Software Integration and Development Guide

For Linux USB Platforms

SIERRA WIRELESS®

41114212
3.06
October 29, 2021

Important Notice

Due to the nature of wireless communications, transmission and reception of data can never be guaranteed. Data may be delayed, corrupted (i.e., have errors) or be totally lost. Although significant delays or losses of data are rare when wireless devices such as the Sierra Wireless modem are used in a normal manner with a well-constructed network, the Sierra Wireless modem should not be used in situations where failure to transmit or receive data could result in damage of any kind to the user or any other party, including but not limited to personal injury, death, or loss of property. Sierra Wireless accepts no responsibility for damages of any kind resulting from delays or errors in data transmitted or received using the Sierra Wireless modem, or for failure of the Sierra Wireless modem to transmit or receive such data.

Safety and Hazards

Do not operate the Sierra Wireless modem in areas where cellular modems are not advised without proper device certifications. These areas include environments where cellular radio can interfere such as explosive atmospheres, medical equipment, or any other equipment which may be susceptible to any form of radio interference. The Sierra Wireless modem can transmit signals that could interfere with this equipment. Do not operate the Sierra Wireless modem in any aircraft, whether the aircraft is on the ground or in flight. In aircraft, the Sierra Wireless modem **MUST BE POWERED OFF**. When operating, the Sierra Wireless modem can transmit signals that could interfere with various onboard systems.

Note: Some airlines may permit the use of cellular phones while the aircraft is on the ground and the door is open. Sierra Wireless modems may be used at this time.

The driver or operator of any vehicle should not operate the Sierra Wireless modem while in control of a vehicle. Doing so will detract from the driver or operator's control and operation of that vehicle. In some states and provinces, operating such communications devices while in control of a vehicle is an offence.

Limitations of Liability

This manual is provided "as is". Sierra Wireless makes no warranties of any kind, either expressed or implied, including any implied warranties of merchantability, fitness for a particular purpose, or noninfringement. The recipient of the manual shall endorse all risks arising from its use.

The information in this manual is subject to change without notice and does not represent a commitment on the part of Sierra Wireless. SIERRA WIRELESS AND ITS AFFILIATES SPECIFICALLY DISCLAIM LIABILITY FOR ANY AND ALL DIRECT, INDIRECT, SPECIAL, GENERAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES INCLUDING, BUT NOT LIMITED TO, LOSS OF PROFITS OR REVENUE OR ANTICIPATED PROFITS OR REVENUE ARISING OUT OF THE USE OR INABILITY TO USE ANY SIERRA WIRELESS PRODUCT, EVEN IF SIERRA WIRELESS AND/OR ITS AFFILIATES HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR THEY ARE FORESEEABLE OR FOR CLAIMS BY ANY THIRD PARTY.

Notwithstanding the foregoing, in no event shall Sierra Wireless and/or its affiliates aggregate liability arising under or in connection with the Sierra Wireless product, regardless of the number of events, occurrences, or claims giving rise to liability, be in excess of the price paid by the purchaser for the Sierra Wireless product.

Customer understands that Sierra Wireless is not providing cellular or GPS (including A-GPS) services. These services are provided by a third party and should be purchased directly by the Customer.

SPECIFIC DISCLAIMERS OF LIABILITY: CUSTOMER RECOGNIZES AND ACKNOWLEDGES SIERRA WIRELESS IS NOT RESPONSIBLE FOR AND SHALL NOT BE HELD LIABLE FOR ANY DEFECT OR DEFICIENCY OF ANY KIND OF CELLULAR OR GPS (INCLUDING A-GPS) SERVICES.

Patents

This product may contain technology developed by or for Sierra Wireless Inc.

This product includes technology licensed from QUALCOMM®.

This product is manufactured or sold by Sierra Wireless Inc. or its affiliates under one or more patents licensed from InterDigital Group and MMP Portfolio Licensing.

Copyright

Trademarks

Sierra Wireless®, AirPrime®, AirLink®, AirVantage®, WISMO® and the Sierra Wireless and Open AT logos are registered trademarks of Sierra Wireless, Inc. or one of its subsidiaries.

Watcher® is a registered trademark of NETGEAR, Inc., used under license.

Windows® is a registered trademark of Microsoft Corporation.

Macintosh® and Mac OS X® are registered trademarks of Apple Inc., registered in the U.S. and other countries.

QUALCOMM® is a registered trademark of QUALCOMM Incorporated. Used under license.

Other trademarks are the property of their respective owners.

Contact Information

Sales Desk:	Phone:	1-604-232-1488
	Hours:	8:00 AM to 5:00 PM Pacific Time
	Contact:	http://www.sierrawireless.com/sales
Post:	Sierra Wireless 13811 Wireless Way Richmond, BC Canada V6V 3A4	
Technical Support:	support@sierrawireless.com	
RMA Support:	repairs@sierrawireless.com	
Fax:	1-604-231-1109	
Web:	http://www.sierrawireless.com/	

Consult our website for up-to-date product descriptions, documentation, application notes, firmware upgrades, troubleshooting tips, and press releases: www.sierrawireless.com

Document History

Version	Date	Updates
3.06	October 29, 2021	MBPL R24.0.21, Build 5182
3.05.0	March 19, 2021	Initial Release, R18 Build 5146

Table of Content

1. Linux Host Software

Sierra Wireless Mobile Broadband Package for Linux platforms utilizing Qualcomm based USB embedded modules is a complete software suite that includes:

- Device Drivers
  - Complete source code
  - Binaries for supporting platforms and Linux kernels
  - Readme.txt (detailed instructions for building, using and debugging drivers)
- SDK
  - Libraries
    - Binaries for supporting platforms
      - lite-qmi
      - lite-qmux
      - lite-mbim
      - lite-fw
    - SDK Documentation
      - Doxygen-generated documentation (APIs, structures, etc.)
    - Sample Apps
      - Sample apps support USB and PCIe host interfaces
      - Sample apps have no glib dependencies
      - The list of sample apps is subject to change with every MBPL release
      - For details, refer to MBPL Release Notes documents
- Tools
  - Firmware Logging and troubleshooting Utilities
  - Complete source code
- Documentation
  - Developer's Guide (this document)
  - Release Notes and additional instructions

1.1. Supported Host Interfaces and Products

Sierra Wireless modules using Qualcomm chipsets support the following physical host interfaces:

- USB
  - USB3.1 (SDX55 and newer)
  - USB3.0 (9x50, 9x30)
  - USB2 (9x07, 9x15)
  - Configurable USB Composite Interface

All Sierra Wireless products using the above host interfaces are supported. Supported products are referenced here based on the Qualcomm chipset used:

- SDX55 and newer
- 9x50, 9x30
- 9x07, 9x15

1.2. Supported Platforms and Processors

The following is a summary of supported processors and platforms:

- ARM (64-bit)
  - RockPro64
  - NXP (LS1043, LS1046)
- ARM (32-bit)
- MIPS
- Intel (64-bit/32-bit)
  - Intel and AMD based desktops with USB ports

1.3. Supported Kernel Versions

MBPL supports open source kernel versions 4.4 and newer.

The package includes binaries for several (but not all) kernel versions

.

2. Device Drivers

2.1. USB Interface

Sierra Wireless modules expose a configurable USB composite interface.

The available logical USB interfaces are:

- MBIM – Standard interface supported by open source device drivers
- RmNet – Qualcomm proprietary network interface
  - Note: MBIM and RmNet interfaces are mutually exclusive
- Serial – Serial interfaces supported by open source device drivers:
  - AT (AT Command port)
  - DM (DM port (Diagnostics))
  - QDLoader (Note – This interface is exposed only during FW Download operations.)

2.1.1. Host Software Architecture Diagram (USB-MBIM)

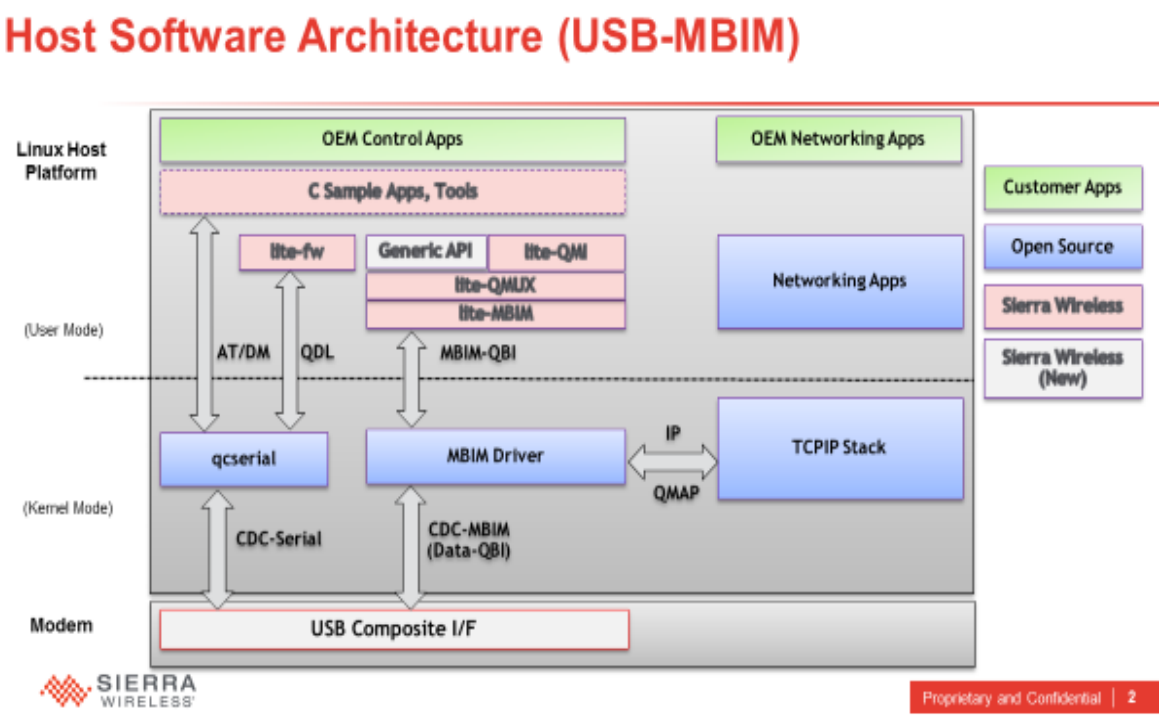

Figure 1. Host Software Architecture – USB, MBIM Interfaces

2.1.2. USB VID/PID/Interface Assignment

Table 1 lists the USB Vendor ID (VID) and the USB Interface IDs used by Sierra Wireless modules. PIDs are product category and SKU specific.

Table 1. USB VID/PID and Supported Interfaces (SDX55 based modules)

Mode	VID	PID	USB Interface ID	Interface Type
Application	0x1199	Device (SKU) Specific	0x00	MBIM network interface
			0x01	
			0x02	GNSS (not supported on Linux)
			0x03	AT Command Port
			0x04	DM Port
Boot	0x1199	Device (SKU) Specific	0x00	QDLoader Port

Table 2. USB VID/PID and Supported Interfaces (9x50, 9x30 based modules)

Mode	VID	PID	USB Interface ID	Interface Type
Application	0x1199	Device (SKU) Specific	0x00	DM Port
			0x02	NMEA
			0x03	AT Command Port
			0x0c	MBIM network interface
			0x0d	
Boot	0x1199	Device (SKU) Specific	0x00	QDLoader Port

2.1.3. Device Drivers

The USB drivers are based on open source drivers.

Minor changes are required to support the Sierra Wireless USB VID/PID and the USB composition/interface mapping.

The provided source code includes additional fixes, improvements and build instructions.

- MBIM Interface: open source MBIM driver (cdc_mbim)
  - This driver loads automatically as it detects the USB MBIM class interface
- RmNet Interface: open source (qmi_wwan)
- Serial Interfaces: open source serial driver (qcserial)
  - /dev/ttyUSB0, DM Port
  - /dev/ttyUSB1, NMEA Port
  - /dev/ttyUSB2, AT Port

2.1.4. GNSS Location Tracking

For products supporting USB compositions that expose an NMEA Serial Interface the GNSS (NMEA) location information tracking can be obtained using the following 2 methods:

- **NMEA Serial Port**
  - Location tracking session must be explicitly started issuing the following commands
    - sudo -i
    - echo -e "\$GPS_START" > /dev/ttyUSB1
    - (ttyUSB1 may vary, it is the associated NMEA port in this example).
  - This results in NMEA data being streamed over the NMEA Serial Port. The $GPS_START must be issued upon every power up or after every modem reset.

- **Lite-qmi-loc SDK sample application**
  - QMI host interface protocol provides a rich set of functions and notifications including NMEA sentences
  - A location tracking session can be started via AT and lite-qmi-loc in parallel if app indications and NMEA sentences via NMEA port are needed

2.1.5. Conflict with ModemManager

The Modem Manager daemon must be closed, since it uses the AT port and the MBIM interface, which prevents the interfaces from being used properly by the device drivers. Enter the following command:

```
sudo systemctl disable ModemManager
```

2.1.6. Conflict with Gobi Drivers

Gobi drivers which were included in older Sierra Wireless Linux SDK releases are no longer supported.

As indicated in previous sections, Mobile Broadband Package for Linux relies and includes the following open source drivers:

- cdc_mbim for the MBIM interfaces
- qmi_wwan for the RmNet interface(included)
- qcserial for all serial interfaces (included)

Using the sample apps included in MBPL require the old Gobi drivers to be removed or blacklisted:

```
blacklist GobiNet
blacklist GobiSerial
```

2.1.7. Host Software Architecture Diagram (USB-RmNet)

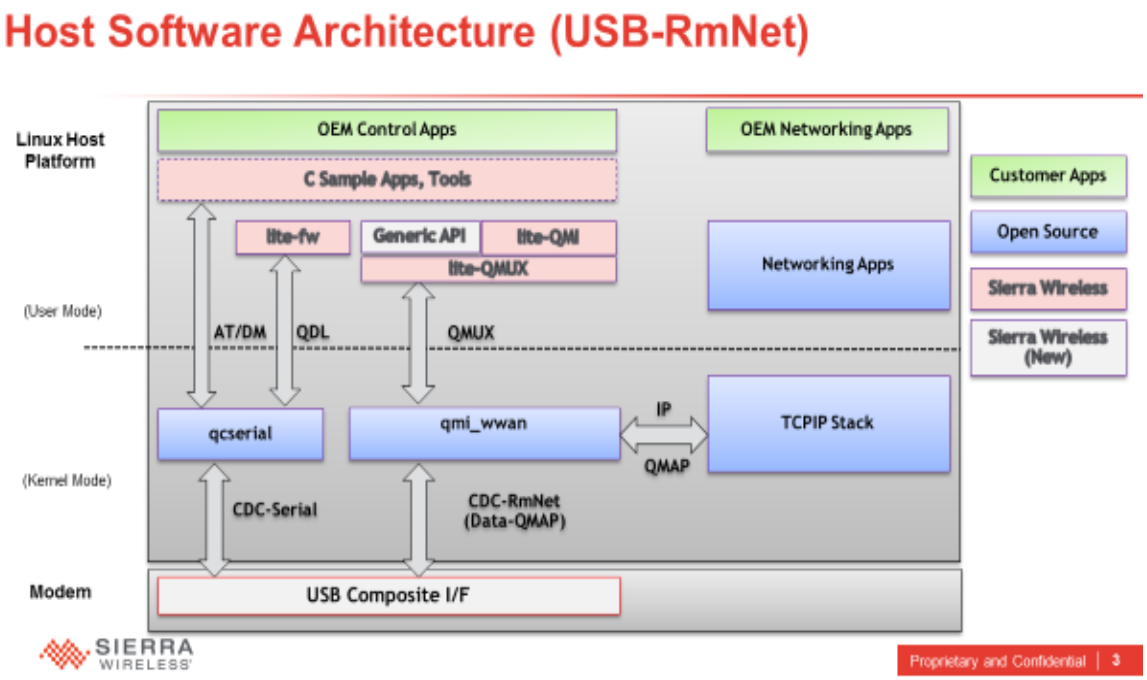

Figure 2. Host Software Architecture – USB, MBIM Interfaces

2.1.8. USB VID/PID/Interface Assignment

Table 1 lists the USB Vendor ID (VID), Product ID (PID) and USB Interface mapping used by Sierra Wireless modules.

Table 3. USB VID/PID and Supported Interfaces (9x50, 9x30, 9x07 based modules)

Mode	VID	PID	USB Interface ID	Interface Type
Application	0x1199	Device (SKU) Specific	0x00	DM Port
			0x02	NMEA
			0x03	AT Command Port
			0x08	Rmnet interface
Boot	0x1199	Device (SKU) Specific	0x00	QDLoader Port

2.1.9. Device Drivers

The following open source drivers must be used:

- RmNet Interface: open source (qmi_wwan)
- Serial Interfaces: open source serial driver (qcserial)
  - /dev/ttyUSB0, DM Port
  - /dev/ttyUSB1, NMEA Port (product and chipset dependent)
  - /dev/ttyUSB2, AT Port

Build instructions for the device drivers are located in the USB drivers source folder.

Location tracking NMEA data can be obtained via 2 methods:

- NMEA Serial Port
  - Location tracking session must be explicitly started issuing the following commands
    - sudo -i
    - echo -e "\$GPS_START" > /dev/ttyUSB1

- Lite-qmi-loc SDK sample application
  - A location tracking session can be started via AT and lite-qmi-loc in parallel if app indications and NMEA sentences via NMEA port are needed

2.1.10. Conflict with Gobi Drivers

Gobi drivers which were included in older Sierra Wireless Linux SDK releases are no longer supported.

As indicated in previous sections, Mobile Broadband Package for Linux uses and includes the following open source drivers:

- cdc_mbim for the MBIM interfaces
- qmi_wwan for the RmNet interface
- qcserial for all serial interfaces

Using the sample apps included in MBPL require the old Gobi drivers to be removed or blacklisted:

```
blacklist GobiNet
blacklist GobiSerial
```

3. Linux SDK

The MBPL's Linux SDK consists of several layers handling low-level host interface protocols.

The SDK and all its libraries have a low memory footprint. The application developer is responsible for handling inter-process communication. The top layer of the SDK (the lite-qmi library) includes a set of APIs that provide thread synchronization techniques.

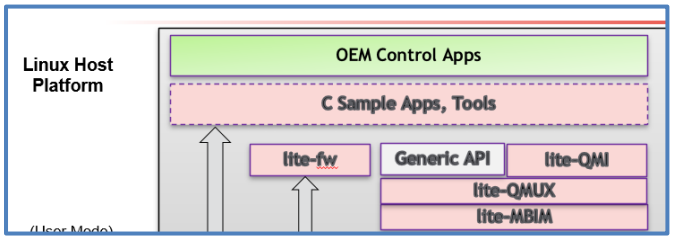

Figure 3. Linux SDK Structure

3.1. lite-qmi

QMI (Qualcomm MSM Interface) is an asynchronous host-module communication protocol used by Qualcomm chipset-based Sierra Wireless modules.

The lite-qmi library provides a simplified wrapper around QMI objects, which allows application developers to manage inter-process communication without directly working with QMI packets.

APIs are provided to facilitate the following:

- Populate (pack) outgoing (host-to-module) QMI packets from fixed user parameter structs
- Parse/unpack incoming (module-to-host) QMI packets to fixed user parameter structs

The APIs are logical interface (or device path) agnostic:

- The APIs have no knowledge of the QMUX protocol and do not prepend QMUX headers.
- The APIs have no knowledge of the host protocol transport methods for QMI packets to/from the module. (the logical interface used to communicate with the module)

Note: lite-qmi has no dependency on lite-qmux or lite-mbim and may be used independently.

3.2. lite-qmux

The lite-qmux library implements the QMUX protocol, which allows multiplexing of messages associated with different QMI services, where each QMI service is assigned a unique 'client ID'.

Note: This is an optional library, which makes minimal use of the pthread library.

The QMUX protocol uses QMI packets as its SDU (Service Data Unit).

The lite-qmux layer is responsible for transferring QMI packets between lite-qmi and device drivers. It acquires and releases QMI client IDs using the QMI control service.

The library shields the application developer from the underlying host interface used (direct QMUX, or MBIM, PCIe or USB).

For library details (APIs, data structures, etc.), refer to documentation in the SDK docs folder.

Note: The APIs which operate on these structs typically have a parameter of that type, called "pThis". This is to mimic the concept of the C++ "this" pointer.

3.3. Multiple-application support

Only one application may communicate with Lite QMI APIs at any time.

If several applications must communicate with the module simultaneously, use the following process:

1. Use lite-qmi to build/parse QMI packets.
   Use the open-source libmbim library's mbim-proxy to multiplex the QMI packets over the MBIM char device to/from the module. (Note – mbim-proxy will manage the file descriptor.)

Note: The lite-qmux and lite-mbim libraries do not support multiple-applications.

3.4. Supporting AT!SCACT data connections

Some Sierra Wireless USB modems that expose the RmNet interface support initiating data connections using an AT command (AT!SCACT).

MBPL includes a simple sample app (lite-cm-daemon) that support this scenario (qmi_wwan driver).

This daemon performs the following functionality:

- Initializes the WDS QMI service
- Sets the IP address, MTU size and routing based on data connections status changes.

This replaces similar functionality that was provided by the old GobiNet drivers (QMI WDS services and DHCP).

3.5. Multiple PDN connections

Driver releases starting with R10.0.20 B5068 support multiple PDN connections.

The included connection-manager samples demonstrate this feature:

- lite-qmi-connection manager
- lite-mbim-connection manager

3.6. MTU Size

MTU size is typically advertised by most cellular WWAN networks and must be configured correctly for best performance.

The MTU supported by the WWAN network can be retrieved depending on the control interface used:

- MBIM (USB)
  - Use the MBIM_CID_IP_CONFIGURATION command
- QMUX (RmNet)
  - Use the GetRuntimeSettings() API

Note: The MTU size does NOT include the ethernet header.

Lite-qmi-connection-manager and lite-mbim-connection-manager sample apps demonstrate:

- how to acquire the MTU size used by the WWAN network, and
- how to configure the network interface (including the multiple PDN/VLAN scenario):
    - Set the MTU of the base interface (using netlink) to the maximum value of all the MTUs of all the sessions as reported by WWAN.
    - Set individual VLAN interface MTUs as the minimum for all MTUs reported by the WWAN network for that specific session
    - The important point is that no VLAN interface should have an MTU greater than the MTU of the base interface, otherwise netlink reports an ERANGE error.

Example:

- Session 1 IP config IPv4 MTU 1460
- Session 1 IP config IPv6 MTU 1500
- Session 2 IP config has IPv4 MTU 1500
- Base interface MTU must be set to 1500

3.7. Developer Notes

3.7.1. Buffers

Several APIs in the lite-qmi, lite-qmux, and lite-mbim libraries require the caller to pass in a parameter that is a pointer to a buffer, and a second parameter representing the buffer's length.

It is the caller's responsibility to ensure the buffer length matches the buffer capacity exactly. Otherwise, buffer overruns may occur resulting in stack/heap corruption and associated unrecoverable errors.

3.7.2. QMI Services

For QMI service registration and release, applications should:

- Wait at least 1 second between a QMI service registration and release.
- Wait at least 1 second between a QMI service release and re-registration on the same service.

For example, wait at least 1 second between acquiring a QMI service (via CtlService_InitializeRegularService()) and releasing the service (via CtlService_ShutDownRegularService()).

3.7.3. Sample App makefiles

Developers may set the CROSS_COMPILE variable when invoking make to override the default tool path:

- Building natively (using the default tool path):

```
make complete CPU=arm64 CROSS_COMPILE=
```

- Building with a specific set of tools:

```
make complete CPU=arm64 CROSS_COMPILE=<specific_tools_preamble>
```

The <specific_tools_preamble> is typically the full path to the specific gcc binary with the rightmost "gcc" removed.

For example, if the path to the gcc binary is /tools/arm-xxx-x64/bin/arm-linux-gnueabi-gcc, the make command would be:

```
make complete CPU=arm64 CROSS_COMPILE=/tools/arm-xxx-x64/bin/arm-linux-gnueabi-
```

When cross-compiling, developers must ensure the appropriate system includes/libraries are referenced, if necessary, by setting the variable GCC_SYS_ROOT. This adds the gcc option --sysroot=<dir>. If the cross-compiler does not support --sysroot option, it is up to developer to modify makefiles directly to set the appropriate system includes/libraries path. For example:

```
make complete GCC_SYS_ROOT=<dir>
```

4. API Documentation

Detailed API documentation, including QMUX and MBIM layers is available in the ‘docs’ folder. (lite-qmi, lite-qmux, lite-mbim)

Each sample app comes with a detailed readme.txt file that provides build and usage instructions.

Additional documentation may also be provided outside the MBPL Package.

5. Reference Documents

[1] MBPL Release Notes

6. Glossary of terms and definitions

Table 4. Table 1 Glossary of Terms and Definitions

Abbreviation/Acronym	Definitions
API	Application Programming Interface
Device	Sierra Wireless Embedded Module
Host	Linux based Host Platform
Image	Device Firmware Image
MBIM	Mobile Broadband Interface Model
MBPL	Mobile Broadband Package for Linux
MHI	Modem Host Interface
PRI	Product Release Instructions
QDL	Qualcomm Download Mode
QMAP	Qualcomm IP Multiplexing Protocol
QMI	Qualcomm MSM Interface (Host-Modem communication protocol)
QMUX	QMI Multiplexing Protocol
SDK	Software Development Kit
SDU	Service Data Unit – information buffer passed between different software layers
skb	Socket buffers