT commands or QMI commands), Google divides the RIL into two components: an RIL Daemon (RILD) and a Vendor RIL. The RILD monitors the requests (for example, a request for making a call) from the upper layers of Android, and invokes the interfaces implemented at the Vendor RIL to send the requests to the module/modem. Then the module/modem handles the requests. For Huawei modules, this process is to convert the requests into corresponding AT commands and send the commands to the modules.

The RILD has been designed by Google and does not require any modification. The Vendor RIL is designed by vendors, and it is the Huawei RIL in this document.

**Figure 1-1** RIL architecture



## 1.3 Huawei RIL Deliverables

Huawei RIL deliverables are:

- RIL library file: **libhuawei-ril.so**.

  This is the Vendor RIL.
- ip-up script

  When a Point-to-Point Protocol (PPP) link is established successfully, this script is executed to set the **ip**, **dns** and **gw** parameters.
- ip-down script

  When a PPP link failed to be established, this script is executed to clear the **ip**, **dns** and **gw** parameters.
- Release notes

  The version information of the Huawei RIL.

# 2 What Are Supported

## 2.1 About This Chapter

This chapter describes what are supported by the Huawei RIL, including:

- Supported Functions
- Supported Android Versions

## 2.2 Supported Functions

The Huawei RIL supports the following:

- Basic device and network management service for WCDMA, CDMA2000 and TD-SCDMA;
- Data service for WCDMA, CDMA2000 and TD-SCDMA;
- SMS service for WCDMA, CDMA2000 and TD-SCDMA;
- Basic PCM voice service for WCDMA, CDMA2000 and TD-SCDMA;
- Phonebook service for WCDMA and TD-SCDMA;
- Voice supplementary services for WCDMA and CDMA;
- USSD service for WCDMA;
- Support for GPS + GLONASS two positioning system positioning;

## 2.3 Supported Android Versions

At present, the Huawei RIL supports the following versions of Android:

- Android 2.x
- Android 3.x
- Android 4.x

# 3 Integration

## 3.1 About This Chapter

This chapter describes the integration of Huawei RIL.

## 3.2 Integrating the Huawei RIL

The following integration procedure is based on a platform of Nvidia Tegra 2 + Android 2.2. Integration on other platforms may be slightly different.

### 3.2.1 Launching libhuawei-ril.so

Modifying the initialization script (system/core/rootdir/init.rc):

service ril-daemon /system/bin/rild **-l libhuawei-ril.so**

Put **libhuawei-ril.so** into **/system/lib**

### 3.2.2 Modifying the Rights of Device Ports

Please make sure the device ports, **PCUI and MODEM**, possess readable, writable, and executive rights.

For example, add commands,

chmod 777 /dev/ttyUSB0

chmod 777 /dev/ttyUSB2

or,

chmod 777 /dev/ttyACM0

chmod 777 /dev/ttyACM1

in the initialization script **rootdir/init.rc**.

NOTE

- Usually, for **TD-SCDMA** module, **/dev/ttyACM0** is modem port, while **/dev/ttyACM1** is PCUI port.
- In addition, the modem/pcui/diag/ port names depend on the modules instructions of which you can refer to.

## 3.2.3 Preparing ip-up and ip-down Scripts

Copy **ip-up** and **ip-down** scripts to directory **/etc/ppp**.

NOTE

Scripts ip-up and ip-down need readable, writable and executive rights.

For example, add commands,

chmod 777 /etc/ppp/ip-up

chmod 777 /etc/ppp/ip-down

in the initialization script **rootdir/init.rc**.

## 3.2.4 Modifying the Right of RILD

RILD needs executive right.

For example, comment out the function switchUser() in the file **/hardware/ril/rild/rild.c**:

// switchUser();

## 3.2.5 Notification for CDMA2000 Support on Android 2.3

To support CDMA2000 module in Android 2.3, the preferred network mode should be changed to "**4**", preferred_network_mode of WCDMA and TD-SCDMA module is "**0**". For example:

Using adb shell to execute the following command:

sqlite3 /data/data/com.android.providers.settings/databases/settings.db

update secure set value='4' where name='preferred_network_mode';

.quit

Now, the system can use the CDMA module.

## 3.2.6 Notifications for CDMA2000 SMS Support

1. Modify the following functions on the file: {android_root}/frameworks/base/telephony/java/com/android/internal/telephony/cdma/SmsMessage.java

(1) **privateGetSubmitPdu()**:modify ***“int teleservice = bearerData.hasUserDataHeader ? SmsEnvelope.TELESERVICE_WEMT : SmsEnvelope.TELESERVICE_WMT;”*** to ***“int teleservice = SmsEnvelope.TELESERVICE_WMT;”***

(2) **createFromEfRecord()**:modify ***"int size = data[1];"*** to ***"int size = data[1] & 0xff;"***

(3) **parsePduFromEfRecord()**:modify ***"int parameterLen = dis.readByte();"*** to ***"int parameterLen = dis.readByte() & 0xff;"***

2. Modify the following function on the file:

{android_root}/frameworks/base/telephony/java/com/android/internal/telephony/cdma/sms/BearderData.java

(1) **decode7bitAscii()**:modify ***"int wantedBits = (offset * 8) + (numFields * 7);"*** to

```
"if (48 == offset) {
            offset++;
            numFields = numFields - 7;
  }
 int wantedBits = offset + (numFields * 7);"
```

## 3.2.7 Sending AT Commands Through OEM-HOOK-STRING

RIL OEM-HOOK interfaces allow users to send AT commands to module/modem directly. Currently, Huawei RIL only implements the interface OEM-HOOK-STRING. Following example shows how to send AT command from framework directly.

Send command "AT^SWSPATH=2" to switch the module to PCM mode when SIM is ready.

File: GsmServiceStateTracker.java or CdmaServiceStateTracker.java,

Function: public void handleMessage (Message msg),

Sample:

case EVENT_SIM_READY:

**phone.invokeOemRilRequestStrings(new String[]{ "AT^SWSPATH=2" }, mHandler.obtainMessage(EVENT_OEM_RIL_MESSAGE));**

## 3.2.8 Using LTE Only Option

There is no "LTE Only Option" on Android UI. So if you want the module/modem to use LTE, you must to modify Android codes.

# 4 Other Deliverables

## 4.1 About This Chapter

This chapter describes other deliverables, including:

- UpdateWizard For EM820W/MC509/MU509
- UpdateWizard For MU733/MU739/MU609T/MU609
- QXDMlog Tool
- GPS

## 4.2 UpdateWizard For EM820W/MC509/MU509

### 4.2.1 Deliverables

Huawei UpdateWizard tool is running on Android to upgrade firmware.

Deliverables:

- Upgrade tool: UpdateWizard
- Configurable file: *StringList.strings*
- Firmware will be released as an **.bin* file, whose file name is in the form of *ModuleName_VersionNumber.bin.*

### 4.2.2 Procedures

Step 1 Change the firmware name from ModuleName_VersionNumber.bin to **MOBILE_CONNECT.BIN**.

Step 2 Modify StringList.strings.

- The "**IDS_UPD_VERSION**" and "**IDS_PRODUCE_ID**" should be the *ModuleName*.
- The "**IDS_PRODUCE_Version**" should be the *VersionNumber.*

Step 3 Connect to Android system, and create a new directory named **update**. (the name can be any as you like).

Step 4 Use **adb push** command to push upgrade tool and upgrade package into the directory **update/**.

NOTE

***UpdateWizard***, ***MOBILE_CONNECT.BIN***, and ***StringList.strings*** should be under the same directory.

Step 5 Enter the directory **update/**.

Step 6 Modify rights of UpdateWizard: **chmod 777 *UpdateWizard***.

Step 7 Terminate ril: **stop ril-daemon.**

Step 8 If the upgrade requires a password, run **./UpdateWizard ./** *XXXXXXXX*. If it does not require a password, run **./UpdateWizard ./**.

NOTE

- XXXXXXXX: an 8-digit password.
- The executable file name and the password in the command must be separated with a space.

Step 9 Restart ril: start ril-daemon.

### 4.2.3 Requirements

The Android where the UpdateWizard tool is running must support plug and play.

## 4.3 UpdateWizard For MU733/MU739/MU609T/MU609

### 4.3.1 Deliverables

Huawei UpdateWizard tool is running on Android to upgrade firmware.

Deliverables:

- Upgrade tool: UpdateWizard
- Firmware will be released as an **.bin* file, whose file name is in the form of *ModuleName_VersionNumber.bin.*

### 4.3.2 Procedures

Step 1 Copy **android-ndk-r8b-linux-x86.tar.tar** to any folder in linux, and unpack it.

Step 2 Copy **UpdateWizard3.1\code\sdk\build\Android\jni.zip** to any folder in linux, and unpack it.

Step 3 Copy folder build, include, and src in UpdateWizard3.1\Tool-Android_Update **-WizardV300R002B001\code\sdk** to folder jni in linux, and merge the original folder **build, include, and src** in **jni**.

**NOTE**

"merge" must be used rather than “replace”.

Step 4 In the directory of **jni** in linux, open the terminal and entry the command: **android-ndk-r8b/ndk-build** to generate the **UpdateWizard** tool.

Step 5 Connect to Android system, and create a new directory named **update** (the name can be any as you like).

Step 6 Use adb push command to push upgrade tool and upgrade package into the directory **update/**.

**NOTE**

**UpdateWizard** and **ModuleName_VersionNumber.bin** should be under the same directory.

Step 7 Enter the directory **update/**.

Step 8 Modify rights of UpdateWizard: **chmod 777 UpdateWizard**.

Step 9 Terminate ril: **stop ril-daemon.**

Step 10 If the upgrade requires a password, run **./UpdateWizard ./** *XXXXXXXX*. If it does not require a password, run **./UpdateWizard ./ ModuleName_VersionNumber.bin**.

**NOTE**

- XXXXXXXX: an 8-digit password.
- The executable file name and the password in the command must be separated with a space.

Step 11 Restart ril: **start ril-daemon**.

### 4.3.3 Requirements

The Android where the UpdateWizard tool is running must support plug and play.

## 4.4 QXDMlog Tool

The QXDMLog tool is used to obtain the logs of Huawei modules in Android.

### 4.4.1 Deliverable

QXDMLog tool

**NOTE**

The Huawei QXDMLog tool is named CTOOL-ANDROID, which can obtain up to 40 MB log information at a time. If the log information is larger than 40 MB, it will be saved to the SD card on the module. If the SD card does not exist, the process of obtaining the log information is terminated.

### 4.4.2 Procedures for Obtaining a Huawei Module Log

Step 1 Copy the QXDMLog tool to any folder except root directories. For example, to copy the QXDMLog tool to **/data/**, run **adb push CTOOL-ANDROID /data/**.

Step 2 Run **chmod 777 /data/CTOOL-ANDROID**.

Step 3 Run **cd /data.**

Step 4 Run **./CTOOL-ANDROID** to obtain the log. A log file named **HWQXDMLog** will be created in the same directory where the tool is stored.

Step 5 Press **Ctrl+C** to stop the tool.

Step 6 To copy the **HWQXDMLog** file to the PC, run **adb pull /data/HWQXDMLog**.

## 4.5 GPS

### 4.5.1 Supported GPS Functions

Supported GPS functions are:

- Standalone GPS and Assisted GPS (AGPS) for WCDMA
- Standalone GPS for CDMA
- Standalone GPS and XTRA GPS for WCDMA

### 4.5.2 Interfaces Supported by the GPS Function

1. GPS Interface

```
static const GpsInterface   HwGpsInterface = {
          sizeof(GpsInterface),//for android 2.3 or later version
          hw_gps_init,
          hw_gps_start,
          hw_gps_stop,
          hw_gps_cleanup,
          hw_gps_inject_time,
          hw_gps_inject_location,
          hw_gps_delete_aiding_data,
          hw_gps_set_position_mode,
          hw_gps_get_extension,
};
```

2. AGPS Interface

```
static const AGpsInterface    HwAGpsInterface = {
    sizeof(AGpsInterface), //for android 2.3 or later version
    hw_agps_init,
    hw_agps_data_conn_open,
    hw_agps_data_conn_closed,
    hw_agps_data_conn_failed,
    hw_agps_set_server,
};
```

3. XTRA inferface

```
static const GpsXtraInterface hwXTRAInterface =
{
    sizeof(GpsXtraInterface),
    hw_xtra_init,
    hw_inject_xtra_data,
};
```

### 4.5.3 Android Versions Supported by the GPS Function

The GPS function supports Android 2.3, 3.x and 4.x.

### 4.5.4 GPS Deliverable

On Android 2.3, 3.x and 4.x, GPS deliverable is **/system/lib/hw/gps.default.so**.

### 4.5.5 GPS Integration

#### Launching gps.default.so

1. Delete the gps.xxx.so file under android directory /system/lib/hw/.
2. Push the gps.default.so file into the directory /system/lib/hw/.
3. Push the updated gps.conf file into the directory /system/etc/.

#### For Android 2.3

The following integration procedure is based on a platform of Freescale I.MX51 + Android 2.3. Integration on other platforms may be slightly different.

1. Add JNI layer GPS init code for Android 2.3.

   File to be modified:

mydroid/frameworks/base/services/jni/com_android_server_location_GpsLocationProvider.cpp

Function to be modified:

Function :android_location_GpsLocationProvider_init

```
const GpsInterface* interface = GetGpsInterface(env, obj);
if (!interface)
    return false;
if (NULL != interface->init && 0 != interface->init(&sGpsCallbacks)) {
    LOGD("GpsInterface init failed!");
    return false;
}
sAGpsInterface = (const AGpsInterface*) interface-
    >get_extension(AGPS_INTERFACE);
 if (sAGpsInterface)
    sAGpsInterface->init(&sAGpsCallbacks);
```

The bolder code above should be added in function android_location_GpsLocationProvider_init.

Function to be modified:

GetGpsInterface

```
static const GpsInterface* GetGpsInterface(JNIEnv* env, jobject obj) {
    // this must be set before calling into the HAL library
    if (!mCallbacksObj)
        mCallbacksObj = env->NewGlobalRef(obj);

    if (!sGpsInterface) {
        sGpsInterface = get_gps_interface();
        if (!sGpsInterface) {
            sGpsInterface = NULL;
            return NULL;
        }
    }
    return sGpsInterface;
}
```

2. Modify the access right of the GPS data and control ports.

   Modify etc file:

   **mydroid/system/core/rootdir/etc/ueventd.freescale.rc**

   modify ttyUSB* access permission as below:

   ```
   /dev/ttyUSB*        0777   radio    radio
   ```

**NOTE**

This is for freescale i.mx51 board. For another platform, it may be slightly different.

3. Create a gps.conf file.

   In **mydroid/out/target/product/imx51_bbg/system/etc**, add a **gps.conf** file containing the following contents:

   ```
   NTP_SERVER=asia.pool.ntp.org
   SUPL_HOST=http://supl.google.com
   SUPL_PORT=7276
   XTRA_SERVER_1=http://xtra1.gpsonextra.net/xtra.bin
   XTRA_SERVER_2=http://xtra2.gpsonextra.net/xtra.bin
   XTRA_SERVER_3=http://xtra3.gpsonextra.net/xtra.bin
   ```

**NOTE**

- SUPL_HOST is an AGPS server which is carrier specified.
- XTRA_SERVER is an XTRA server which is carrier specified.

4. Compile the Android system again and flash the re-compiled Android image onto the Android tablet.

## For Android 3.x

1. Create a gps.conf file.

   In **mydroid/out/target/product/xxx/system/etc**, add a **gps.conf** file containing the following contents:

   ```
   NTP_SERVER=asia.pool.ntp.org
   SUPL_HOST=http://supl.google.com
   SUPL_PORT=7276
   XTRA_SERVER_1=http://xtra1.gpsonextra.net/xtra.bin
   XTRA_SERVER_2=http://xtra2.gpsonextra.net/xtra.bin
   XTRA_SERVER_3=http://xtra3.gpsonextra.net/xtra.bin
   ```

**NOTE**

- SUPL_HOST is an AGPS server which is carrier specified.
- XTRA_SERVER is an XTRA server which is carrier specified.

2. Compile the Android system again and flash the re-compiled Android image onto the Android tablet.

### For Android 4.x

The following integration procedure is based on a platform of pandaboard. Integration on other platforms may be slightly different.

1. Create a gps.conf file.

   In **mydroid/out/target/product/pandaboard/system/etc**, add a **gps.conf** file containing the following contents:

   ```
   NTP_SERVER=asia.pool.ntp.org
   SUPL_HOST=http://supl.google.com
   SUPL_PORT=7276
   XTRA_SERVER_1=http://xtra1.gpsonextra.net/xtra.bin
   XTRA_SERVER_2=http://xtra2.gpsonextra.net/xtra.bin
   XTRA_SERVER_3=http://xtra3.gpsonextra.net/xtra.bin
   ```

**NOTE**

- SUPL_HOST is an AGPS server which is carrier specified.
- XTRA_SERVER is an XTRA server which is carrier specified.

2. Compile the Android system again and flash the re-compiled Android image onto the Android tablet.

# 5 Abbreviations

| Abbreviations | Full spelling |
|---|---|
| AGPS | Assisted Global Positioning System |
| CDMA | Code Division Multiple Access |
| GPS | Global Positioning System |
| NMEA | National Marine Electronics Association |
| RIL | Radio Interface Layer |
| TD-SCDMA | Time Division-Synchronous Code Division Multiple Access |
| WCDMA | Wideband CDMA |
| XTRA GPS | Xtra Global Position System |