AirCard / AirPrime

USB Driver Developer's Guide

2130634
Rev 5

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 blasting is in progress, where explosive atmospheres may be present, near medical equipment, near life support equipment, or any equipment which may be susceptible to any form of radio interference. In such areas, the Sierra Wireless modem **MUST BE POWERED OFF**. 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.

Limitation of Liability

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.

Patents

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

This product includes technology licensed from QUALCOMM® 4G.

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

Copyright

©2011 Sierra Wireless. All rights reserved.

Trademarks

AirCard® is a registered trademark of Sierra Wireless. Sierra Wireless™, AirPrime™, AirLink™, AirVantage™, Watcher™, and the Sierra Wireless logo are trademarks of Sierra Wireless.

Windows® and Windows Vista® are registered trademarks 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
	E-mail:	sales@sierrawireless.com
Post:	Sierra Wireless 13811 Wireless Way Richmond, BC Canada V6V 3A4	
Fax:	1-604-231-1109	
Web:	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

Revision History

Revision number	Release date	Changes
1.3	Jun 2007	- Added MC8785 / AC885E / AC885U - Updated idProduct values
1.4	Sep 2007	- Added vendor commands: Get TRU-Install Information, Set TRU-Install Mode

Revision number	Release date	Changes
1.5	Feb 2008	- Removed old products, added C885 - Updated USB descriptors for MC8785V / AC885E / C885 - Updated USB architecture for MC8785V / AC885E / C885 - Updated descriptions on non-MUX and MUX mode for MC8785V / AC885E / C885
1.6	Apr 2008	- Added MC5727 - Removed AC597U
1.8	Sep 2008	- Changed document title (removed 'Embedded Module', added 'Compass' - Added MC8790, AC8790V, C597, C888 - Reorganized and updated Table 1-1, Supported devices (by QUALCOMM chipset), on page 15 - Updated Mux, Non-Mux, and Hybrid-Mux sections in Chapter 2 to reflect details of new devices, including NMEA and Mass Storage - Added Mass Storage section - Split and updated USB descriptor information into multiple tables in Appendix A
1.9	Sep 2008	- Added MC8791V and MC8792V
1.10	Feb 2009	- Added AC402, MC5727V, and MC5728V - Added QUALCOMM DM Data Logging on page 103 - Added an Index on page 113 - Added List of Tables. - Added a summary table (previously was text) to the beginning of USB descriptors chapter. - Changed formatting (width/alignment) of tables
1.11	May 2009	- Added USB 306 / C889 - Added Direct IP content
1.12	Jul 2009	- Removed MC8785V - Added AC501, USB 307 - Updated Table 2-12, UMTS devices Direct IP—Non-MUX endpoints, on page 49 and Table 2-12, UMTS devices Direct IP—Non-MUX endpoints, on page 49 - Updated MC8700/USB306 USB descriptors - Updated Direct IP description (interfaces used)
2.0	May 2010	- Added MC8795V, MC8700, AC 310U - Removed MC5725/25V, MC8785V - Combined USB Architecture chapter and USB Descriptors appendix
3	May 2011	- Added SL5010 and SL5011. - Removed MC8775, MC8775V, MC8780, and MC8781. - Edits to the Patents section. - Added a List of Figures.
4	June 2011	- Added MDM6270/QSC6270 (SL808X) - Removed MDM6280, MSM6800, MSM6800A, MSM7200/7200A/7201/720A
5	November 2011	- Added QMI content

Contents

Introduction 15
Document organization 17
Related documents 18

USB architecture 19
USB standards compliance 19
USB endpoints 19
Supported driver architectures 20
Default operating mode 20
Activating / Deactivating MUX mode 21
AirCard/AirPrime USB architectures 21
CDMA module-specific USB architectures 22
CDMA devices—Non-MUX 23
CDMA devices—MUX 28
CDMA devices—Hybrid-MUX 30
UMTS module-specific USB architectures 32
UMTS devices (legacy VID/PID)—Non-MUX 33
UMTS devices (legacy VID/PID)—MUX 44
UMTS devices (DIP VID/PID)—Non-MUX 47
UMTS devices (QMI VID/PID)—Non-MUX 59

Services 65
USB protocols 65
Control 65
Interrupt 65

Data services 66
AT/PPP 66
HIP (Host Interface Protocol) 67
CnS (Control and Status) 67
DM (QUALCOMM Diagnostic Monitoring) 68
NMEA (National Marine Electronics Association) 69
Data 1, Data 2, Data 3 (Additional PPP) 70
Mass Storage 71
Direct IP 71
QMI (NET interface) 72

Vendor-specific commands 73

Command availability 73
Set Device Power State 75
Set Mode Non-MUX 76
Set Mode MUX 77
Get Mode MUX 79
Get NDIS Support 81
Get NDIS Preference 82
Get Attributes 83
Set Mode NMEA 85
Get Mode NMEA 86
Set Host Power State 87
Get TRU-Install Information 88
Set TRU-Install Mode 90
Get Config Item 91
Set Config Item 93
Set Device Reset 95
Set Host Info 96

Supported CDC USB commands 101

QUALCOMM DM Data Logging 103

Handling DM for logging purposes 103

Pass-thru application considerations 104

Direct IP mode 105

Performance considerations. 105

Direct IP drivers 105

Direct IP interface. 105

Direct IP communication / session setup 106

Examples 106

Switching between PPP and Direct IP modes 107

Configuring USB interface composition in Direct IP mode 107

QMI 109

QMI drivers. 109

QMI interface 109

QMI communication / session setup 109

Switching between QMI and Direct IP modes 109

Configuring USB interface composition in QMI mode 110

User-developed USB Drivers 111

User-developed drivers 111

Handshaking 111

Suspend 111

Resume 111

Host USB driver requirements 112

Index 113

List of Tables

Supported devices (by QUALCOMM chipset) 15
Related documentation 18
MUX mode activation/deactivation methods 21
Supported architectures 21
CDMA devices—Non-MUX endpoints 23
CDMA devices (non-MUX)—USB descriptors 25
CDMA devices—MUX endpoints/DLCI assignments 28
CDMA devices—Hybrid-MUX endpoints/DLCI assignments 30
UMTS devices (legacy VID/PID)—Non-MUX endpoints 33
UMTS devices (legacy VID/PID)—USB descriptors 36
MSM6290 (C888/C889) (TRU-Install mode)—USB descriptors 42
UMTS devices (legacy VID/PID)—MUX endpoints/DLCI assignments 44
Example—Interface-dependent endpoint assignments 48
UMTS devices Direct IP—Non-MUX endpoints 49
UMTS devices (DIP VID/PID)—Device/Configuration Descriptors 52
UMTS devices (DIP VID/PID)—HIP descriptors (Interface/Endpoint) 53
UMTS devices (DIP VID/PID)—DM descriptors (Interface/Endpoint) 53
UMTS devices (DIP VID/PID)—AT descriptors (Interface/Endpoint) 54
UMTS devices (DIP VID/PID)—MDM1 descriptors (Interface/Endpoint) 55
UMTS Mass Storage-capable devices (DIP VID/PID)—MS descriptors (Interface/Endpoint) . . 55
UMTS devices (DIP VID/PID)—DIP1 descriptors (Interface/Endpoint) 56
UMTS devices (DIP VID/PID)—DIP2 descriptors (Interface/Endpoint) 57
UMTS devices (DIP VID/PID)—DIP3 descriptors (Interface/Endpoint) 58
UMTS devices (QMI VID/PID)—Non-MUX endpoints 59
UMTS devices (QMI VID/PID)—USB descriptors 61
AT/PPP data channel endpoints (by chipset and mode) 66
HIP data channel endpoints (by chipset and mode) 67
DM data channel endpoints (by chipset and mode) 68
NMEA data channel endpoints (by chipset and mode) 69
Data 1/Data 2/Data 3 data channel endpoints (by chipset and mode) 70
Mass Storage data channel endpoints (by chipset and mode) 71
Direct IP data channel endpoints (by chipset) 72
QMI data channel endpoints (by chipset) 72
Command availability by chipset 73
Set Device Power State—Setup stage 75
Set Mode Non-MUX—Setup stage 76

Set Mode MUX—Setup stage 77
Set Mode MUX—Data stage (OUT) 78
Get Mode MUX—Setup stage 79
Get Mode MUX—Data stage (IN) 79
Get NDIS Support—Setup stage 81
Get NDIS Support—Data stage (IN) 81
Get NDIS Preference—Setup stage 82
Get NDIS Preference—Data stage (IN) 82
Get Attributes—Setup stage 83
Get Attributes—Data stage (IN) 84
Set Mode NMEA—Setup stage 85
Get Mode NMEA—Setup stage 86
Get Mode NMEA—Data stage (IN) 86
Set Host Power State—Setup stage 87
Get TRU-Install Information—Setup stage 88
Get TRU-Install Information—Data stage (IN) 89
Set TRU-Install Mode—Setup stage 90
Get Config Item—Setup stage 91
Get Config Item—Data stage (IN) 92
Configuration information details 92
Set Config Item—Setup stage 93
Set Config Item—Data stage (OUT) 94
Configuration information details 94
Set Device Reset—Setup stage 95
Set Host Info—Setup stage 96
Set Host Info—Data stage (OUT) 96
Configuration information—Feature Configuration 97
Configuration information—Host Type 97
Configuration information—Service pack version 98
Configuration information—Host and system up time 98
Supported commands/requests by chipset 101
Supported class-specific notifications by chipset 102

List of Figures

Figure 2-1: CDMA devices—Non-MUX endpoints 24
Figure 2-2: CDMA devices—MUX endpoints 29
Figure 2-3: CDMA devices—Hybrid-MUX endpoints/DLCI assignments 31
Figure 2-4: UMTS devices (legacy VID/PID)—Non-MUX endpoints 35
Figure 2-5: UMTS devices (legacy VID/PID)—MUX endpoints/DLCI assignments 46
Figure 2-6: UMTS devices Direct IP—Non-MUX endpoints 51
Figure 2-7: UMTS devices (QMI VID/PID)—Non-MUX endpoints 60
Figure C-1: Recommended setup for logging QUALCOMM DM data 103
Figure C-2: Handling of DM messages by your USB driver 104
Figure D-1: Direct IP—Receive Link Status Indication 106
Figure D-2: Direct IP—Host/modem synchronization 107

1: Introduction

The Sierra Wireless AirCard mobile broadband devices and AirPrime intelligent embedded modules listed in Table 1-1 include USB interfaces for communication with host devices. Each device's USB interface is based on the identified QUALCOMM chipset.

Table 1-1: Supported devices (by QUALCOMM chipset)[a]

Chipset	Network	Devices	Device type	PID	Network standards
MDM6085	CDMA	Data-only equivalent of QSC6085. See QSC6085 row (next page) for details.			
MDM6200	GSM	SL8090	SL	0x683C (Legacy) 0x683D (Legacy) 0x683E (Legacy) 0x68A3 (Direct IP)[b]	GSM/GPRS/EDGE/ WCDMA/HSDPA/ HSUPA
		SL8091	SL		
		SL8092	SL		
		SL8093	SL		
MDM6270	GSM	Data-only equivalent of QSC6270. See QSC6270 row (next page) for details.			
MDM8200	GSM	AC503	2-in-1 PC/E	0x68A3 (Direct IP)	GSM/GPRS/EDGE/ WCDMA/HSDPA/ HSUPA/HSPA+
		AirCard USB 308[c]	U		
		AirCard USB 309[a]	U		
		AirCard 310U[a]	U		
		MC8700	MC		
		USB 306	U		
		USB 307	U		
MDM8200A	GSM	AC318U	U	0x68A3 (Direct IP)	GSM/GPRS/EDGE/ WCDMA/HSDPA/ HSUPA/HSPA+
		AC326U	U		
		MC8704	MC		
		MC8705	MC		
MDM8220	GSM	AC312U	U	0x68A3 (Direct IP)	GSM/GPRS/EDGE/ WCDMA/HSDPA/ HSUPA/HSPA+
		AC319U	U		
		MC8801	MC		
MDM9200	GSM	AC313U	U	0x68A2 (QMI) 0x68A3 (Direct IP) 0x68AA (Direct IP)[b]	GSM/GPRS/EDGE/ WCDMA/HSDPA/ HSUPA/HSPA+/LTE
		AC320U	U		
		MC7700	MC		
		MC7710	MC		

Table 1-1: Supported devices (by QUALCOMM chipset)[a] (Continued)

Chipset	Network	Devices	Device type	PID	Network standards
MDM9600	GSM/CDMA	MC7750	MC	0x68A2 (QMI)	CDMA 1X/HDR/ eHRPD/LTE (HSPA capability to be available in a future firmware revision)
MSM6290	GSM	AC501	U	0x6890 (Legacy)	GSM/GPRS/EDGE/ WCDMA/HSDPA/ HSUPA
		AC504	2-in-1 PC/E	0x68A3 (Direct IP) 0x68AA (Direct IP)[b]	
		C888	U	0x6890 (Legacy)	
		C889	U	0x0FFF (TRU-Install) 0x68A3 (Direct IP) 0x68AA (Direct IP)[b]	
		MC8790	MC	0x683C (Legacy)	
		MC8790V	MC	0x68A3 (Direct IP) 0x68AA (Direct IP)[b]	
		MC8791V	MC		
		MC8792V	MC		
		MC8795V	MC		
		USB301	U	0x6890 (Legacy)	
		USB302	U	0x68A3 (Direct IP) 0x68AA (Direct IP)[b]	
QSC6085 **MDM6085** (Data-only)	CDMA	AC250U	U	0x0301	IS-95A/CDMA 1X 1xEV-DO Rev-0/ Rev-A
		AC402	2-in-1 PC/E	0x0027	
		AC598U	U	0x0025	
		MC5728V	MC	0x0028	
		SL5010	SL	0x0300	
		SL5011	SL	0x0300	
QSC6270 **MDM6270** (Data-only)	GSM	SL8080	SL	0x683C (Legacy) 0x683D (Legacy) 0x683E (Legacy) 0x68A3 (Direct IP) 0x68AA (Direct IP)[b]	GSM/GPRS/EDGE/ WCDMA/HSDPA
		SL8081[d]	SL		
		SL8082	SL		
		SL8083[d]	SL		
		SL8084	SL		
		SL8085[d]	SL		

a. E—ExpressCard; MC—Mini Card; PC—PC Card; U—USB; 2n1 PC/E—2-in-1 PC Card/ExpressCard; SL—SL module
b. Direct IP PID 0x68A3 is the 'standard' Direct IP PID. 0x68AA is carrier-specific.
c. AirCard USB 308, AirCard USB 309, and AirCard 310U may be referred to as USB308, USB309, and USB310 in this document.

d. SL8081, SL8083, and SL8085 may use either MDM6270 or QSC6270 chipsets.

Purpose of this guide

Note: Sierra Wireless provides drivers and a connection manager application (Watcher®) for Windows XP and Windows Vista®.

This guide is intended for use when designing non-Windows drivers for these devices. It describes the following information:

- Services (protocols) available over the USB connection (AT/PPP, HIP, CnS, DM, NMEA, Mass storage, Direct IP, QMI)
- The physical USB interface (device and endpoint descriptors)
- Supported driver architectures (non-MUX mode, MUX mode, hybrid-MUX mode)
- Commands used over the USB interface to control the module's state

Document organization

This guide includes the following sections:

- **Introduction** (this section)
- USB architecture on page 19
  Describes the physical interface, supported driver architectures, and device endpoint descriptor information.
- Services on page 65
  Describes available data transfer services (protocols).
- Vendor-specific commands on page 73
  Describes commands that can be used to control the module's state.
- Supported CDC USB commands on page 101
  Describes support for CDC USB commands, requests, and notifications.
- QUALCOMM DM Data Logging on page 103
  Contains information for OEMs who need to implement DM logging.
- Direct IP mode on page 105
  Describes Direct IP interface, performance, communications setup, and configuation.
- QMI on page 109
  Describes the NET interface used to support QMI.
- User-developed USB Drivers on page 111
  Provides information for developers who are creating their own drivers for Sierra Wireless MC- and SL-series embedded modules.

Related documents

Related and supporting documents and products include:

Table 1-2: Related documentation

Document title	Related content
AirCard/AirPrime UMTS Devices Supported AT Command Reference (Document 2130617)	Standard and proprietary AT commands supported by Sierra Wireless UMTS devices For descriptions of standard AT commands, see *AT Command Set for User Equipment (UE) (Release 6)*.
AT Command Set for User Equipment (UE) (Release 6)	Standard AT commands
CDMA 1xEV-DO CnS Reference (Document 2130754)	CnS (Control and Status) messages for use with Sierra Wireless CDMA devices
AT Command Reference (Document 2130620)	Standard AT commands supported by Sierra Wireless CDMA AirCards and MiniCards. See also *AT Command Set for User Equipment (UE) (Release 6)* (above).
CDMA SL501x AT Command Reference (Document 4110801)	Standard AT commands supported by Sierra Wireless SL501x devices. See also *AT Command Set for User Equipment (UE) (Release 6)* (above).
Extended AT Command Reference (Document 2130621)	Proprietary AT commands for Sierra Wireless CDMA devices For UMTS-specific commands, see *AirPrime MC/SL-series (UMTS/LTE) Extended AT Command Reference (Document 2130616)*.
AirPrime UMTS MC and SL Series CnS Reference (Document 2130602)	CnS (Control and Status) messages for use with Sierra Wireless UMTS devices.
AirPrime MC/SL-series (UMTS/LTE) Extended AT Command Reference (Document 2130616)	Proprietary AT commands for Sierra Wireless UMTS MC- and SL-series devices. For CDMA-specific commands, see the *Extended AT Command Reference (Document 2130621)*.
Terminal Equipment to User Equipment (TE-UE) multiplexer protocol (Release 6)	3GPP technical specification describing the 27.010 multiplexer protocol The document (3GPP TS 27.010) can be downloaded from www.3gpp.org.
Universal Serial Bus Class Definitions for Communication Devices, Version 1.1	This specification can be downloaded from www.usb.org.
Universal Serial Bus Specification, Rev 2.0	This specification can be downloaded from www.usb.org.

2: USB architecture

This chapter describes supported driver architectures for data transfer, the physical USB interfaces for each architecture by chipsets, and the USB descriptor information for those chipsets.

Note: UMTS devices use Sierra Wireless' Direct IP VID/PID (common to all devices supporting Direct IP), and some of these devices can switch between Direct IP and either a 'legacy' VID/PID (unique to each device) or a QMI VID/PID (common to all devices supporting QMI).

USB standards compliance

The devices listed in Table 1-1 on page 15 comply with the following USB standards:

- USB 2.0 (backwards compatible with USB 1.1)
- USB slave only
- Standard USB flow control
- Standard USB power management—Suspends the USB bus (when idle) to conserve power.
- Data transfer rates:
  - Low-speed (1.5 Mb/s)
  - Full-speed (12 Mb/s)
  - UMTS devices: High-speed (480 Mb/s)

The architecture for these devices is described in the *Universal Serial Bus Class Definitions for Communication Devices, Version 1.1* (CDC) specification, available from www.usb.org.

To support advanced power management, a modified (non-composite) endpoint/interface model combines the Communication and Data Interface pipes with two proprietary pipes (HIP and NMEA/Data1) into one vendor-specific interface.

Note: These devices do not claim any CDC classes in the descriptors and do not support functional descriptors. As a result, these devices are not normally compatible with native CDC-ACM drivers.

USB endpoints

USB endpoints are uniquely addressable portions of a USB device used to transfer information between the host and module. Each defined endpoint is a unidirectional link from the modem to the host (IN) or from the host to the modem (OUT).

In this document:

- Each endpoint is referred to by a logical endpoint number and direction for ease of reading (for example, '2-IN' or '3-OUT').
- Actual (physical) endpoint information is detailed in chipset-specific USB descriptor tables.
- Where bidirectional communication is required for a specific data service, two BULK endpoints are used.
- Some data services also use an INTERRUPT endpoint for data flow control.

Note: When developing your drivers, you must use the actual physical endpoints.

Supported driver architectures

AirCard and AirPrime modules support one or more driver architectures, depending on their chipset and, for some modules, the VID/PID (Vendor ID/Product ID) being used:

- Non-multiplexing (non-MUX) mode—Each supported service channel (see Services on page 65) is enabled over its own set of endpoints.
- Multiplexing (MUX) mode—Based on the 3GPP 27.010 multiplexer protocol, MUX mode defines a framework for supporting several simultaneous sessions over a single asynchronous serial interface (all service channels are carried over a single set of endpoints). If supported, Mass Storage is implemented over a second set of endpoints.
- Hybrid-MUX mode—This is a proprietary architecture (supported by CDMA modules) intended to reduce overhead on data packets and to improve performance. The following service channels are implemented over unique endpoint sets:
  - Data/AT
  - Mass Storage
  - All other services (MUXed)

*Note: Driver Implementation for MUX and hybrid-MUX protocols requires a detailed understanding of the 27.010 requirements. See **Terminal Equipment to User Equipment (TE-UE) multiplexer protocol (Release 6)** for the full protocol specification.*

Default operating mode

UMTS AirCard and AirPrime modules always start in non-MUX mode.

CDMA AirCard and AirPrime modules start in hybrid-MUX mode (if supported by the host system's drivers) or non-MUX mode.

Activating / Deactivating MUX mode

After a USB connection has been established between the host and the module, activate and deactivate MUX mode using the methods shown in Table 2-1.

Table 2-1: MUX mode activation/deactivation methods

Chipset[a]	Activate MUX mode	Deactivate MUX mode
MDM6085 **QSC6085**	• Vendor-specific USB commands[b] • **AT!CMUX** command	• Vendor-specific USB commands[a] • Use the **Close Down Service Request** over the 27.010 Control channel (see *Terminal Equipment to User Equipment (TE-UE) multiplexer protocol (Release 6), section 5.1.7* for details).
MDM6200[c] **MDM6270** **MSM6290**[c] **QSC6270**[c]	• Vendor-specific USB commands[b] • **AT!CMUX** command • **AT+CMUX** command	

a. For a list of devices using these chipsets, see Table 1-1 on page 15.
b. See Vendor-specific commands on page 73 for applicable commands.
c. Device must be using a legacy VID/PID. MUX mode is not supported when using Direct IP VID/PID.

AirCard/AirPrime USB architectures

Table 2-2: Supported architectures

Chipset[a]	PID			non-MUX	MUX	hybrid MUX	Notes
	Legacy	Direct IP	QMI				
CDMA							
MDM6085 **QSC6085**	✔	N/A	N/A	✔	✔	✔	non-MUX: • Proprietary, non-composite architecture • Each service type enabled over specific endpoints
UMTS/LTE							
MDM6200	✔[b]	✔	✘	✔[b]	✔[bc]	N/A	non-MUX: • Composite architecture • Each service type enabled over its own interface • (Direct IP only) Endpoint assignments depend on which services are enabled
MDM6270	✔[b]	✔	✘	✔[b]	✔[bc]	N/A	
MDM8200	✘	✔	✘	✔	✘	N/A	
MDM8200A	✘	✔	✘	✔	✘	N/A	
MDM8220	✘	✔	✘	✔	✘	N/A	
MDM9200	✘	✔	✔	✔	✘	N/A	
MDM9600	✘	✘	✔	✔	✘	N/A	
MSM6290	✔	✔	✘	✔	✔[c]	N/A	
QSC6270	✔[b]	✔	✘	✔[b]	✔[bc]	N/A	

a. For a list of devices using these chipsets, see Table 1-1 on page 15.
b. Device uses different PIDs for composite MUX (0x683C), composite non-MUX (0x683D), and non-composite MUX (0x683E).
c. MUX mode available only if device uses legacy VID/PID.

CDMA module-specific USB architectures

Sierra Wireless CDMA AirCard and AirPrime modules (detailed in Table 1-1 on page 15) implement the following chipset-specific USB architectures:

- CDMA devices—Non-MUX (See page 23.)
- CDMA devices—MUX (See page 28.)
- CDMA devices—Hybrid-MUX (See page 30.)

CDMA devices—Non-MUX

This section describes non-MUX mode for CDMA devices.

Device features	Details
Chipsets	• MDM6085 • QSC6085 • For a list of devices using these chipsets, see Table 1-1 on page 15.
Architecture	• Table 2-3, CDMA devices—Non-MUX endpoints, on page 23 • Figure 2-1, CDMA devices—Non-MUX endpoints, on page 24
Descriptors	• Table 2-4, CDMA devices (non-MUX)—USB descriptors, on page 25

CDMA devices—Non-MUX architecture

Table 2-3: CDMA devices—Non-MUX endpoints

Interface	Endpoint	Supported service
-	**0 (Control)**	Management element • Purpose: 0-OUT is used to emulate control line signals. • For details, see “SetControlLineState” in *Universal Serial Bus Class Definitions for Communication Devices, Version 1.1*.
0	**1 IN (Interrupt)**	Notification element • Purpose: Send asynchronous notification of modem status (for example, DSR, DCD, and RI) to the host. • For details, see “SerialState” in the *Universal Serial Bus Class Definitions for Communication Devices, Version 1.1*.
	2 IN / OUT (Bulk)	Primary data channel (AT/PPP)
	4 IN / OUT (Bulk)	HIP (carrying CnS) • If required, OEMs can consult with Sierra Wireless to discuss how to change the default data service. • Supported service types (device-dependent): DM, HIP (carrying CnS), and HIP (carrying CnS and DM).
	5 IN / OUT (Bulk)	NMEA
	8 IN / OUT (Bulk)	DM

*Note: You can customize port mappings for these devices in non-MUX mode using the **AT!SERIALPORTMAP** command. For details, see **Extended AT Command Reference (Document 2130621)**.*

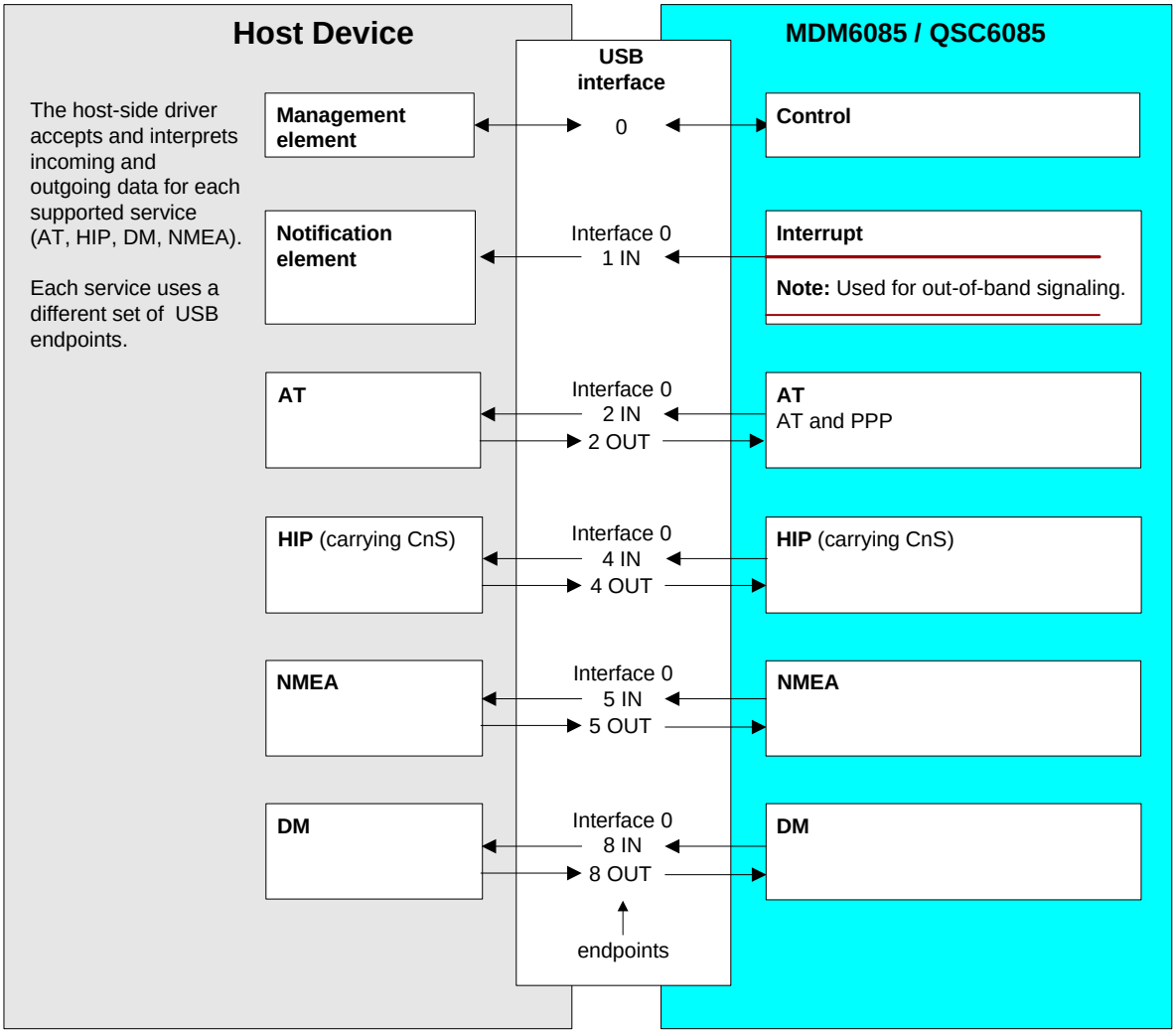

Figure 2-1: CDMA devices—Non-MUX endpoints

CDMA devices—Non-MUX descriptors

Table 2-4: CDMA devices (non-MUX)—USB descriptors

Descriptor	Field identifier	Value	
Device Descriptor	bLength	0x12	
	bDescriptorType	0x01	
	bcdUSB	0x0110	USB spec 01.10
	bDeviceClass	0x00	IFCs supply own
	bDeviceSubClass	0x00	
	bDeviceProtocol	0x00	
	bMaxPacketSize0	0x40	
	idVendor	0x1199	Sierra Wireless Inc.
	idProduct	(Default values)	
		0x0025	AC598U
		0x0027	AC402
		0x0028	MC5728V
		0x0300	SL5010/SL5011
		0x0301	AC250U
	bcdDevice	0x0001	Release 00.01
	iManufacturer	0x01	
	iProduct	0x02	
	iSerialNumber	0x00	
	bNumConfigurations	0x01	1 configuration
Configuration Descriptor (Index 0x00)	bLength	0x09	
	bDescriptorType	0x02	
	wTotalLength	0x0043	67 bytes
	bNumInterfaces	0x01	1 interface
	bConfigurationValue	0x01	
	iConfiguration	0x00	
	bmAttributes	0xE0	Self powered—remote wakeup
	MaxPower	0x00	0 mA
Interface Descriptor (Number 0x00 Alternate Setting 0x00)	bLength	0x09	
	bDescriptorType	0x04	
	bInterfaceNumber	0x00	
	bAlternateSetting	0x00	
	bNumEndpoints	0x09	
	bInterfaceClass	0xFF	Vendor-specific device class
	bInterfaceSubClass	0xFF	Vendor-specific device subclass
	bInterfaceProtocol	0xFF	Vendor-specific
	iInterface	0x03	
Endpoint Descriptor (0x81)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x81	ep #1, IN
	bmAttributes	0x03	Transfer Type: Interrupt
	wMaxPacketSize	0x0010	
	bInterval	0x80	128 ms

Table 2-4: CDMA devices (non-MUX) — USB descriptors (Continued)

Descriptor	Field identifier	Value	
Endpoint Descriptor (0x82)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x82	ep #2, IN
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0040	
	bInterval	0x00	0 ms
Endpoint Descriptor (0x02)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x02	ep #2, OUT
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0040	
	bInterval	0x00	0 ms
Endpoint Descriptor (0x84)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x84	ep #4, IN
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0040	
	bInterval	0x00	0 ms
Endpoint Descriptor (0x04)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x04	ep #4, OUT
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0040	
	bInterval	0x00	0 ms
Endpoint Descriptor (0x85)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x85	ep #5, IN
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x40	
	bInterval	0x00	0 ms
Endpoint Descriptor (0x05)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x05	ep #5, OUT
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x40	
	bInterval	0x00	0 ms
Endpoint Descriptor (0x88)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x88	ep #8, IN
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x40	
	bInterval	0x00	0 ms

Table 2-4: CDMA devices (non-MUX)—USB descriptors (Continued)

Descriptor	Field identifier	Value
Endpoint Descriptor (0x08)	bLength	0x07
	bDescriptorType	0x05
	bEndpointAddress	0x08 ep #8, OUT
	bmAttributes	0x02 Transfer Type: Bulk
	wMaxPacketSize	0x40
	bInterval	0x00 0 ms
Manufacturer String Descriptor	bLength	60
	bDescriptorType	03
	bString	(Default value) “Sierra Wireless, Incorporated”
Product String Descriptor	bLength	62
	bDescriptorType	0x03
	bString	“Sierra Wireless Device”
Interface String Descriptor	bLength	30
	bDescriptorType	0x03
	bString	“Data Interface”
String Descriptor 0 (Language Descriptor)	bLength	4
	bDescriptorType	0x03
	wLANGID[0]	0x0409 English—United States
Port Configuration String Descriptor (0xF0)	bLength	0x08
	bDescriptorType	0x03
	bString	(Six consecutive bytes) 0x0031 MUX supported 0x0030 Hybrid not supported 0x0031 NMEA supported

CDMA devices—MUX

This section describes MUX mode for CDMA devices.

Device features	Details
Chipsets	• MDM6085 • QSC6085 • For a list of devices using these chipsets, see Table 1-1 on page 15.
Architecture	• Table 2-5, CDMA devices—MUX endpoints/DLCI assignments, on page 28 • Figure 2-2, CDMA devices—MUX endpoints, on page 29

CDMA devices—MUX architecture

Table 2-5: CDMA devices—MUX endpoints/DLCI assignments

Interface	Endpoints	Supported service
-	**0 (Control)**	Management element • Purpose: 0-OUT is used to emulate control line signals. • For details, see "SetControlLineState" in *Universal Serial Bus Class Definitions for Communication Devices, Version 1.1*.
0	**1 IN (Interrupt)**	Notification element • Purpose: Send asynchronous notification of modem status (for example, DSR, DCD, and RI) to the host. • For details, see "SerialState" in *Universal Serial Bus Class Definitions for Communication Devices, Version 1.1*.

	2 IN / OUT (Bulk)	Data
		• DLCI Channels:
		  ◦ 0—27.010 Control
		  ◦ 1—AT/PPP
		  ◦ 2—DM
		  ◦ 3—HIP (carrying CnS)
		  ◦ 4—NMEA (enabled by default on AC402/MC5728V/SL5010/SL5011)

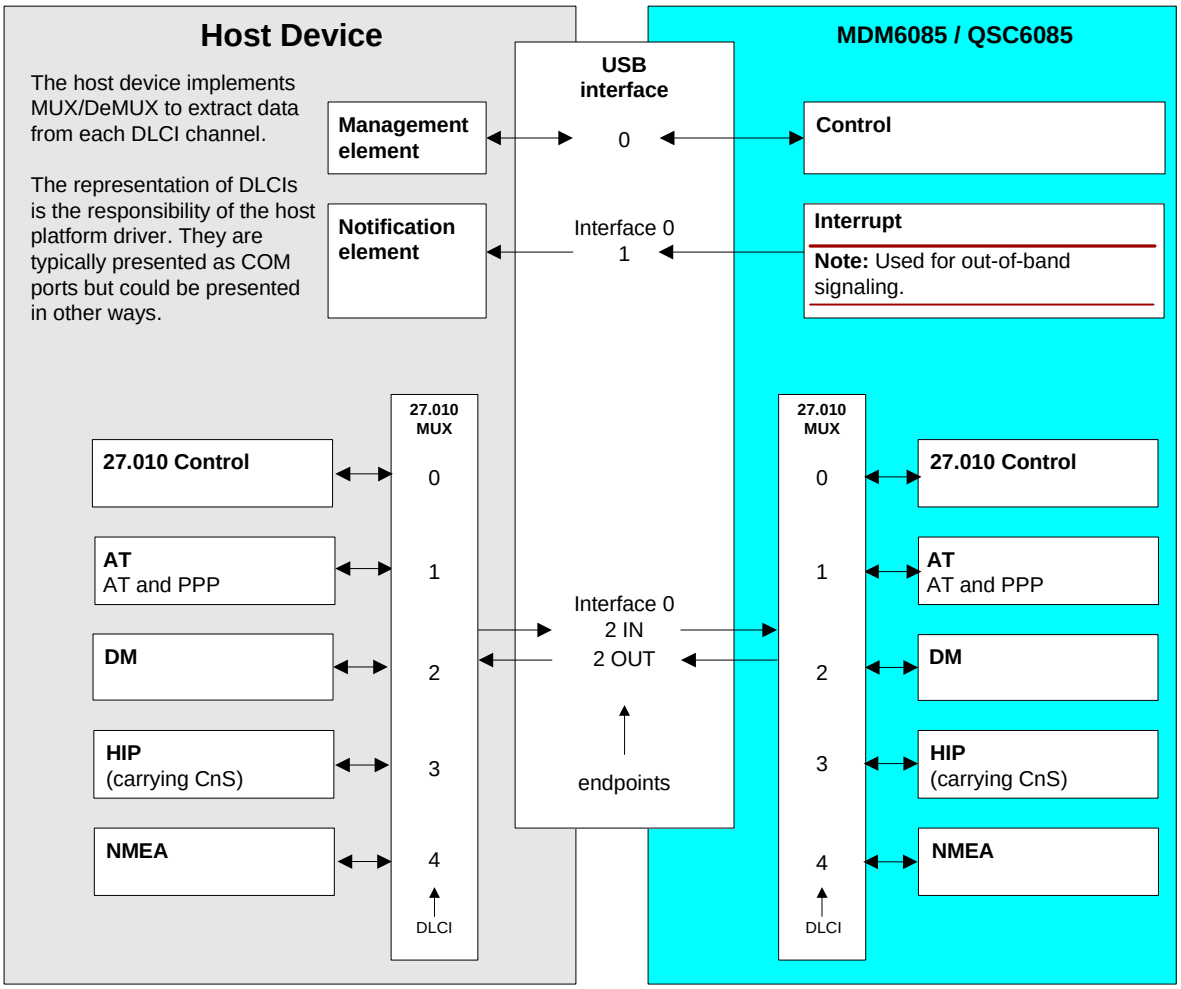

Figure 2-2: CDMA devices—MUX endpoints

CDMA devices—Hybrid-MUX

This section describes hybrid-MUX mode for CDMA devices.

Device features	Details
Chipsets	• MDM6085 • QSC6085 • For a list of devices using these chipsets, see Table 1-1 on page 15.
Architecture	• Table 2-6, CDMA devices—Hybrid-MUX endpoints/DLCI assignments, on page 30 • Figure 2-3, CDMA devices—Hybrid-MUX endpoints/DLCI assignments, on page 31

CDMA devices—Hybrid-MUX architecture

Table 2-6: CDMA devices—Hybrid-MUX endpoints/DLCI assignments

Interface	Endpoints	Supported service
	0 (Control)	Management element • Purpose: 0-OUT is used to emulate control line signals. • For details, see "SetControlLineState" in *Universal Serial Bus Class Definitions for Communication Devices, Version 1.1*.
0	**1 IN (Interrupt)**	Notification element • Purpose: Send asynchronous notification of modem status (for example, DSR, DCD, and RI) to the host. • For details, see "SerialState" in *Universal Serial Bus Class Definitions for Communication Devices, Version 1.1*.
	2 IN / OUT (Bulk)	Data (AT/PPP)

	4 IN / OUT (Bulk)	Data
		• DLCI Channels:
		  ◦ 0—27.010 Control
		  ◦ 2—DM
		  ◦ 3—HIP (carrying CnS)
		  ◦ 4—NMEA (Only on GPS-enabled devices. Enabled by default on AC402/MC5728V/SL5010/SL5011.)

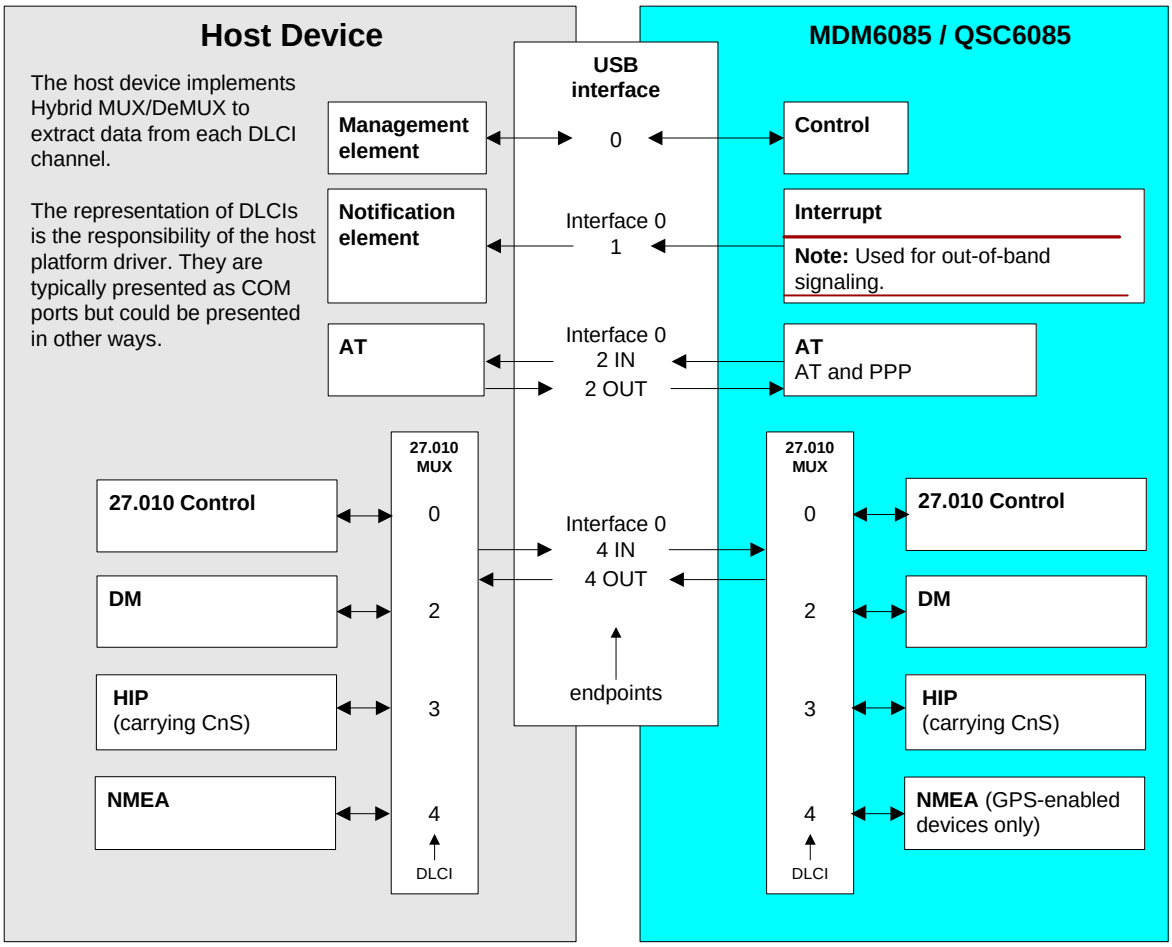

Figure 2-3: CDMA devices—Hybrid-MUX endpoints/DLCI assignments

UMTS module-specific USB architectures

Sierra Wireless UMTS AirCard and AirPrime modules (detailed in Table 1-1 on page 15) implement the following chipset-specific USB architectures:

- UMTS devices (legacy VID/PID)—Non-MUX (See page 33.)
- UMTS devices (legacy VID/PID)—MUX (See page 44.)
- UMTS devices (DIP VID/PID)—Non-MUX (See page 47.)
- UMTS devices (QMI VID/PID)—Non-MUX (See page 59.)

UMTS devices (legacy VID/PID)—Non-MUX

This section describes non-MUX mode for UMTS devices using a ‘legacy’ VID/PID.

Device features	Details
Chipsets	• MDM6200 • MDM6270 • MSM6290 • QSC6270 • For a list of devices using these chipsets, see Table 1-1 on page 15.
Architecture	• Table 2-7, UMTS devices (legacy VID/PID)—Non-MUX endpoints, on page 33 • Figure 2-4, UMTS devices (legacy VID/PID)—Non-MUX endpoints, on page 35
Descriptors	• Table 2-8, UMTS devices (legacy VID/PID)—USB descriptors, on page 36 • Table 2-9, MSM6290 (C888/C889) (TRU-Install mode)—USB descriptors, on page 42

UMTS devices (legacy VID/PID)—Non-MUX architecture

Table 2-7: UMTS devices (legacy VID/PID)—Non-MUX endpoints

Interface	Endpoint	Supported service
	0 (Control)	Management element • Purpose: 0-OUT is used to emulate control line signals. • For details, see “SetControlLineState” in the “Universal Serial Bus Class Definitions for Communication Devices” document.
0	**1 IN / OUT (Bulk)**	HIP
1	**2 IN / OUT (Bulk)**	QUALCOMM DM diagnostics
2	**3 IN / OUT (Bulk)**	NMEA
3	**4 IN (Interrupt)**	Notification element • Purpose: Send asynchronous notification of modem status (for example, DSR, DCD, and RI) to the host. • For details, see “SerialState” in *Universal Serial Bus Class Definitions for Communication Devices, Version 1.1*.
	5 IN / 4 OUT (Bulk)	AT

Table 2-7: UMTS devices (legacy VID / PID) — Non-MUX endpoints (Continued)

Interface	Endpoint	Supported service
4	**6 IN (Interrupt)**	Notification element • Purpose: Send asynchronous notification of modem status (for example, DSR, DCD, and RI) to the host. • For details, see "SerialState" in *Universal Serial Bus Class Definitions for Communication Devices, Version 1.1*.
	7 IN / 5 OUT (Bulk)	Data 1 (PDP context 1)
5 (MSM6290 only)	**8 IN (Interrupt)**	Notification element • Purpose: Send asynchronous notification of modem status (for example, DSR, DCD, and RI) to the host. • For details, see "SerialState" in the *Universal Serial Bus Class Definitions for Communication Devices, Version 1.1*.
	9 IN / 6 OUT (Bulk)	Data 2 (PDP context 2)
6 (MSM6290 only)	**10 IN (Interrupt)**	Notification element • Purpose: Send asynchronous notification of modem status (for example, DSR, DCD, and RI) to the host. • For details, see "SerialState" in the *Universal Serial Bus Class Definitions for Communication Devices, Version 1.1*.
	11 IN / 7 OUT (Bulk)	Data 3 (PDP context 3)
7 (MSM6290 only)	**12 IN / 8 OUT (Bulk)** *Note: TRU-Install mode uses Interface 0 (endpoints 1-IN and 1-OUT).*	(C8XX only) Mass storage (when enabled) • You can access the microSD device in TRU-Install mode (Interface 0) and modem mode (Interface 7). • In TRU-Install mode, the TRU-Install CD-ROM and SD disk are presented as two logical unit numbers (LUNs).

*Note: In these products, each COM port has its own interface and their endpoint assignments are found in the USB interface descriptor. Do not use the **AT!NVPORTMAP** command with these products.*

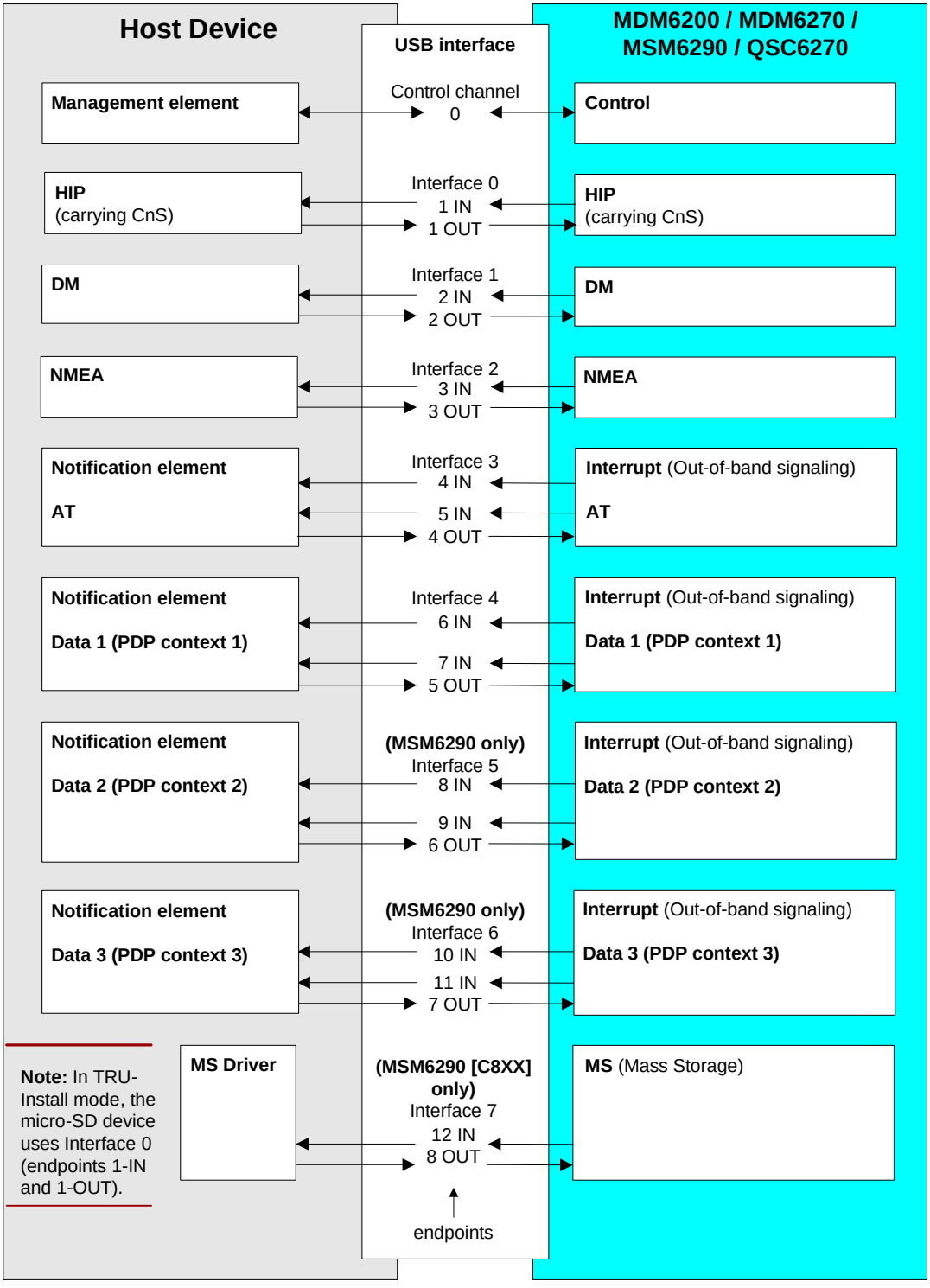

Figure 2-4: UMTS devices (legacy VID/PID)—Non-MUX endpoints

UMTS devices (legacy VID/PID)—Non-MUX descriptors

Table 2-8: UMTS devices (legacy VID / PID) — USB descriptors

Descriptor	Field identifier	Value	
Device Descriptor	bLength	0x12	
	bDescriptorType	0x01	
	bcdUSB	0x0200	USB spec version 2
	bDeviceClass	0x00	
	bDeviceSubClass	0x00	
	bDeviceProtocol	0x00	
	bMaxPacketSize0	0x40	Max packet size for endpoint 0 = 64
	idVendor	0x1199	Sierra Wireless Inc.
	idProduct	0x683C	MC8790, MC8790V, MC8791V, MC8792V, MC8795V
		0x683D	SL8080, SL8081, SL8082, SL8083, SL8084, SL8085 SL8090, SL8091, SL8092, SL8093
		0x6890	AC501, AC504, C888, C889, USB301, USB302
	bcdDevice	0x0006	
	iManufacturer	0x03	
	iProduct	0x02	
	iSerialNumber	0x00	
	bNumConfigurations	0x01	
Configuration Descriptor (Index 0x00)	bLength	0x09	
	bDescriptorType	0x02	
	wTotalLength	0x00C6	
	bNumInterfaces	0x05	SL8080, SL8081, SL8082, SL8083, SL8084, SL8085 SL8090, SL8091, SL8092, SL8093
		0x07	AC501, AC504, MC8790, MC8790V, MC8791V, MC8792V, MC8795V, USB301, USB302
		0x07	C888, C889 (Mass Storage not enabled)
		0x08	C888, C889 (Mass Storage enabled)
	bConfigurationValue	0x01	
	iConfiguration	0x00	
	bmAttributes	0xE0	
	MaxPower	0x00	

Table 2-8: UMTS devices (legacy VID/PID)—USB descriptors (Continued)

Descriptor	Field identifier	Value	
Interface Descriptor (Number 0x00 Alternate Setting 0x00)	bLength	0x09	
	bDescriptorType	0x04	
	bInterfaceNumber	0x00	
	bAlternateSetting	0x00	
	bNumEndpoints	0x02	Number of endpoints used by this interface is 2, excluding endpoint 0.
	bInterfaceClass	0xFF	Vendor-specific device class
	bInterfaceSubClass	0xFF	Vendor-specific device subclass
	bInterfaceProtocol	0xFF	
	iInterface	0x00	
Endpoint Descriptor (0x81)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x81	ep #1, IN
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms
Endpoint Descriptor (0x01)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x01	ep #1, OUT
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms
Interface Descriptor (Number 0x01 Alternate Setting 0x00)	bLength	0x09	
	bDescriptorType	0x04	
	bInterfaceNumber	0x01	
	bAlternateSetting	0x00	
	bNumEndpoints	0x02	Number of endpoints used by this interface is 2, excluding endpoint 0.
	bInterfaceClass	0xFF	Vendor-specific device class
	bInterfaceSubClass	0xFF	Vendor-specific device subclass
	bInterfaceProtocol	0xFF	
	iInterface	0x00	
Endpoint Descriptor (0x82)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x82	ep #2, IN
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms
Endpoint Descriptor (0x02)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x02	ep #2, OUT
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms

Table 2-8: UMTS devices (legacy VID / PID) — USB descriptors (Continued)

Descriptor	Field identifier	Value	
Interface Descriptor (Number 0x02 Alternate Setting 0x00)	bLength	0x09	
	bDescriptorType	0x04	
	bInterfaceNumber	0x02	
	bAlternateSetting	0x00	
	bNumEndpoints	0x02	Number of endpoints used by this interface is 2, excluding endpoint 0.
	bInterfaceClass	0xFF	Vendor-specific device class
	bInterfaceSubClass	0xFF	Vendor-specific device subclass
	bInterfaceProtocol	0xFF	
	iInterface	0x00	
Endpoint Descriptor (0x83)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x83	ep #3, IN
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms
Endpoint Descriptor (0x03)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x03	ep #3, OUT
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms
Interface Descriptor (Number 0x03 Alternate Setting 0x00)	bLength	0x09	
	bDescriptorType	0x04	
	bInterfaceNumber	0x03	
	bAlternateSetting	0x00	
	bNumEndpoints	0x03	Number of endpoints used by this interface is 3, excluding endpoint 0.
	bInterfaceClass	0xFF	
	bInterfaceSubClass	0xFF	
	bInterfaceProtocol	0xFF	
	iInterface	0x00	
Endpoint Descriptor (0x84)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x84	ep #4, IN
	bmAttributes	0x03	Transfer Type: Interrupt
	wMaxPacketSize	0x0040	
	bInterval	0x05	5 ms
Endpoint Descriptor (0x85)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x85	ep #5, IN
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms

Table 2-8: UMTS devices (legacy VID/PID)—USB descriptors (Continued)

Descriptor	Field identifier	Value	
Endpoint Descriptor (0x04)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x04	ep #4, OUT
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms
Interface Descriptor (Number 0x04 Alternate Setting 0x00)	bLength	0x09	
	bDescriptorType	0x04	
	bInterfaceNumber	0x04	
	bAlternateSetting	0x00	
	bNumEndpoints	0x03	Number of endpoints used by this interface is 3, excluding endpoint 0.
	bInterfaceClass	0xFF	Vendor-specific device class
	bInterfaceSubClass	0xFF	
	bInterfaceProtocol	0xFF	
	iInterface	0x00	
Endpoint Descriptor (0x86)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x86	ep #6, IN
	bmAttributes	0x03	Transfer Type: Interrupt
	wMaxPacketSize	0x0040	
	bInterval	0x05	5 ms
Endpoint Descriptor (0x87)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x87	ep #7, IN
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms
Endpoint Descriptor (0x05)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x05	ep #5, OUT
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms
(MSM6290 only) **Interface Descriptor (Number 0x05 Alternate Setting 0x00)**	bLength	0x09	
	bDescriptorType	0x04	
	bInterfaceNumber	0x05	
	bAlternateSetting	0x00	
	bNumEndpoints	0x03	Number of endpoints used by this interface is 3, excluding endpoint 0.
	bInterfaceClass	0xFF	Vendor-specific device class
	bInterfaceSubClass	0xFF	
	bInterfaceProtocol	0xFF	
	iInterface	0x00	

Table 2-8: UMTS devices (legacy VID / PID) — USB descriptors (Continued)

Descriptor	Field identifier	Value	
(MSM6290 only) **Endpoint Descriptor (0x88)**	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x88	ep #8, IN
	bmAttributes	0x03	Transfer Type: Interrupt
	wMaxPacketSize	0x0040	
	bInterval	0x05	5 ms
(MSM6290 only) **Endpoint Descriptor (0x89)**	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x89	ep #9, IN
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms
(MSM6290 only) **Endpoint Descriptor (0x06)**	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x06	ep #6, OUT
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms
(MSM6290 only) **Interface Descriptor (Number 0x06 Alternate Setting 0x00)**	bLength	0x09	
	bDescriptorType	0x04	
	bInterfaceNumber	0x06	
	bAlternateSetting	0x00	
	bNumEndpoints	0x03	Number of endpoints used by this interface is 3, excluding endpoint 0.
	bInterfaceClass	0xFF	Vendor-specific device class
	bInterfaceSubClass	0xFF	
	bInterfaceProtocol	0xFF	
	iInterface	0x00	
(MSM6290 only) **Endpoint Descriptor (0x8A)**	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x8A	ep #10, IN
	bmAttributes	0x03	Transfer Type: Interrupt
	wMaxPacketSize	0x0040	
	bInterval	0x05	5 ms
(MSM6290 only) **Endpoint Descriptor (0x8B)**	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x8B	ep #11, IN
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms
(MSM6290 only) **Endpoint Descriptor (0x07)**	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x07	ep #7, OUT
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	
	bInterval	0x20	32 ms

Table 2-8: UMTS devices (legacy VID/PID)—USB descriptors (Continued)

Descriptor	Field identifier	Value
(C888/C889—Mass Storage enabled) **Interface Descriptor (Number 0x07 Alternate Setting 0x00)**	bLength	0x09
	bDescriptorType	0x04
	bInterfaceNumber	0x07
	bAlternateSetting	0x00
	bNumEndpoints	0x02 Number of endpoints used by this interface is 2, excluding endpoint 0.
	bInterfaceClass	0x08 Interface implements the Mass Storage class
	bInterfaceSubClass	0x06 Interface implements the SCSI Transparent subclass
	bInterfaceProtocol	0x50 Interface uses the Bulk-only protocol
	iInterface	0x00
(C888/C889—Mass Storage enabled) **Endpoint Descriptor (0x08)**	bLength	0x07
	bDescriptorType	0x05
	bEndpointAddress	0x08 ep #8, OUT
	bmAttributes	0x02 Transfer Type: Bulk
	wMaxPacketSize	0x0200
	bInterval	0x00
(C888/C889—Mass Storage enabled) **Endpoint Descriptor (0x8C)**	bLength	0x07
	bDescriptorType	0x05
	bEndpointAddress	0x8C ep #12, IN
	bmAttributes	0x02 Transfer Type: Bulk
	wMaxPacketSize	0x0200
	bInterval	0x00
Manufacturer String Descriptor	bLength	60
	bDescriptorType	0x03
	bString	(Default value) “Sierra Wireless, Incorporated”
Product String Descriptor	bLength	0x14 MC8790, MC8790V, MC8791V, MC8792V, MC8795V SL8080, SL8081, SL8082, SL8083, SL8084, SL8085 SL8090, SL8091, SL8092, SL8093 0x16 AC501, AC504 C888, C889 USB301, USB302
	bDescriptorType	0x03
	bString	“Mini Card” M8790, MC8790V, MC8791V, MC8792V, MC8795V “HSPA Modem” AC501, AC504, C888, C889 USB301, USB302 “SL8080”, “SL8081”, etc. SL8080, SL8081, SL8082, SL8083, SL8084, SL8085 SL8090, SL8091, SL8092, SL8093

Table 2-8: UMTS devices (legacy VID / PID) — USB descriptors (Continued)

Descriptor	Field identifier	Value
String Descriptor 0 (Language Descriptor)	bLength	4
	bDescriptorType	0x03
	wLANGID[0]	0x0409 English—United States
Port Configuration String Descriptor (0xF0)	bLength	0x08
	bDescriptorType	0x03
	bString	(Six consecutive bytes) 0x0031 MUX supported 0x0030 Hybrid not supported 0x0030 NMEA not enabled

MSM6290 descriptors (TRU-Install mode)

Table 2-9: MSM6290 (C888 / C889) (TRU-Install mode) — USB descriptors

Descriptor	Field identifier	Value
Device Descriptor	bLength	0x12
	bDescriptorType	0x01
	bcdUSB	0x0200 USB spec version 2
	bDeviceClass	0x00
	bDeviceSubClass	0x00
	bDeviceProtocol	0x00
	bMaxPacketSize0	0x40 Max packet size for endpoint 0 = 64
	idVendor	0x1199 Sierra Wireless Inc.
	idProduct	0x0FFF
	bcdDevice	0x0000
	iManufacturer	0x02
	iProduct	0x01
	iSerialNumber	0x03
	bNumConfigurations	0x01
Configuration Descriptor (Index 0x00)	bLength	0x09
	bDescriptorType	0x02
	wTotalLength	0x0020
	bNumInterfaces	0x01
	bConfigurationValue	0x01
	iConfiguration	0x00
	bmAttributes	0xE0
	MaxPower	0x00
Interface Descriptor (Number 0x00 Alternate Setting 0x00)	bLength	0x09
	bDescriptorType	0x04
	bInterfaceNumber	0x00
	bAlternateSetting	0x00
	bNumEndpoints	0x02 Number of endpoints used by this interface is 2, excluding endpoint 0.
	bInterfaceClass	0x08 Mass Storage class
	bInterfaceSubClass	0x06 SCSI Transparent subclass
	bInterfaceProtocol	0x50 Bulk-Only protocol
	iInterface	0x00

Table 2-9: MSM6290 (C888/C889) (TRU-Install mode)—USB descriptors

Descriptor	Field identifier	Value	
Endpoint Descriptor (0x01)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x01	ep #1, OUT
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x00	
Endpoint Descriptor (0x81)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x81	ep #1, IN
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms

UMTS devices (legacy VID/PID)—MUX

This section describes MUX mode for UMTS devices using a 'legacy' VID/PID.

Device features	Details
Chipsets	• MDM6200 • MDM6270 • MSM6290 • QSC6270 • For a list of devices using these chipsets, see Table 1-1 on page 15.
Architecture	• Table 2-10, UMTS devices (legacy VID/PID)—MUX endpoints/DLCI assignments, on page 44 • Figure 2-5, UMTS devices (legacy VID/PID)—MUX endpoints/DLCI assignments, on page 46

UMTS devices (legacy VID/PID)—MUX architecture

Table 2-10: UMTS devices (legacy VID/PID)—MUX endpoints/DLCI assignments

Interface	Endpoints	Supported service
	0 (Control)	Management element • Purpose: 0-OUT is used to emulate control line signals. • For details, see "SetControlLineState" in *Universal Serial Bus Class Definitions for Communication Devices, Version 1.1*.

0	**1 IN / OUT (Bulk)**	HIP (carrying CnS), DM, NMEA, AT, Data
		• DLCI Channels:
		  ◦ 0—Control
		  ◦ 1—Primary AT/PPP
		  ◦ 2—DM
		  ◦ 3—HIP (carrying CnS)
		  ◦ 4—NMEA (GPS-enabled devices only)
		  ◦ 5—Data 1 (PDP context 1)
		  ◦ 6—Data 2 (PDP context 2)
		  ◦ 7—Data 3 (PDP context 3)
		Note: NMEA must be enabled by Sierra Wireless, and may not be the default shipping configuration.

1	**2 IN / 2 OUT (Bulk)**	Not used
2	**3 IN / 3 OUT (Bulk)**	Not used
3	**4 IN (Interrupt)**	Not used
	5 IN / 4 OUT (Bulk)	Not used

Table 2-10: UMTS devices (legacy VID/PID) — MUX endpoints / DLCI assignments

Interface	Endpoints	Supported service
4	**6 IN (Interrupt)**	Not used
	7 IN / 5 OUT (Bulk)	Not used
5	**8 IN (Interrupt)**	Not used
	9 IN / 6 OUT (Bulk)	Not used
6	**10 IN (Interrupt)**	Not used
	11 IN / 7 OUT (Bulk)	Not used
7	**12 IN / 8 OUT (Bulk)** *Note: TRU-Install mode uses interface 0.*	(C8XX only) Mass storage (when enabled) • You can access the microSD device in TRU-Install mode (Interface 0) and modem mode (Interface 7). • In TRU-Install mode, the TRU-Install CD-ROM and SD disk are presented as two logical unit numbers (LUNs).

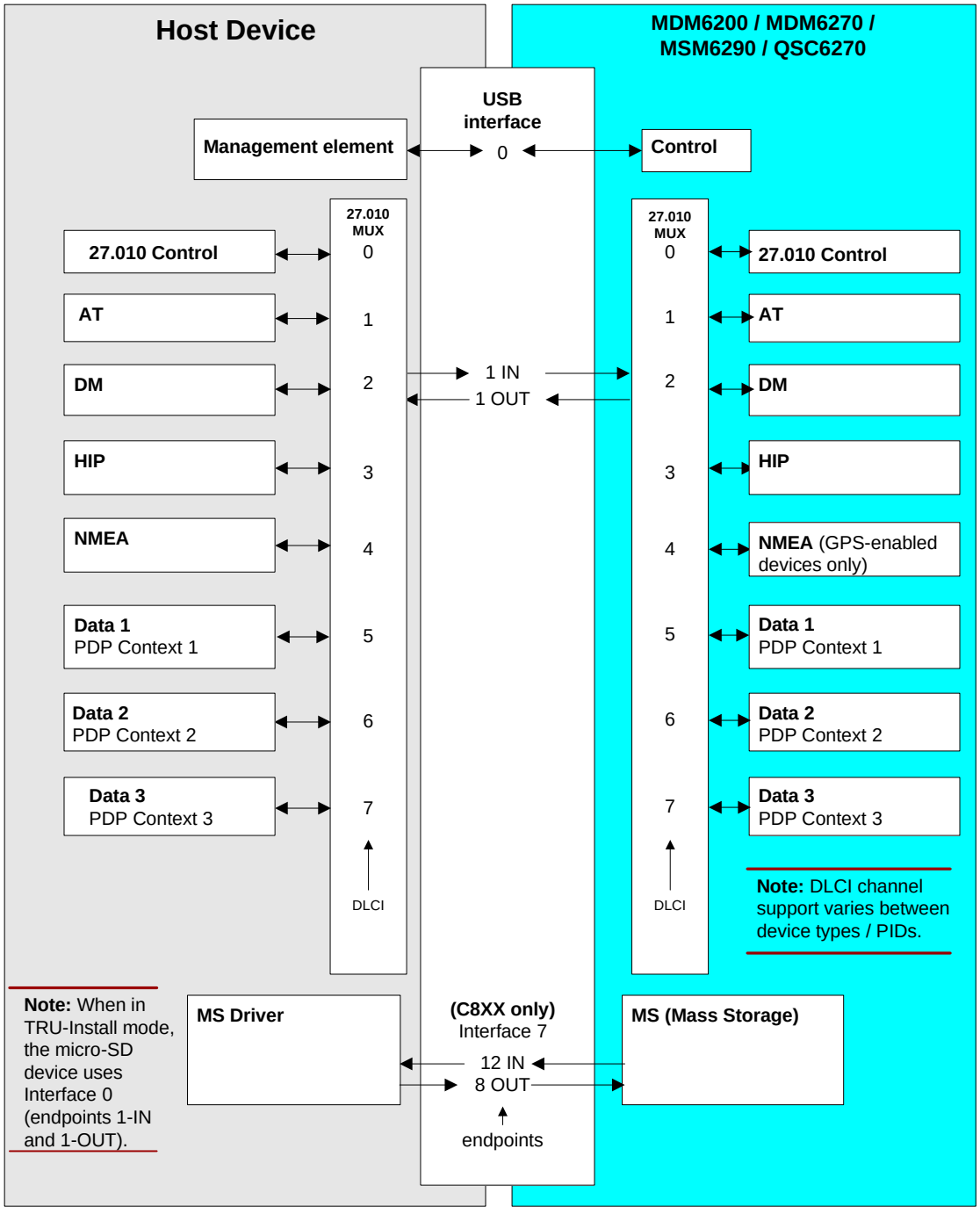

Figure 2-5: UMTS devices (legacy VID/PID)—MUX endpoints/DLCI assignments

UMTS devices (DIP VID/PID)—Non-MUX

This section describes non-MUX mode for UMTS devices using the Direct IP VID/PID.

Device features	Details
Chipsets	• MDM6200 • MDM6270 • MDM8200 • MDM8200A • MDM8220 • MDM9200 • MSM6290 • QSC6270 • For a list of devices using these chipsets, see Table 1-1 on page 15.
Architecture	• Table 2-12, UMTS devices Direct IP—Non-MUX endpoints, on page 49 • Figure 2-6, UMTS devices Direct IP—Non-MUX endpoints, on page 51
Descriptors	• Table 2-13, UMTS devices (DIP VID/PID)—Device/Configuration Descriptors, on page 52 • Table 2-14, UMTS devices (DIP VID/PID)—HIP descriptors (Interface/Endpoint), on page 53 • Table 2-15, UMTS devices (DIP VID/PID)—DM descriptors (Interface/Endpoint), on page 53 • Table 2-16, UMTS devices (DIP VID/PID)—AT descriptors (Interface/Endpoint), on page 54 • Table 2-17, UMTS devices (DIP VID/PID)—MDM1 descriptors (Interface/Endpoint), on page 55 • Table 2-18, UMTS Mass Storage-capable devices (DIP VID/PID)—MS descriptors (Interface/Endpoint), on page 55 • Table 2-19, UMTS devices (DIP VID/PID)—DIP1 descriptors (Interface/Endpoint), on page 56 • Table 2-20, UMTS devices (DIP VID/PID)—DIP2 descriptors (Interface/Endpoint), on page 57 • Table 2-21, UMTS devices (DIP VID/PID)—DIP3 descriptors (Interface/Endpoint), on page 58

Dynamic endpoint assignment for Direct IP VID/PID

The default interface configuration for modems using the Direct IP VID/PID is SKU-dependent and includes some combination of these interfaces:

Interface	Service
0	HIP
1	DM (Diagnostic Monitoring)
2	NMEA
3	AT
4	MDM1 (Modem port 1)
7	DIP1 (Direct IP 1)
9	MS (Mass Storage)
10	DIP2 (Direct IP 2)
11	DIP3 (Direct IP 3)

Typical interface configurations include:

- HIP/DM/AT/MS/DIP1
- HIP/DM/AT/MDM1/DIP1/MS
- HIP/DM/AT/DIP1/DIP2/DIP3/MS

The modem's configuration can be further customized by enabling or disabling the NMEA, DM, or MS interfaces. Because endpoint numbers are assigned dynamically to enabled interfaces, the endpoints associated with a given interface may be different when any of these interfaces are disabled.

Endpoint numbers are assigned sequentially to enabled interfaces, starting with:

- 0x81 for Interrupt or Bulk IN endpoints
- 0x01 for Bulk OUT endpoints

For example, Table 2-11 shows the endpoints assigned to each interface in the configuration HIP/DM/AT/MS/DIP1 when MS is enabled or disabled. When MS is enabled, it uses endpoints 0x85 (5-IN) and 0x04 (4-OUT). However, when MS is disabled, the next interface (DIP1) uses those endpoints.

Table 2-11: Example—Interface-dependent endpoint assignments

	Endpoints (assigned sequentially to enabled services)				
	HIP (Int 0)	**DM (Int 1)**	**AT (Int 3)**	**MS (Int 4)**	**DIP1 (Int 7)**
MS enabled	0x81	0x82	0x83 0x84	0x85	0x86 0x87
	0x01	0x02	0x03	0x04	0x05
MS disabled	0x81	0x82	0x83 0x84	n/a	0x85 0x86
	0x01	0x02	0x03	n/a	0x04

UMTS devices (DIP VID/PID)—Non-MUX architecture

Table 2-12: UMTS devices Direct IP—Non-MUX endpoints[a]

Interface	Endpoint	Supported service
	0 (Control)	Management element • Purpose: 0-OUT is used to emulate control line signals and send encapsulated commands. 0-IN is used to receive encapsulated commands. • For details, see “SetControlLineState”, “SendEncapsulatedCommand”, and “GetEncapsulatedCommand” in *Universal Serial Bus Class Definitions for Communication Devices, Version 1.1*.
0	**IN / OUT (Bulk)**	HIP
1	**IN / OUT (Bulk)**	QUALCOMM DM diagnostics
2	**IN / OUT (Bulk)**	NMEA
3	**IN (Interrupt)**	Notification element • Purpose: Send asynchronous notification of modem status (for example, DSR, DCD, and RI) to the host. • For details, see “SerialState” in *Universal Serial Bus Class Definitions for Communication Devices, Version 1.1*.
	IN / OUT (Bulk)	AT/PPP
4[b]	**IN (Interrupt)**	Notification element • Purpose: Send asynchronous notification of modem status (for example, DSR, DCD, and RI) to the host. • For details, see “SerialState” in *Universal Serial Bus Class Definitions for Communication Devices, Version 1.1*.
	IN / OUT (Bulk)	MDM1 (Modem port 1 (PDP context 1))/PPP (Also called Data1)
7	**IN (Interrupt)**	Notification element • Purpose: Used by Direct IP to send notification indication to host.
	IN / OUT (Bulk)	(ECM) Direct IP (DIP1)
9[c]	**IN / OUT (Bulk)** *Note: TRU-Install mode uses Interface 9 (endpoints 1-IN and 1-OUT).*	(AC3XXU/C888/C889/USB 3XX) Mass storage (when enabled) • You can access the microSD device in both TRU-Install mode and modem mode (Interface 9). • In TRU-Install mode, the TRU-Install CD-ROM and SD disk are presented as two logical unit numbers (LUNs).
10[c]	**IN (Interrupt)**	Notification element • Purpose: Used by Direct IP to send notification indication to host.
	IN / OUT (Bulk)	(ECM) Direct IP (DIP2)

Table 2-12: UMTS devices Direct IP—Non-MUX endpoints[a] (Continued)

Interface	Endpoint	Supported service
11[c]	**IN (Interrupt)**	Notification element • Purpose: Used by Direct IP to send notification indication to host.
	IN / OUT (Bulk)	(ECM) Direct IP (DIP3)

*Note: In these products, each COM port has its own interface; their endpoint assignments are found in the USB interface descriptor. Do not use the **AT!NVPORTMAP** command with these products.*

a. The modem's default interface configuration is SKU-dependent and includes some combination of the listed interfaces.
b. Not available on MDM6270/QSC6270-based devices.
c. Not available on MDM6200/MDM6270/QSC6270-based devices.

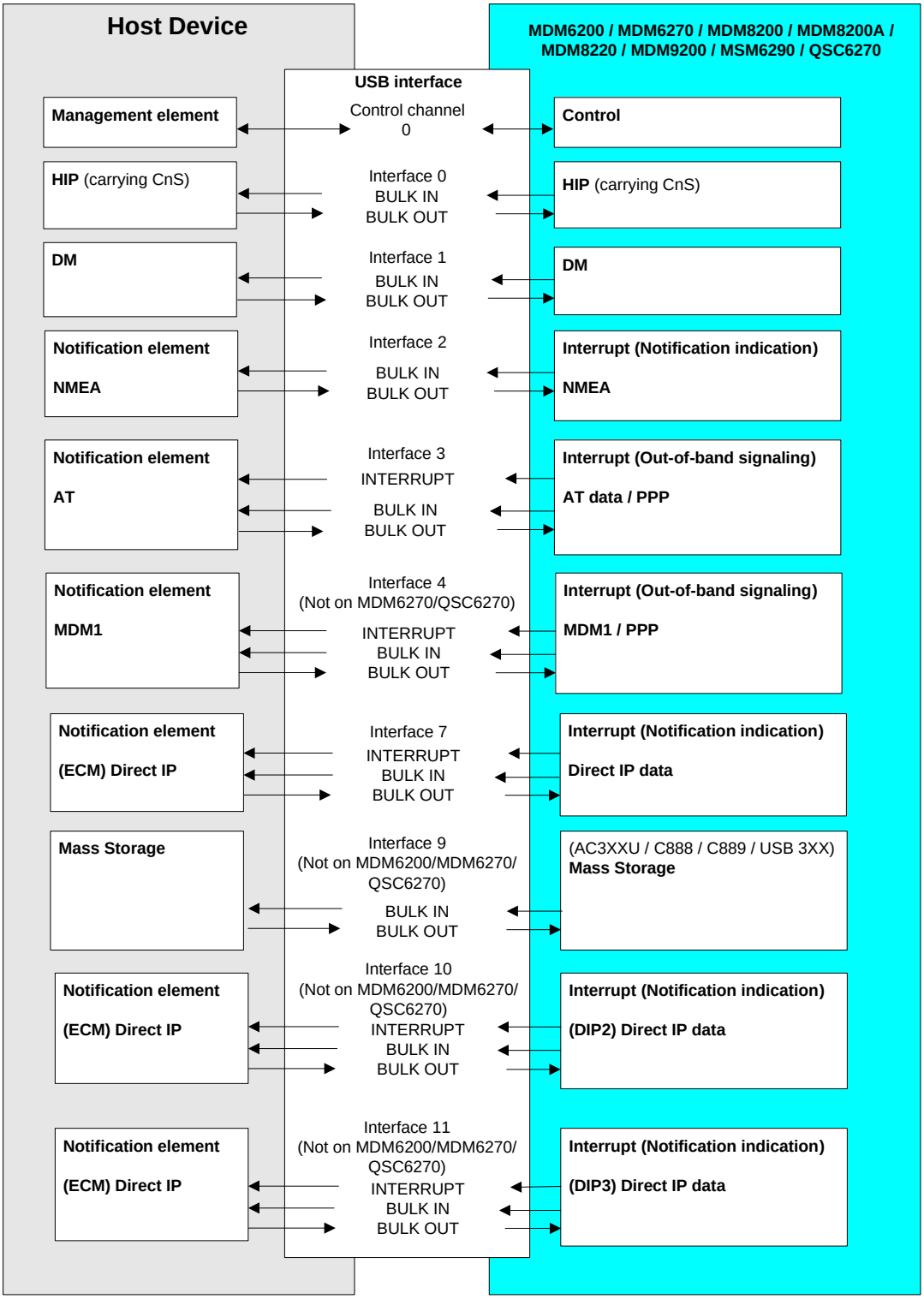

Figure 2-6: UMTS devices Direct IP—Non-MUX endpoints

UMTS devices (DIP VID/PID)—Non-MUX descriptors

Table 2-13: UMTS devices (DIP VID/PID)—Device/Configuration Descriptors

Descriptor	Field identifier	Value
Device Descriptor	bLength	0x12
	bDescriptorType	0x01
	bcdUSB	0x0200 USB spec version 2
	bDeviceClass	0x00
	bDeviceSubClass	0x00
	bDeviceProtocol	0x00
	bMaxPacketSize0	0x40 Max packet size for endpoint 0 = 64
	idVendor	0x1199 Sierra Wireless Inc.
	idProduct	0x68A3 (Standard Direct IP PID) 0x68AA (Carrier-specific Direct IP PID)
	bcdDevice	0x0006
	iManufacturer	0x03[a]
	iProduct	0x02[a]
	iSerialNumber	0x00[a]
	bNumConfigurations	0x01
Configuration Descriptor (Index 0x00)	bLength	0x09
	bDescriptorType	0x02
	wTotalLength	Calculated automatically as the sum of the bLength fields in: • Configuration Descriptor • Each enabled Interface Descriptor and associated Endpoint Descriptors
	bNumInterfaces	0x03–0x08 • Configuration supports 3 to 8 interfaces—Actual value depends on provisioned configuration and state (enabled/disabled) of NMEA, DM, and MS interfaces.
	bConfigurationValue	0x01
	iConfiguration	0x01
	bmAttributes	0xA0 AC501, AC503, AC504, C8XX, USB 30X 0xE0 MC77XX MC87XX SL808X SL809X
	MaxPower	0xFA AC501, AC503, AC504, C8XX, USB 30X 0x00 MC77XX MC87XX SL808X SL809X

a. iManufacturer, iProduct, and iSerial values are SKU-dependent and may vary from the values shown here.

Table 2-14: UMTS devices (DIP VID/PID)—HIP descriptors (Interface/Endpoint)

Descriptor	Field identifier	Value	
Interface Descriptor (Number 0x00 Alternate Setting 0x00)	bLength	0x09	
	bDescriptorType	0x04	
	bInterfaceNumber	0x00	
	bAlternateSetting	0x00	
	bNumEndpoints	0x02	Number of endpoints used by this interface is 2, excluding endpoint 0.
	bInterfaceClass	0xFF	
	bInterfaceSubClass	0xFF	
	bInterfaceProtocol	0xFF	
	iInterface	0x00	
Endpoint Descriptor (0x8*n*—Sequentially assigned, beginning at 0x81 for first enabled interface)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x8*n*	ep #8*n*, IN[a]
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms
Endpoint Descriptor (0x0*n*—Sequentially assigned, beginning at 0x01 for first enabled interface)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x0*n*	ep #*n*, OUT[a]
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms

a. The number of endpoints (*n*) is listed in the **bNumInterfaces field of the Configuration Descriptor (Index 0x00)** on page 52**.**

Table 2-15: UMTS devices (DIP VID/PID)—DM descriptors (Interface/Endpoint)

Descriptor	Field identifier	Value	
Interface Descriptor (Number 0x01 Alternate Setting 0x00)	bLength	0x09	
	bDescriptorType	0x04	
	bInterfaceNumber	0x01	
	bAlternateSetting	0x00	
	bNumEndpoints	0x02	Number of endpoints used by this interface is 2, excluding endpoint 0.
	bInterfaceClass	0xFF	
	bInterfaceSubClass	0xFF	
	bInterfaceProtocol	0xFF	
	iInterface	0x00	
Endpoint Descriptor (0x8*n*—Sequentially assigned, beginning at 0x81 for first enabled interface)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x8*n*	ep #*n*, IN[a]
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms

Table 2-15: UMTS devices (DIP VID/PID) — DM descriptors (Interface / Endpoint) (Continued)

Descriptor	Field identifier	Value	
Endpoint Descriptor (0x0*n* — Sequentially assigned, beginning at 0x01 for first enabled interface)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x0*n*	ep #*n*, OUT[a]
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms

a. The number of endpoints (*n*) is listed in the **bNumInterfaces field of the Configuration Descriptor (Index 0x00)** on page 52**.**

Table 2-16: UMTS devices (DIP VID/PID) — AT descriptors (Interface / Endpoint)

Descriptor	Field identifier	Value	
Interface Descriptor (Number 0x03 Alternate Setting 0x00)	bLength	0x09	
	bDescriptorType	0x04	
	bInterfaceNumber	0x03	
	bAlternateSetting	0x00	
	bNumEndpoints	0x03	Number of endpoints used by this interface is 3, excluding endpoint 0.
	bInterfaceClass	0xFF	
	bInterfaceSubClass	0xFF	
	bInterfaceProtocol	0xFF	
	iInterface	0x00	
Endpoint Descriptor (0x8*n* — Sequentially assigned, beginning at 0x81 for first enabled interface)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x8*n*	ep #*n*, IN[a]
	bmAttributes	0x03	Transfer Type: Interrupt
	wMaxPacketSize	0x0040	
	bInterval	0x05	
Endpoint Descriptor (0x8*n* — Sequentially assigned, beginning at 0x81 for first enabled interface)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x8*n*	ep #*n*, IN[a]
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms
Endpoint Descriptor (0x0*n* — Sequentially assigned, beginning at 0x01 for first enabled interface)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x0*n*	ep #*n*, OUT[a]
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms

a. The number of endpoints (*n*) is listed in the **bNumInterfaces field of the Configuration Descriptor (Index 0x00)** on page 52**.**

Table 2-17: UMTS devices (DIP VID/PID)—MDM1 descriptors (Interface/Endpoint)[a]

Descriptor	Field identifier	Value	
Interface Descriptor (Number 0x04 Alternate Setting 0x00)	bLength	0x09	
	bDescriptorType	0x04	
	bInterfaceNumber	0x04	
	bAlternateSetting	0x00	
	bNumEndpoints	0x03	Number of endpoints used by this interface is 3, excluding endpoint 0.
	bInterfaceClass	0xFF	
	bInterfaceSubClass	0xFF	
	bInterfaceProtocol	0xFF	
	iInterface	0x00	
Endpoint Descriptor (0x8*n*—Sequentially assigned, beginning at 0x81 for first enabled interface)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x8*n*	ep #*n*, IN[b]
	bmAttributes	0x03	Transfer Type: Interrupt
	wMaxPacketSize	0x0040	
	bInterval	0x05	
Endpoint Descriptor (0x8*n*—Sequentially assigned, beginning at 0x81 for first enabled interface)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x8*n*	ep #*n*, IN[b]
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms
Endpoint Descriptor (0x0*n*—Sequentially assigned, beginning at 0x01 for first enabled interface)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x0*n*	ep #*n*, OUT[b]
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms

a. Not supported by MDM6270/QSC6270-based devices.
b. The number of endpoints (*n*) is listed in the **bNumInterfaces field of the Configuration Descriptor (Index 0x00)** on page 52**.**

Table 2-18: UMTS Mass Storage-capable devices (DIP VID/PID)—MS descriptors (Interface/Endpoint)[a]

Descriptor	Field identifier	Value	
Interface Descriptor (Number 0x09 Alternate Setting 0x00)	bLength	0x09	
	bDescriptorType	0x04	
	bInterfaceNumber	0x09	
	bAlternateSetting	0x00	
	bNumEndpoints	0x02	Number of endpoints used by this interface is 2, excluding endpoint 0.
	bInterfaceClass	0x08	Mass Storage class
	bInterfaceSubClass	0x06	SCSI Transparent subclass
	bInterfaceProtocol	0x50	Bulk-Only protocol
	iInterface	0x00	

Table 2-18: UMTS Mass Storage-capable devices (DIP VID / PID) — MS descriptors (Interface / Endpoint)[a] (Continued)

Descriptor	Field identifier	Value	
Endpoint Descriptor (0x0*n* — Sequentially assigned, beginning at 0x01 for first enabled interface)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x0*n*	ep #*n*, OUT[b]
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms
Endpoint Descriptor (0x8*n* — Sequentially assigned, beginning at 0x81 for first enabled interface)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x8*n*	ep #*n*, IN[b]
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms

a. Not supported by MDM6200/MDM6270/QSC6270-based devices.
b. The number of endpoints (*n*) is listed in the **bNumInterfaces field of the Configuration Descriptor (Index 0x00)** on page 52**.**

Table 2-19: UMTS devices (DIP VID / PID) — DIP1 descriptors (Interface / Endpoint)

Descriptor	Field identifier	Value	
Interface Descriptor (Number 0x07 Alternate Setting 0x00)	bLength	0x09	
	bDescriptorType	0x04	
	bInterfaceNumber	0x07	
	bAlternateSetting	0x00	
	bNumEndpoints	0x03	Number of endpoints used by this interface is 3, excluding endpoint 0.
	bInterfaceClass	0xFF	
	bInterfaceSubClass	0xFF	
	bInterfaceProtocol	0xFF	
	iInterface	0x00	
Endpoint Descriptor (0x8*n* — Sequentially assigned, beginning at 0x81 for first enabled interface)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x8*n*	ep #*n*, IN[a]
	bmAttributes	0x03	Transfer Type: Interrupt
	wMaxPacketSize	0x0040	
	bInterval	0x05	
Endpoint Descriptor (0x8*n* — Sequentially assigned, beginning at 0x81 for first enabled interface)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x8*n*	ep #*n*, IN[a]
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms

Table 2-19: UMTS devices (DIP VID/PID)—DIP1 descriptors (Interface/Endpoint) (Continued)

Descriptor	Field identifier	Value	
Endpoint Descriptor (0x0*n*—Sequentially assigned, beginning at 0x01 for first enabled interface)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x0*n*	ep #*n*, OUT[a]
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms

a. The number of endpoints (*n*) is listed in the **bNumInterfaces field of the Configuration Descriptor (Index 0x00)** on page 52**.**

Table 2-20: UMTS devices (DIP VID/PID)—DIP2 descriptors (Interface/Endpoint)[a]

Descriptor	Field identifier	Value	
Interface Descriptor (Number 0x0A Alternate Setting 0x00)	bLength	0x09	
	bDescriptorType	0x04	
	bInterfaceNumber	0x0A	
	bAlternateSetting	0x00	
	bNumEndpoints	0x03	Number of endpoints used by this interface is 3, excluding endpoint 0.
	bInterfaceClass	0xFF	
	bInterfaceSubClass	0xFF	
	bInterfaceProtocol	0xFF	
	iInterface	0x00	
Endpoint Descriptor (0x8*n*—Sequentially assigned, beginning at 0x81 for first enabled interface)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x8*n*	ep #*n*, IN[b]
	bmAttributes	0x03	Transfer Type: Interrupt
	wMaxPacketSize	0x0040	
	bInterval	0x05	
Endpoint Descriptor (0x8*n*—Sequentially assigned, beginning at 0x81 for first enabled interface)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x8*n*	ep #*n*, IN[b]
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms
Endpoint Descriptor (0x0*n*—Sequentially assigned, beginning at 0x01 for first enabled interface)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x0*n*	ep #*n*, OUT[b]
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms

a. Not supported by MDM6200/MDM6270/QSC6270-based devices.
b. The number of endpoints (*n*) is listed in the **bNumInterfaces field of the Configuration Descriptor (Index 0x00)** on page 52**.**

Table 2-21: UMTS devices (DIP VID / PID) — DIP3 descriptors (Interface / Endpoint)[a]

Descriptor	Field identifier	Value	
Interface Descriptor (Number 0x0B Alternate Setting 0x00)	bLength	0x09	
	bDescriptorType	0x04	
	bInterfaceNumber	0x0B	
	bAlternateSetting	0x00	
	bNumEndpoints	0x03	Number of endpoints used by this interface is 3, excluding endpoint 0.
	bInterfaceClass	0xFF	
	bInterfaceSubClass	0xFF	
	bInterfaceProtocol	0xFF	
	iInterface	0x00	
Endpoint Descriptor (0x8*n* — Sequentially assigned, beginning at 0x81 for first enabled interface)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x8*n*	ep #*n*, IN[b]
	bmAttributes	0x03	Transfer Type: Interrupt
	wMaxPacketSize	0x0040	
	bInterval	0x05	
Endpoint Descriptor (0x8*n* — Sequentially assigned, beginning at 0x81 for first enabled interface)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x8*n*	ep #*n*, IN[b]
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms
Endpoint Descriptor (0x0*n* — Sequentially assigned, beginning at 0x01 for first enabled interface)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x0*n*	ep #*n*, OUT[b]
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms

a. Not supported by MDM6200/MDM6270/QSC6270-based devices.
b. The number of endpoints (*n*) is listed in the **bNumInterfaces field of the Configuration Descriptor (Index 0x00)** on page 52**.**

UMTS devices (QMI VID/PID)—Non-MUX

This section describes non-MUX mode for UMTS devices using a QMI VID/PID.

Chipsets	• MDM9200 • MDM9600 • For a list of devices using these chipsets, see Table 1-1 on page 15.
Architecture	• Table 2-22, UMTS devices (QMI VID/PID)—Non-MUX endpoints, on page 59 • Figure 2-7, UMTS devices (QMI VID/PID)—Non-MUX endpoints, on page 60
Descriptors	• Table 2-23, UMTS devices (QMI VID/PID)—USB descriptors, on page 61

UMTS devices (QMI VID/PID) non-MUX architecture

Table 2-22: UMTS devices (QMI VID/PID)—Non-MUX endpoints

Interface	Endpoint	Supported service
	0 (Control)	Management element • Purpose: 0-OUT is used to emulate control line signals. • For details, see “SetControlLineState” in the “Universal Serial Bus Class Definitions for Communication Devices” document.
0	**1 IN / OUT (Bulk)**	QUALCOMM DM diagnostics
2	**2 IN / OUT (Bulk)**	NMEA
3	**3 IN (Interrupt)**	Notification element • Purpose: Send asynchronous notification of modem status (for example, DSR, DCD, and RI) to the host. • For details, see “SerialState” in *Universal Serial Bus Class Definitions for Communication Devices, Version 1.1*.
	4 IN / 3 OUT (Bulk)	AT (MDM) Modem port providing full-featured AT command support.
8	**5 IN (Interrupt)**	Notification element • Purpose: Used to send notification indications for to host.
	6 IN / 4 OUT (Bulk)	NET QMI network interface

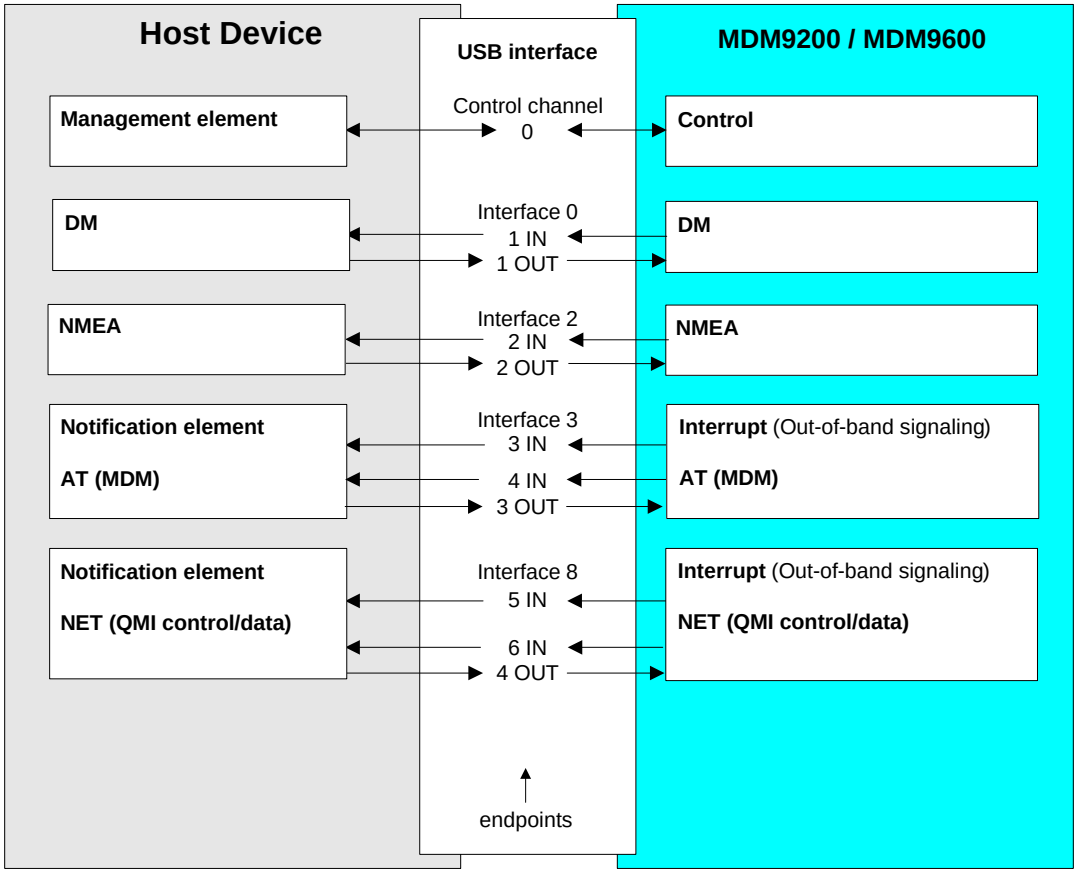

Figure 2-7: UMTS devices (QMI VID/PID)—Non-MUX endpoints

UMTS devices (QMI VID/PID) non-MUX descriptors

Table 2-23: UMTS devices (QMI VID/PID)—USB descriptors

Descriptor	Field identifier	Value	
Device Descriptor	bLength	0x12	
	bDescriptorType	0x01	
	bcdUSB	0x0200	USB spec version 2
	bDeviceClass	0x00	
	bDeviceSubClass	0x00	
	bDeviceProtocol	0x00	
	bMaxPacketSize0	0x40	Max packet size for endpoint 0 = 64
	idVendor	0x1199	Sierra Wireless Inc.
	idProduct	0x68A2	
	bcdDevice	0x0006	
	iManufacturer	0x03	
	iProduct	0x02	
	iSerialNumber	0x04	
	bNumConfigurations	0x01	
Configuration Descriptor (Index 0x00)	bLength	0x09	
	bDescriptorType	0x02	
	wTotalLength	0x0073	
	bNumInterfaces	0x04	
	bConfigurationValue	0x01	
	iConfiguration	0x01	
	bmAttributes	0xE0	
	MaxPower	0x00	
Interface Descriptor (Number 0x00 Alternate Setting 0x00)	bLength	0x09	
	bDescriptorType	0x04	
	bInterfaceNumber	0x00	
	bAlternateSetting	0x00	
	bNumEndpoints	0x02	Number of endpoints used by this interface is 2, excluding endpoint 0.
	bInterfaceClass	0xFF	Vendor-specific device class
	bInterfaceSubClass	0xFF	Vendor-specific device subclass
	bInterfaceProtocol	0xFF	
	iInterface	0x00	
Endpoint Descriptor (0x81)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x81	ep #1, IN
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms

Table 2-23: UMTS devices (QMI VID / PID) — USB descriptors (Continued)

Descriptor	Field identifier	Value	
Endpoint Descriptor (0x01)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x01	ep #1, OUT
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms
Interface Descriptor (Number 0x02 Alternate Setting 0x00)	bLength	0x09	
	bDescriptorType	0x04	
	bInterfaceNumber	0x02	
	bAlternateSetting	0x00	
	bNumEndpoints	0x02	Number of endpoints used by this interface is 2, excluding endpoint 0.
	bInterfaceClass	0xFF	Vendor-specific device class
	bInterfaceSubClass	0xFF	Vendor-specific device subclass
	bInterfaceProtocol	0xFF	
	iInterface	0x00	
Endpoint Descriptor (0x82)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x82	ep #2, IN
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms
Endpoint Descriptor (0x02)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x02	ep #2, OUT
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms
Interface Descriptor (Number 0x03 Alternate Setting 0x00)	bLength	0x09	
	bDescriptorType	0x04	
	bInterfaceNumber	0x03	
	bAlternateSetting	0x00	
	bNumEndpoints	0x03	Number of endpoints used by this interface is 3, excluding endpoint 0.
	bInterfaceClass	0xFF	Vendor-specific device class
	bInterfaceSubClass	0xFF	Vendor-specific device subclass
	bInterfaceProtocol	0xFF	
	iInterface	0x00	
Endpoint Descriptor (0x83)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x83	ep #3, IN
	bmAttributes	0x03	Transfer Type: Interrupt
	wMaxPacketSize	0x0040	
	bInterval	0x05	5 ms

Table 2-23: UMTS devices (QMI VID/PID)—USB descriptors (Continued)

Descriptor	Field identifier	Value	
Endpoint Descriptor (0x04)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x84	ep #4, IN
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms
Endpoint Descriptor (0x03)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x03	ep #3, OUT
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms
Interface Descriptor (Number 0x08 Alternate Setting 0x00)	bLength	0x09	
	bDescriptorType	0x04	
	bInterfaceNumber	0x08	
	bAlternateSetting	0x00	
	bNumEndpoints	0x03	Number of endpoints used by this interface is 3, excluding endpoint 0.‘
	bInterfaceClass	0xFF	
	bInterfaceSubClass	0xFF	
	bInterfaceProtocol	0xFF	
	iInterface	0x00	
Endpoint Descriptor (0x85)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x85	ep #5, IN
	bmAttributes	0x03	Transfer Type: Interrupt
	wMaxPacketSize	0x0040	
	bInterval	0x05	5 ms
Endpoint Descriptor (0x86)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x86	ep #6, IN
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms
Endpoint Descriptor (0x04)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x04	ep #4, OUT
	bmAttributes	0x02	Transfer Type: Bulk
	wMaxPacketSize	0x0200	Must be high speed device
	bInterval	0x20	32 ms

3: Services

This chapter describes the services (logical protocols) supported by the AirCard and AirPrime devices for the transfer of data and control information. These services include (dependent on device type):

- Control on page 65
- Interrupt on page 65
- AT/PPP on page 66
- HIP (Host Interface Protocol) on page 67
- CnS (Control and Status) on page 67
- DM (QUALCOMM Diagnostic Monitoring) on page 68
- NMEA (National Marine Electronics Association) on page 69
- Data 1, Data 2, Data 3 (Additional PPP) on page 70
- Mass Storage on page 71
- Direct IP on page 71

USB protocols

Control

The standard control channel (management element) is implemented on endpoint 0 (as required by the USB and ACM specifications). This supports the use of selected USB interface and power management, ACM, and Sierra Wireless' proprietary commands and requests. For a list of supported CDC / ACM messages, see Supported CDC USB commands on page 101.

Interrupt

USB interrupt channels (notification elements) are implemented on the following endpoints:

- MDM6200/MDM6270/QSC6270 (using legacy VID/PID)—4, 6
- MSM6290 (using legacy VID/PID)—4, 6, 8, 10
- All UMTS devices with Direct IP VID/PID—Endpoints vary depending on modem configuration. Two interrupt channels are used for the AT and ECM services.
- All other modules—1

This is part of the typical Communication Interface Class for a CDC device and supports the notifications in the ACM specification and proprietary notifications defined by Sierra Wireless.

Data services

AT/PPP

The AT/PPP data channel is implemented using the endpoints listed in Table 3-1 for devices using the specified chipsets.

Table 3-1: AT/PPP data channel endpoints (by chipset and mode)

Chipset[a]	Non-MUX endpoints		MUX endpoints		Hybrid-MUX endpoints	
	Data	Flow control	Data	Flow control	Data	Flow control
MDM6085 **QSC6085**	(Interface 0) 2-IN 2-OUT	1-IN	(Interface 0; 2-IN/2-OUT) DLCI-1	DLCI-0	(Interface 0) 2-IN 2-OUT	1-IN
(Legacy VID/PID) **MDM6200** **MDM6270** **MSM6290** **QSC6270**	(Interface 3) 5-IN 4-OUT	4-IN	(Interface 0; 1-IN/1-OUT) DLCI-1	DLCI-0	N/A	
(Direct IP VID/PID) **MDM6200** **MDM6270** **MDM8200** **MDM8200A** **MDM8220** **MDM9200** **MSM6290** **QSC6270**	(Interface 3) IN/OUT (Bulk)	IN (Interrupt)	N/A		N/A	
	Note: Endpoints vary depending on which services have been implemented.					
(QMI VID/PID) **MDM9200** **MDM9600**	(Interface 3) 4-IN 3-OUT	3-IN	N/A		N/A	
	Note: Interface 3 is a MDM interface featuring full AT command support.					

a. For a list of devices using these chipsets, see Table 1-1 on page 15.

Note: The Heatherington escape method (TIES) is not supported.

The AT/PPP data channel supports the transfer of AT commands for module control, and TCP/IP (or other transport protocol) data via PPP for network communication.

For lists of supported AT commands, see:

- CDMA devices
  - *AT Command Set for User Equipment (UE) (Release 6)*
  - *AT Command Reference (Document 2130620)*

- *Extended AT Command Reference (Document 2130621) (for AirCards and MiniCards)*
- *CDMA SL501x AT Command Reference (Document 4110801) (for SL-series devices)*

- UMTS devices
  - *AirCard/AirPrime UMTS Devices Supported AT Command Reference (Document 2130617)*

HIP (Host Interface Protocol)

HIP is a proprietary protocol developed to multiplex different information types over a single interface (for example, carrying CnS commands and DM logging data).

The HIP data channel is implemented using the endpoints listed in Table 3-2 for devices using the specified chipsets.

Table 3-2: HIP data channel endpoints (by chipset and mode)

Chipset[a]	Non-MUX endpoints		MUX endpoints		Hybrid-MUX endpoints	
	Data	Flow control	Data	Flow control	Data	Flow control
MDM6085 **QSC6085**	(Interface 0) 4-IN 4-OUT	—	(Interface 0; 2-IN/2-OUT) DLCI-3	DLCI-0	(Interface 0; 4-IN/4-OUT) DLCI-3	1-IN
(Legacy VID/PID) **MDM6200** **MDM6270** **MSM6290** **QSC6270**	(Interface 0) 1-IN 1-OUT	—	(Interface 0; 1-IN/1-OUT) DLCI-3	DLCI-0	N/A	
(Direct IP VID/PID) **MDM6200** **MDM6270** **MDM8200** **MDM8200A** **MDM8220** **MDM9200** **MSM6290** **QSC6270**	(Interface 0) IN/OUT (Bulk)	—	N/A		N/A	
	Note: Endpoints vary depending on which services have been implemented.					

a. For a list of devices using these chipsets, see Table 1-1 on page 15.

For details on the HIP protocol, refer to *CDMA 1xEV-DO CnS Reference (Document 2130754), "HIP Basics"* chapter, or *AirPrime UMTS MC and SL Series CnS Reference (Document 2130602), Chapter 3: "HIP Basics"*.

CnS (Control and Status)

CnS objects (commands) are used to manage the module's configuration, and to provide status reporting and monitoring of the device.

These objects are transported between the host and module by embedding them in HIP messages (see HIP (Host Interface Protocol) on page 67).

For details on CnS objects, refer to *CDMA 1xEV-DO CnS Reference (Document 2130754)* or *AirPrime UMTS MC and SL Series CnS Reference (Document 2130602)*.

DM (QUALCOMM Diagnostic Monitoring)

The QUALCOMM DM logging data provides information about the operation of the wireless protocol stack, and other diagnostic information for firmware developers. It can be used for configuration, logging, testing, and debugging of the embedded module.

For details on implementing protocol stack logging to access QUALCOMM DM information, refer to *AirPrime UMTS MC and SL Series CnS Reference (Document 2130602)*.

The DM data channel is implemented using the endpoints listed in Table 3-3 for devices using the specified chipsets.

For additional information on logging DM data, see QUALCOMM DM Data Logging on page 103.

Table 3-3: DM data channel endpoints (by chipset and mode)

Chipset[a]	Non-MUX endpoints		MUX endpoints		Hybrid-MUX endpoints	
	Data	Flow control	Data	Flow control	Data	Flow control
MDM6085 **QSC6085**	(Interface 0) 8-IN 8-OUT	—	(Interface 0) 2-IN/2-OUT DLCI-2	DLCI-0	(Interface 0) 4-IN/4-OUT DLCI-2	—
(Legacy VID/PID) **MDM6200** **MDM6270** **MSM6290** **QSC6270**	(Interface 1) 2-IN 2-OUT	—	(Interface 0) 1-IN/1-OUT DLCI-2	—	N/A	
(Direct IP VID/PID) **MDM6200** **MDM6270** **MDM8200** **MDM8200A** **MDM8220** **MDM9200** **MSM6290** **QSC6270**	(Interface 1) IN/OUT (Bulk)	—	N/A		N/A	
	Note: Endpoints vary depending on which services have been implemented.					
(QMI VID/PID) **MDM9200** **MDM9600**	(Interface 0) 1-IN 1-OUT	—	N/A		N/A	

a. For a list of devices using these chipsets, see Table 1-1 on page 15.

NMEA (National Marine Electronics Association)

The NMEA data channel is used to transfer GPS receiver data (NMEA-0183 formatted ‘sentences’) between the module and the host for use in Location-Based Services (LBS) applications.

The NMEA data channel is implemented using the endpoints listed in Table 3-4 for GPS-enabled devices using the specified chipsets.

Table 3-4: NMEA data channel endpoints (by chipset and mode)

Chipset[a]	Non-MUX endpoints		MUX endpoints		Hybrid-MUX endpoints	
	Data	Flow control	Data	Flow control	Data	Flow control
MDM6085 **QSC6085**	(Interface 0) 5-IN 5-OUT	—	(Interface 0) 2-IN/2-OUT DLCI-4	DLCI-0	(Interface 0) 4-IN/4-OUT DLCI-4	1-INS
(Legacy VID/PID) **MDM6200** **MDM6270** **MSM6290** **QSC6270**	(Interface 2) 3-IN 3-OUT	—	(Interface 0) 1-IN/1-OUT DLCI-4	DLCI-0	N/A	
(Direct IP VID/PID) **MDM6200** **MDM6270** **MDM8200** **MDM8200A** **MDM8220** **MDM9200** **MSM6290** **QSC6270**	(Interface 2) IN/OUT (Bulk)	—	N/A		N/A	
	Note: Endpoints vary depending on which services have been implemented.					
(QMI VID/PID) **MDM9200** **MDM9600**	(Interface 2) 2-IN 2-OUT	—	N/A		N/A	

a. For a list of devices using these chipsets, see Table 1-1 on page 15.

Note: The modem’s default shipping configuration may include NMEA support. For those modems that support NMEA, the default NMEA channel is ON for MDM6085- and QSC6085-based devices, and OFF for all others.

Data 1, Data 2, Data 3 (Additional PPP)

The Data 1, Data 2, and Data 3 data channels are implemented using the endpoints listed in Table 3-5 for devices using the specified chipsets.

Table 3-5: Data 1/Data 2/Data 3 data channel endpoints (by chipset and mode)

Chipset[a]	Non-MUX endpoints		MUX endpoints	
	Data	Flow control	Data	Flow control
(Legacy VID/PID) **MSM6290**	(Interface 4) Data 1: 7-IN/5-OUT (Interface 5) Data 2: 9-IN/6-OUT (Interface 6) Data 3: 11-IN/7-OUT	(Interface 4) Data 1: 6-IN (Interface 5) Data 2: 8-IN (Interface 6) Data 3: 10-IN	(Interface 0) 1-IN/1-OUT Data 1: DLCI-5 Data 2: DLCI-6 Data 3: DLCI-7	DLCI-0
(Direct IP VID/PID) **MDM8200** **MDM8200A** **MDM8220** **MDM9200** **MSM6290**	(Interface 4) MDM1 (Data 1): IN/OUT (Bulk)	(Interface 4) IN	N/A	
	Note: Endpoints vary depending on which services have been implemented.			

a. For a list of devices using these chipsets, see Table 1-1 on page 15.

These channels are used to establish additional data connections to the remote network. These connections may have one or more of the following properties:

- A different quality of service than the other connections
- A different APN (Access Point Name)

Limitations of these channels include:

- ATDT and ATI are the only supported AT commands

Mass Storage

(C888/C889/USB 306/USB 307/AC USB 308/AC USB 309/AC 310U)

In modem mode (non-MUX, MUX), the module can be configured to add a mass storage device using the endpoints listed in Table 3-6 for devices using the specified chipsets.

Table 3-6: Mass Storage data channel endpoints (by chipset and mode)

Chipset[a]	Modem mode endpoints (Non-MUX, MUX)	TRU-Install mode endpoints
(Legacy VID/PID) **MSM6290[b]**	Interface 7 12-IN 8-OUT	Interface 0 1-IN 1-OUT
(Direct IP VID/PID) **MDM8200** **MDM8200A** **MDM8220** **MDM9200** **MSM6290[a]**	Interface 9 IN/OUT (Bulk) *Note: Endpoints vary depending on which services have been implemented.*	Interface 9 1-IN 1-OUT

a. For a list of devices using these chipsets, see Table 1-1 on page 15.
b. Device must be configured to support Mass Storage.

Note: You can access the microSD device in both the TRU-Install and modem modes. In TRU-Install mode, the TRU-Install CD-ROM and SD disk are presented as two logical unit numbers (LUNs).

Direct IP

(UMTS devices using Direct IP VID/PID)

Direct IP mode provides a means of achieving higher throughput rates and improved CPU efficiency on HSPA+ networks compared to data transfers using PPP. The Sierra Wireless driver achieves this by transporting IP packets over USB without using PPP.

Direct IP is available in non-Mux mode as described in Table 3-7 on page 72.

Endpoints are dynamically assigned based on the interfaces that are enumerated.

For details about Direct IP mode on Sierra Wireless modems, see Direct IP mode on page 105.

Table 3-7: Direct IP data channel endpoints (by chipset)

Chipset[a]	Non-MUX interfaces
(Direct IP VID/PID) **MDM6200**[b] **MDM6270**[b] **MDM8200** **MDM8200A** **MDM8220** **MDM9200** **MSM6290** **QSC6270**[b]	DIP1—Interface 7[c] DIP2—Interface 10 DIP3—Interface 11 IN/OUT (Bulk) IN (Notification) *Note: Endpoints vary depending on which services have been implemented.*

a. For a list of devices using these chipsets, see Table 1-1 on page 15.
b. DIP2/DIP3 are not supported by MDM6200, MDM6270, QSC6270.
c. DIP1 is the default interface for configurations supporting multiple DIP interfaces.

QMI (NET interface)

(UMTS devices using QMI VID/PID)

Devices operating in QMI mode (Qualcomm's propietary framework) use the NET interface to transfer data packets over USB rather than using PPP.

QMI is available in non-Mux mode as described in Table 3-8 on page 72.

For details about QMI support on Sierra Wireless modems, see the *Linux QMI SDK Application Developer's Guide*.

Table 3-8: QMI data channel endpoints (by chipset)

Chipset[a]	Non-MUX interfaces
(QMI VID/PID) **MDM9200** **MDM9600**	NET—Interface 8 6-IN/4-OUT (Bulk) 5-IN (Notification)

a. For a list of devices using these chipsets, see Table 1-1 on page 15.

A: Vendor-specific commands

A

The modem's state can be controlled using vendor-specific commands over the USB control interface as an alternative to using AT commands.

As described in *Universal Serial Bus Specification, Rev 2.0*, each control transfer includes the following:

- Setup stage (host to module)
- An optional Data stage (information being transferred host to module, or module to host)
- Status stage (opposite direction from the previous stage).

Command availability

Table A-1: Command availability by chipset[a]

Command / Description	MDM6085 QSC6085	(Legacy VID/PID) MDM6200 MDM6270 MSM6290 QSC6270	(Direct IP VID/PID) MDM6200 MDM6270 MDM8200 MDM8200A MDM8220 MDM9200 MSM6290 QSC6270	(QMI VID/PID) MDM9200 MDM9600
Set Device Power State on page 75 Powers the modem on or off	✔	✔	✔	✔
Set Mode Non-MUX on page 76 Tears down the 27.010 MUX protocol on the selected channel	✔	✔	✘	✘
Set Mode MUX on page 77 Sets up the 27.010 MUX protocol on the selected channel	✔	✔	✘	✘
Get Mode MUX on page 79 Tells the module to return the 27.010 MUX protocol status for a selected channel	✔	✔	✘	✘
Get NDIS Support on page 81 Tells the module to indicate if the NDIS driver is supported	✔	✔	✔	✔
Get NDIS Preference on page 82 Tells the module to indicate if the NDIS driver is established or disabled	✔	✔	✔	✔
Get Attributes on page 83 Query the attributes of the firmware image	✔	✔[b]	✔	✔

Table A-1: Command availability by chipset[a] (Continued)

Command / Description	MDM6085 QSC6085	(Legacy VID/PID) MDM6200 MDM6270 MSM6290 QSC6270	(Direct IP VID/PID) MDM6200 MDM6270 MDM8200 MDM8200A MDM8220 MDM9200 MSM6290 QSC6270	(QMI VID/PID) MDM9200 MDM9600
Set Mode NMEA on page 85 Start or stop the NMEA channel	✔	✘	✘	✔
Get Mode NMEA on page 86 Query the current status of the NMEA channel	✔	✘	✘	✘
Set Host Power State on page 87 Indicate the current power state of the host device	✔	✔	✔	✔
Get TRU-Install Information on page 88 Indicate which software installation packages are stored on the module	✔[c]	✔	✔[d]	✔[d]
Set TRU-Install Mode on page 90 Request mode of operation (mass-storage device or modem)	✔[c]	✔	✔[d]	✔[d]
Get Config Item on page 91	✔	✔	✔[e]	✔
Set Config Item on page 93	✔	✔	✔[e]	✔
Set Device Reset on page 95	✔	✔	✔[f]	✔
Set Host Info on page 96	✘	✔	✘[g] ✔[h]	✘

a. For a list of devices using these chipsets, see Table 1-1 on page 15.
b. MC devices only
c. All devices except AC595/AC595U
d. AC/USB devices only; not supported by MC/SL devices
e. MDM9200 - TRU-Install and SD only
f. MDM9200
g. MDM6200, MDM8200, MDM8220, MDM9200
h. MDM6270/QSC6270, MDM8200A, MSM6290

Note: There are no commands available in boot-and-hold mode for UMTS devices using Direct IP VID/PID.

In boot-and-hold mode, only the following commands are supported:

- CDMA devices:
  - Get Attributes on page 83
- GSM / UMTS devices:
  - Get Mode MUX on page 79
  - Set Mode MUX on page 77
  - Get NDIS Support on page 81
  - Get NDIS Preference on page 82
  - Get Attributes on page 83

Set Device Power State

Power the modem on or off.

Stages:

- Setup—Host tells modem to power on or off.
- Status(IN)—Modem indicates completion of requested operation.

Setup stage

Direction: Host to modem

Purpose: Tell modem to power on or off.

Details: The following table describes the outgoing packet structure.

Table A-2: Set Device Power State—Setup stage

Offset	Size (bytes)	Field	Value
0	1	Request type	0x40
1	1	Request	0x00
2	2	Value	State • 0x0000 (D0 state, modem power on) • 0x0003 (D3 state, modem power off)
4	2	Index	0x0000
6	2	Length	0x0000

Status stage (IN)

Direction: Modem to host

Purpose: Notify host that the requested operation has completed.

Details: Modem returns a zero-length packet when the requested operation has completed.

Set Mode Non-MUX

Tear down the 27.010 MUX protocol on a selected channel.

Stages:

- Setup—Host tells modem to use non-MUX protocol on a specific channel.
- Status (IN)—Modem indicates completion of requested operation.

Setup stage

Direction: Host to modem

Purpose: Tell modem to use non-MUX protocol on a specific channel.

Details: The following table describes the outgoing packet structure.

Table A-3: Set Mode Non-MUX—Setup stage

Offset	Size (bytes)	Field	Value
0	1	Request type	0x40
1	1	Request	0x01
2	2	Value	0x0000 (not used)
4	2	Index	Channel selection • 0x0000 Data • 0x0001 Control • 0x0002 NMEA
6	2	Length	0x0000 (not used)

Status stage (IN)

Direction: Modem to host

Purpose: Notify host that the requested operation has completed.

Details: Modem returns a zero-length packet when the requested operation has completed.

Set Mode MUX

Set up the 27.010 MUX protocol on a selected channel.

Stages:

- Setup—Host tells modem to use MUX protocol on a specific channel.
- Data (OUT)—Host indicates MUX parameters to be used.
- Status (IN)—Modem indicates completion of requested operation.

Note: The modem moves from "Non-MUX" mode to "Transition" mode when it receives the Set Mode MUX request. The modem then moves from "Transition" mode to "MUX" mode when it is ready to receive MUX (27.010) framed messages on the requested interface.

Setup stage

Direction: Host to modem

Purpose: Tell modem to use MUX protocol on a specific channel.

Details: The following table describes the outgoing packet structure.

Table A-4: Set Mode MUX—Setup stage

Offset	Size (bytes)	Field	Value
0	1	Request type	0x40
1	1	Request	0x02
2	2	Value	0x0000 (not used)
4	2	Index	Channel selection • 0x0000 Data • 0x0001 Control (Places all CDMA devices into Hybrid-MUX mode.) • 0x0002 NMEA
6	2	Length	0x0012 (length of data in Data stage)

Data stage (OUT)

Direction: Host to modem

Purpose: Indicate MUX mode parameters.

Details: The following table describes the outgoing packet structure.

Table A-5: Set Mode MUX — Data stage (OUT)

Offset	Size (bytes)	Field	Value
0	2	Mode	See *Terminal Equipment to User Equipment (TE-UE) multiplexer protocol (Release 6)*
2	2	Frame type	
4	2	Port speed	
6	2	Maximum frame size	
8	2	ACK timer	
10	2	Maximum number of retransmissions	
12	2	Response timer	
14	2	Wakeup timer	
16	2	Window size	

Status stage (IN)

Direction: Modem to host

Purpose: Notify host that the requested operation has completed.

Details: Modem returns a zero-length packet when the requested operation has completed.

Get Mode MUX

Request the current 27.010 MUX protocol status of a specific channel.

Stages:

- Setup—Host tells modem to report the status of a specific channel.
- Data (IN)—Modem indicates the channel's current protocol status.
- Status (OUT)—Host indicates that it received the Data stage information.

Setup stage

Direction: Host to modem

Purpose: Tell modem to report the status of a specific channel.

Details: The following table describes the outgoing packet structure.

Table A-6: Get Mode MUX—Setup stage

Offset	Size (bytes)	Field	Value
0	1	Request type	0xC0
1	1	Request	0x03
2	2	Value	0x0000 (not used)
4	2	Index	Channel selection • 0x0000 Data • 0x0001 Control (For CDMA devices, this indicates hybrid-MUX mode.) • 0x0002 NMEA
6	2	Length	0x0002 (length of data in Data stage)

Data stage (IN)

Direction: Modem to host

Purpose: Indicate the requested channel's current 27.010 MUX protocol status.

Details: The following table describes the incoming packet structure.

Table A-7: Get Mode MUX—Data stage (IN)

Offset	Size (bytes)	Field	Value
0	2	Mode	Mode • 0x0000 Init • 0x0001 Non-MUX • 0x0002 Transition • 0x0003 MUX

Status stage (OUT)

Direction: Host to modem

Purpose: Notify modem that the Data stage was received.

Details: Host sends a zero-length packet after receiving the Data stage.

Get NDIS Support

Request that the modem indicate if the NDIS driver is supported.

Stages:

- Setup—Host tells modem to report NDIS support status.
- Data (IN)—Modem indicates whether the NDIS driver is supported.
- Status (OUT)—Host indicates that it received the Data stage information.

Setup stage

Direction: Host to modem

Purpose: Tell modem to report whether the NDIS driver is supported.

Details: The following table describes the outgoing packet structure.

Table A-8: Get NDIS Support—Setup stage

Offset	Size (bytes)	Field	Value
0	1	Request type	0xC0
1	1	Request	0x04
2	2	Value	0x0000 (not used)
4	2	Index	0x0000 (not used)
6	2	Length	0x0002 (length of data in Data stage)

Data stage (IN)

Direction: Modem to host

Purpose: Indicate whether the NDIS driver is supported.

Details: The following table describes the incoming packet structure.

Table A-9: Get NDIS Support—Data stage (IN)

Offset	Size (bytes)	Field	Value
0	2	Supported	NDIS support • 0x0000 Not supported • 0x0001 Supported • 0x00FF Unknown (keep polling, modem not ready to respond)

Status stage (OUT)

Direction: Host to modem

Purpose: Notify modem that the Data stage was received.

Details: Host sends a zero-length packet after receiving the Data stage.

Get NDIS Preference

Request that the modem indicate whether or not the NDIS interface is supported.

Stages:

- Setup—Host tells modem to report whether the NDIS interface is supported.
- Data (IN)—Modem indicates whether the NDIS interface is supported.
- Status (OUT)—Host indicates that it received the Data stage information.

Setup stage

Direction: Host to modem

Purpose: Tell modem to report whether the NDIS interface is supported.

Details: The following table describes the outgoing packet structure.

Table A-10: Get NDIS Preference—Setup stage

Offset	Size (bytes)	Field	Value
0	1	Request type	0xC0
1	1	Request	0x05
2	2	Value	0x0000 (not used)
4	2	Index	0x0000 (not used)
6	2	Length	0x0002 (length of data in Data stage)

Data stage (IN)

Direction: Modem to host

Purpose: Indicate whether the NDIS interface is supported.

Details: The following table describes the incoming packet structure.

Table A-11: Get NDIS Preference—Data stage (IN)

Offset	Size (bytes)	Field	Value
0	2	Preference	NDIS preferred • 0x0000 Not preferred • 0x0001 Preferred • 0x00FF Unknown (keep polling, modem not ready to respond)

Status stage (OUT)

Direction: Host to modem

Purpose: Notify modem that the Data stage was received.

Details: Host sends a zero-length packet after receiving the Data stage.

Get Attributes

Request that the modem return firmware image attributes.

Stages:

- Setup—Host tells modem to report firmware attributes.
- Data (IN)—Modem reports firmware attributes.
- Status (OUT)—Host indicates that it received the Data stage information.

Setup stage

Direction: Host to modem

Purpose: Tell modem to report firmware attributes.

Details: The following table describes the outgoing packet structure.

Table A-12: Get Attributes—Setup stage

Offset	Size (bytes)	Field	Value
0	1	Request type	0xC0
1	1	Request	0x06
2	2	Value	0x0000 (not used)
4	2	Index	0x0000 (not used)
6	2	Length	0x0002 (Length of data in Data stage)

Data stage (IN)

Direction: Modem to host

Purpose: Report the firmware image's attributes.

Details: The following table describes the incoming packet structure.

Table A-13: Get Attributes — Data stage (IN)

Offset	Size (bytes)	Field	Value

0	2	Attributes	16-bit field with each bit representing an attribute of the firmware image
			• Bit 0 (Running image type)
			  ◦ 0 = Boot
			  ◦ 1 = Application
			• Bit 1 (Advanced Power Management support)
			  ◦ 0 = Not supported by firmware
			  ◦ 1 = Supported by firmware
			• Bit 2 (27.010 Flow Control Available)—this value is only meaningful when module is operating in MUX mode.
			  ◦ 0 = Not available
			  ◦ 1 = Available
			• Bit 3-15: Unused (default value = 0)

Status stage (OUT)

Direction: Host to modem

Purpose: Notify modem that the Data stage was received.

Details: Host sends a zero-length packet after receiving the Data stage.

Set Mode NMEA

Start or stop the NMEA data stream over the endpoint used for the NMEA channel.

Stages:

- Setup—Host tells modem to start or stop the stream.
- Status (IN)—Modem indicates completion of requested operation.

Setup stage

Direction: Host to modem

Purpose: Tell modem to start or stop the NMEA data stream.

Details: The following table describes the outgoing packet structure.

Table A-14: Set Mode NMEA—Setup stage

Offset	Size (bytes)	Field	Value
0	1	Request type	0x40
1	1	Request	0x07
2	2	Value	NMEA mode • 0x0000 Stop • 0x0001 Start
4	2	Index	0x0000 (not used)
6	2	Length	0x0000 (not used)

Status stage (IN)

Direction: Modem to host

Purpose: Notify host that the requested operation has completed.

Details: Modem returns a zero-length packet when the requested operation has completed.

Get Mode NMEA

Request that the modem indicate if NMEA streaming is stopped or started.

Stages:

- Setup—Host tells modem to report whether NMEA streaming is stopped or started.
- Data (IN)—Modem indicates the current status of NMEA streaming.
- Status (OUT)—Host indicates that it received the Data stage information.

Setup stage

Direction: Host to modem

Purpose: Tell modem to report whether NMEA streaming is stopped or started.

Details: The following table describes the outgoing packet structure.

Table A-15: Get Mode NMEA—Setup stage

Offset	Size (bytes)	Field	Value
0	1	Request type	0xC0
1	1	Request	0x08
2	2	Value	0x0000 (not used)
4	2	Index	0x0000 (not used)
6	2	Length	0x0002 (length of data in Data stage)

Data stage (IN)

Direction: Modem to host

Purpose: Indicate the current status of NMEA streaming.

Details: The following table describes the incoming packet structure.

Table A-16: Get Mode NMEA—Data stage (IN)

Offset	Size (bytes)	Field	Value
0	2	Mode	NMEA mode • 0x0000 Stopped • 0x0001 Started

Status stage (OUT)

Direction: Host to modem

Purpose: Notify modem that the Data stage was received.

Details: Host sends a zero-length packet after receiving the Data stage.

Set Host Power State

Notify the modem of the current power state of the host device.

Stages:

- Setup—Host tells modem whether the host is awake or suspended.
- Status (IN)—Modem indicates that it received the power state information.

Setup stage

Direction: Host to modem

Purpose: Tell modem whether the host is awake or suspended.

Details: The following table describes the outgoing packet structure.

Table A-17: Set Host Power State—Setup stage

Offset	Size (bytes)	Field	Value
0	1	Request type	0x40
1	1	Request	0x09
2	2	Value	State • 0x0000 Host device is awake • 0x0001 Host device is suspended
4	2	Index	0x0000 (not used)
6	2	Length	0x0000 (not used)

Status stage (IN)

Direction: Modem to host

Purpose: Notify host that the power state information was received.

Details: Modem returns a zero-length packet after it receives the information from the host.

Get TRU-Install Information

Request that the modem indicate which installation packages are stored on a TRU-Install enabled device.

When the host receives this information, it can determine whether to fully enumerate the device as a mass-storage device to load the installation package, or to re-enumerate as a modem because the package has already been loaded.

Stages:

- Setup—Host tells modem to report installation package information.
- Data (IN)—Modem returns the first portion of the CWE header, which indicates the installation package types.
- Status (OUT)—Host indicates that it received the Data stage information.

Setup stage

Direction: Host to modem

Purpose: Tell modem to report installation package information.

Details: The following table describes the outgoing packet structure.

Table A-18: Get TRU-Install Information—Setup stage

Offset	Size (bytes)	Field	Value
0	1	Request type	0xC0
1	1	Request	0x0A
2	2	Value	0x0000 (not used)
4	2	Index	0x0000 (not used)
6	2	Length	0x003C (length of data in Data stage)

Data stage (IN)

Direction: Modem to host

Purpose: Return the first portion of the CWE header, which indicates the installation package types.

Details: The following table describes the incoming packet structure.

Table A-19: Get TRU-Install Information — Data stage (IN)

Offset	Size (bytes)	Field	Value

0	60	Version	CWE Header, first 60 bytes
			• Byte 0: Revision number
			• Windows package details
			  ◦ Bytes 1–2: Product code
			  ◦ Bytes 3–4:Version
			• Mac package details
			  ◦ Bytes 5–6: Product code
			  ◦ Bytes 7–8:Version
			• Bytes 9–59: Reserved

Status stage (OUT)

Direction: Host to modem

Purpose: Notify modem that the Data stage was received.

Details: Host sends a zero-length packet after receiving the Data stage.

Set TRU-Install Mode

(TRU-Install enabled modems only)

Request that the device enumerate as a mass storage device (to install software) or as a modem (for normal use).

Note: The command is ignored if the modem does not support TRU-Install, or if it is in a state where the requested mode is not allowed (for example, boot-and-hold mode).

Stages:

- Setup—Host tells modem how it should enumerate.
- Status (IN)—Modem indicates completion of requested operation.

Setup stage

Direction: Host to modem

Purpose: Tell modem how to enumerate.

Details: The following table describes the outgoing packet structure.

Table A-20: Set TRU-Install Mode—Setup stage

Offset	Size (bytes)	Field	Value
0	1	Request type	0x40
1	1	Request	0x0B
2	2	Value	TRU-Install mode requested • 0x0000 Mass storage • 0x0001 Modem • 0x0002 Reserved
4	2	Index	0x0000 (not used)
6	2	Length	0x0000 (length of data in Data stage)

Status stage (IN)

Direction: Modem to host

Purpose: Notify host that the requested operation has completed.

Details: Modem returns a zero-length packet when the requested operation has completed.

Get Config Item

Request that the modem indicate whether a specific configuration item is supported or enabled.

Stages:

- Setup—Host tells modem to indicate whether a specific configuration item is supported or enabled.
- Data (IN)—Modem indicates support/enable status of requested configuration item.
- Status (OUT)—Host indicates that it received the Data stage information.

Setup stage

Direction: Host to modem

Purpose: Tell modem to return the status of a specific configuration item.

Details: The following table describes the outgoing package structure.

Table A-21: Get Config Item—Setup stage

Offset	Size (bytes)	Field	Value
0	1	Request type	0xC0
1	1	Request	0x0C
2	2	Value	0x0000
4	2	Index	Configuration item requested[a]: • 0x0000—TRU-Install • 0x0001—SD • 0x0002—NDIS (Windows only) • 0x0003—AutoConnect
6	2	Length	Variable—Maximum length of data stage that the host expects to receive • 0x0400 (1024 bytes) maximum

a. Availability of specific configuration item types is device/SKU-dependent.

Data stage (IN)

Direction: Modem to host

Purpose: Return the status of the specified configuration item.

Details: The following table describes the incoming packet structure.

Table A-22: Get Config Item — Data stage (IN)

Offset	Size (bytes)	Field	Value
0	Variable (minimum 2 bytes)	Configuration information	Data is dependent on configuration item requested. See Table A-23 for details.

Table A-23: Configuration information details

Configuration item		Offset	Size (bytes)	Field	Value
Index	Name				
0x0000	TRU-Install	0	2	Supported	• 0x0000 — Not supported • 0x0001 — Supported
		2	2	Enabled	• 0x0000 — Disabled • 0x0001 — Enabled
0x0001	SD	0	2	Supported	• 0x0000 — Not supported • 0x0001 — Supported
		2	2	Enabled	• 0x0000 — Disabled • 0x0001 — Enabled
0x0002	NDIS	0	2	Supported	• 0x0000 — Not supported • 0x0001 — Supported
		2	2	Enabled	• 0x0000 — Disabled • 0x0001 — Enabled
0x0003	AutoConnect	0	2	Supported	• 0x0000 — Not supported • 0x0001 — Supported
		2	2	Enabled	• 0x0000 — Disabled • 0x0001 — Enabled

Status stage (OUT)

Direction: Host to modem

Purpose: Notify modem that the Data stage was received.

Details: Host sends a zero-length packet after receiving the Data stage.

Set Config Item

Configure a specific configuration item.

Note: The device must be reset for any configuration changes to take effect. (Use Set Device Reset on page 95.)

Stages:

- Setup—Host tells module which items will be configured.
- Data (Out)—Host tells module to enable / disable specific configuration items.
- Status (In)—Module indicates that the request has completed.

Setup stage

Direction: Host to module

Purpose: Tell modem to set the characteristics of a specific configuration item.

Details: The following table describes the outgoing package structure.

Table A-24: Set Config Item—Setup stage

Offset	Size (bytes)	Field	Value
0	1	Request type	0x40
1	1	Request	0x0D
2	2	Value	0x0000
4	2	Index	Configuration item requested[a]: • 0x0000—TRU-Install • 0x0001—SD • 0x0002—NDIS (Windows only) • 0x0003—AutoConnect
6	2	Length	Length of the Data stage (OUT). The length is dependent on the configuration item requested.

a. Availability of specific configuration item types is device/SKU-dependent.

Data stage

Direction: Host to module

Purpose: Indicate new details for a specific configuration item.

Details: The following table describes the outgoing package structure.

Table A-25: Set Config Item — Data stage (OUT)

Offset	Size (bytes)	Field	Value
0	Variable (minimum 2 bytes)	Configuration information	Data is dependent on configuration item requested. See Table A-23 for details.

Table A-26: Configuration information details

Configuration item		Offset	Size (bytes)	Field	Value
Index	Name				
0x0000	TRU-Install	0	2	Enable state	• 0x0000—Disable • 0x0001—Enable
0x0001	SD	0	2	Enable state	• 0x0000—Disable • 0x0001—Enable
0x0002	NDIS	0	2	Enable state	• 0x0000—Disable • 0x0001—Enable
0x0003	AutoConnect	0	2	Enable state	• 0x0000—Disable • 0x0001—Enable

Status stage (IN)

Direction: Module to host

Purpose: Notify host that the requested operation has completed.

Details: Module sends a zero-length packet when the requested operation has completed.

Set Device Reset

Tell the modem to reset.

Note: This command must be issued after using Set Config Item on page 93 so that new settings will take effect.

Stages:

- Setup—Host tells modem to reset.
- Status (In)—Modem indicates that device has been reset.

Setup stage

Direction: Host to modem

Purpose: Tell modem to reset so that it will enumerate with new configuration settings.

Details: The following table describes the outgoing packet structure.

Table A-27: Set Device Reset—Setup stage

Offset	Size (bytes)	Field	Value
0	1	Request type	0x40
1	1	Request	0x0E
2	2	Value	0x0000
4	2	Index	0x0000
6	2	Length	0x0000

Status stage (IN)

Direction: Modem to host

Purpose: Notify host that the device has been reset.

Details: Modem returns a zero-length packet after the device resets.

Set Host Info

Tell the module details about the host configuration.

Stages:

- Setup—Host indicates which configuration information is being sent to the modem.
- Data (Out)—Host sends the configuration details.
- Status (In)—Modem indicates that requested operation has completed.

Setup stage

Direction: Host to modem

Purpose: Tell modem which host configuration details are being identified.

Details: The following table describes the outgoing packet structure.

Table A-28: Set Host Info—Setup stage

Offset	Size (bytes)	Field	Value
0	1	Request type	0x40
1	1	Request	0x0F
2	2	Value	0x0000
4	2	Index	Configuration type • 0x0000—Feature configuration • 0x0001—Host type • 0x0002—Service pack version • 0x0003—Host and system up time • 0x0004—Reserved
6	2	Length	Length of Data stage (OUT)

Data stage (Out)

Direction: Host to modem

Purpose: Indicate details about identified configuration type.

Details: The following table describes the outgoing packet structure.

Table A-29: Set Host Info—Data stage (OUT)

Offset	Size (bytes)	Field	Value
0	Variable	Configuration information	Data is dependent on configuration item requested. See Table A-23 for details.

Table A-30: Configuration information—Feature Configuration

Offset	Size (bytes)	Field	Value
0	2	Feature bitmap	
	Bit 0	NMEA listening	• 0—(Default) Host does not listen to the NMEA port • 1—Host listens to the NMEA port
	Bit 1	Restart indication	• 0—Host does not require !RESET restart indication • 1—Host requires !RESET restart indication
	Bit2-15	Reserved	• Not used

Table A-31: Configuration information—Host Type

Offset	Size (bytes)	Field	Value
0	1	OS type	• 0x00—Windows • 0x01—Linux • 0x02—Windows CE • 0x03—Windows Mobile • 0x04—Mac • 0xFF—Other/unknown

1	1	OS version	• Windows
			  • 0x40—Windows NT 4.0
			  • 0x50—Windows 2000
			  • 0x51—Windows XP
			  • 0x60—Windows Vista
			  • 0x61—Windows 7
			• Windows CE
			  • 0x30—CE 3.0
			  • 0x40—CE 4.0
			  • 0x50—CE 5.0
			  • 0x60—CE 6.0
			• Windows Mobile
			  • 0x30—Mobile 2003
			  • 0x50—Mobile 5.0
			  • 0x60—Mobile 6.0
			  • 0x61—Mobile 6.1
			  • 0x65—Mobile 6.5
			  • 0x70—Mobvile 7.0
			• Linux/Mac/Other/Unknown
			  • 0xFF

Table A-31: Configuration information — Host Type (Continued)

Offset	Size (bytes)	Field	Value
2	1	Processor type	• 0x00 — x86 • 0x01 — x64 • 0x02 — MIPS • 0x03 — ARM • 0xFF — Other
3	1	Reserved	Not used

Table A-32: Configuration information — Service pack version

Offset	Size (bytes)	Field	Value
0	1	Service pack version	• 0x00 — Service pack 0 • 0x01 — Service pack 1 • 0x02 — Service pack 2 • etc.

Table A-33: Configuration information — Host and system up time

Offset	Size (bytes)	Field	Value
0	8	System up Time1	Number of seconds since the system was started. • Shown in network byte order ('big-endian')
8	8	System up Time2	Windows 7 • Number of seconds that the system has been awake since it was started. • Shown in network byte order ('big-endian') All other O/S: • Reserved
16	2	Host time (Year)	Year • Shown in network byte order ('big-endian')
18	1	Host time (Month)	Month (1–12)
19	1	Host time (Day)	Day (1–31)
20	1	Host time (Hour)	Hour (0–23)
21	1	Host time (Minute)	Minute (0–59)
22	1	Host time (Second)	Second (0–59)

Table A-33: Configuration information—Host and system up time

Offset	Size (bytes)	Field	Value
23	1	Host time (Timezone)	Offset from UTC (in 15 minute increments) Examples: • -2 = UTC - 30 minutes • -1 = UTC - 15 minutes • 0 = UTC • 1 = UTC + 15 minutes • 2 = UTC + 30 minutes
24	1	Host time (Daylight savings)	Time adjustment when daylight savings is in effect • 0x00—No adjustment • 0x01—Plus one hour • 0x02—Plus two hours

Status stage (IN)

Direction: Module to host

Purpose: Notify host that the requested operation has completed.

Details: Module sends a zero-length packet when the requested operation has completed.

B: Supported CDC USB commands

B

The devices described in this document support various CDC USB commands, requests, and class-specific notifications. The following tables describe the support provided by the devices.

Message support status in these tables is represented by:

- ✔—Supported: Fully implemented on the indicated device.
- E—Empty: Blankly implemented—the modem responds to the request but doesn't actually perform it.
- ✘—Not supported: Not implemented at any level.

Table B-1: Supported commands/requests by chipset[a]

Request	Code	MDM6085 QSC6085	(Legacy VID/PID) MDM6200 MDM6270 MSM6290 QSC6270	(Direct IP VID/PID) MDM6200 MDM6270 MDM8200 MDM8200A MDM8220 MDM9200 MSM6290 QSC6270	(QMI VID/PID) MDM9200 MDM9600
SEND_ENCAPSULATED_COMMAND	00h	E	E	✔ (ECM interface)	✔
GET_ENCAPSULATED_RESPONSE	01h	E	E	✔ (ECM interface)	✔
SET_COMM_FEATURE	02h	E	E	E	✘ (MDM9200)
GET_COMM_FEATURE	03h	E	E	E	TBD
CLEAR_COMM_FEATURE	04h	E	E	E	TBD
SET_LINE_CODING	20h	✔	✔	✔	✔ (MDM9200)
GET_LINE_CODING	21h	✔	✔	✔	✔ (MDM9200)
SET_CONTROL_LINE_STATE	22h	✔	✔	✔	✔ (MDM9200)
SEND_BREAK	23h	E	E	E	TBD

a. For a list of devices using these chipsets, see Table 1-1 on page 15.

Table B-2: Supported class-specific notifications by chipset[a]

Request	Code	MDM6085 QSC6085	(Legacy VID/PID) MDM6200 MDM6270 MSM6290 QSC6270	(Direct IP VID/PID) MDM6200 MDM6270 MDM8200 MDM8200A MDM8220 MDM9200 MSM6290 QSC6270	(QMI VID/PID) MDM9200 MDM9600
NETWORK_CONNECTION	00h	✘	✘	✘	TBD
RESPONSE_AVAILABLE	01h	E	E	E	TBD
SERIAL_STATE	20h	✔	✔	✔	TBD
CONNECTION_SPEED_CHANGE	2Ah	✔	✘	✘	TBD

a. For a list of devices using these chipsets, see Table 1-1 on page 15.

C: QUALCOMM DM Data Logging

This section provides guidelines for OEMs whose devices are running an operating system other than Windows and who need to implement DM logging capability (for carrier certification requirements or field troubleshooting) to support logging tools such as QUALCOMM's QXDM, Spirent's UDM, or Sierra Wireless' SwiLog Plus tool.

The recommended setup is to implement a "pass-thru" of the modem's DM interface to an external interface (on the non-Windows host device) connected to a Windows computer running a Windows-based logging tool mentioned above. See the following figure.

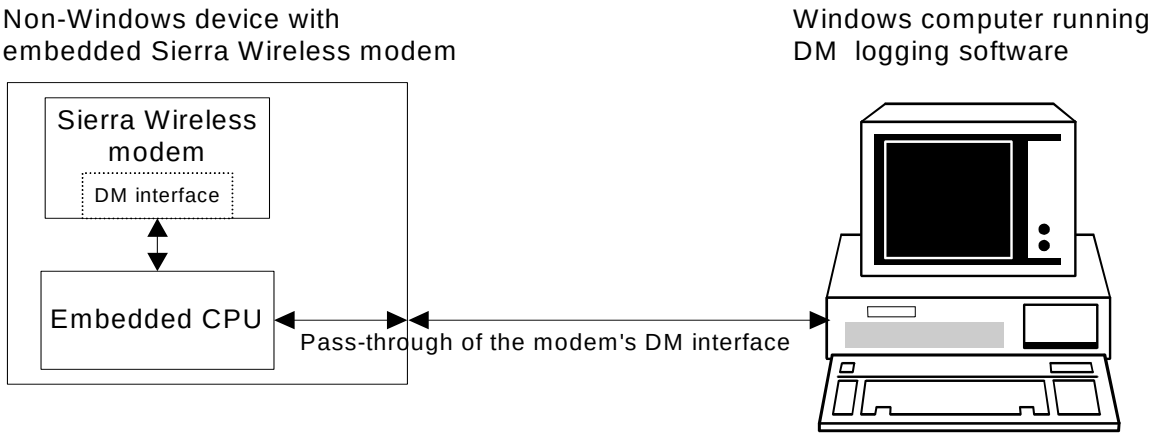

Figure C-1: Recommended setup for logging QUALCOMM DM data

Handling DM for logging purposes

Note: This section applies only to modems that are configured to have DM messages pass over the HIP interface—for example, when USB endpoint 4 is configured for CnS and QUALCOMM DM over HIP.

The host USB driver must handle the HIP encapsulation and de-encapsulation of DM messages; for more information on encapsulation and on HIP in general, see the "HIP Basics" and "HIP format" chapters in the *CDMA 1xEV-DO CnS Reference (Document 2130754)*.Your driver must do the following:

1. Using the Message ID, determine whether the message is a DM message. DM messages have a Message ID of:
   - 0x2D (Modem Protocol Stack Indication), or
   - 0x6D (Host Protocol Stack Indication)

Perform the following step only if the message is a DM message.

2. If the message is from SWILOG (Message ID = 0x2D), encapsulate the message in a HIP packet, and transmit the packet to

the modem.
If the message is to SWILOG (Message ID = 0x6D), remove the encapsulation data and pass the DM message to SWILOG.

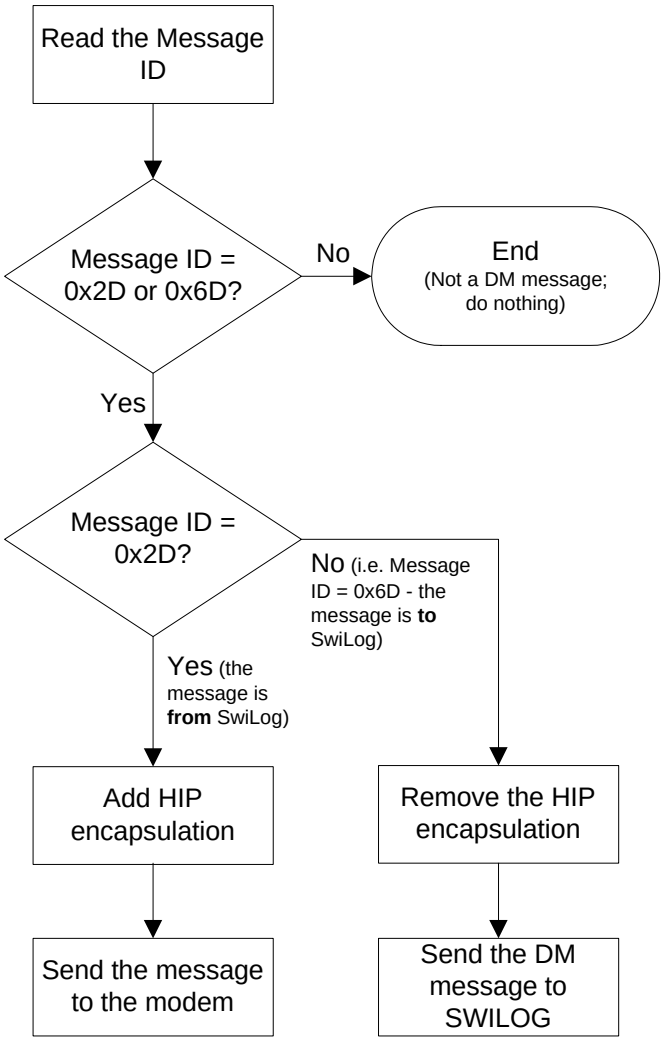

Figure C-2: Handling of DM messages by your USB driver

Pass-thru application considerations

The following are general characteristics of the pass-thru application:

- It does not need to fragment the message into smaller HIP messages before passing it along—it simply passes the HIP packet, unchanged, from one device to the other.
- SWILOG uses 2 Kb buffers.
- The modem's DM buffer has a finite size - if the buffers are not read and the buffer capacity is reached, the oldest packets are dropped, ensuring that the newest data is always available. This could occur if the majority of log items are enabled, particularly if a physical serial port is used as the interface.
- Typically, traffic from the host is sent in bursts.

D: Direct IP mode

This section describes issues pertaining to modems supporting Direct IP mode. See Direct IP on page 71 for a description of this service.

For more detailed information about Direct IP mode, refer to the Direct IP Integration Guide (Document #2131327) (contact your Sierra Wireless representative).

Performance considerations

You can use the Direct IP interface on HSPA and HSPA+ networks. The following are performance considerations for both network types:

- Direct IP on HSPA+ networks
  - Direct IP vs. PPP—Direct IP throughput rates will be at least as high as PPP rates on HSPA+ networks. CPU usage may also be lower for Direct IP vs. PPP.
- Direct IP on HSPA networks
  - Direct IP vs. PPP—There should be no significant difference in data rate between the two services. CPU usage may also be lower for Direct IP vs. PPP.

Direct IP drivers

To obtain and install Linux driver files providing Direct IP support for your modem, refer to the Linux SDK Integration Guide (Document #2131103) (contact your Sierra Wireless representative).

Direct IP interface

Direct IP is implemented using custom ECM (Ethernet Control Module) interfaces—7, 10, and 11, depending on the device's default interface configuration, as described in Direct IP on page 71.

As shown in Table 2-11 on page 48, the default Direct IP interface (DIP1, Interface 7) is implemented over three endpoints—the endpoint numbers are determined based on which interfaces are enumerated:

- Bulk IN and Bulk OUT—These endpoints are used to transfer data from the modem to the host (IN), and from the host to the modem (OUT).
- Interrupt—This endpoint is used to notify the host of interface events.

Direct IP communication / session setup

*Note: The modem must have at least one profile created before a connection can be made. Use **AT!SCPROF** (see AirCard/AirPrime UMTS Devices Supported AT Command Reference (Document 2130617)), **AT+CGDCONT**, and **AT$QCPDPP** (see AirPrime MC/SL-series (UMTS/LTE) Extended AT Command Reference (Document 2130616)) to create and configure profiles.*

To set up a Direct IP data session (start the connection):

- Via AT—Use AT!SCACT=1,<profile> where <profile> has the required APN. ('1' activates the session, '0' terminates the session)
- Via CnS—Use the CnS object 'Manage Packet Session (0x7004)'.

Note: For a detailed explanation of HIP messaging and the Link Status Indication, refer to "HIP Basics" and "Unsolicited HIP messages" in the AirPrime UMTS MC and SL Series CnS Reference (Document 2130602).

To receive a control message (such as a Link Status Indication or HIP Sync) from the modem:

1. Receive a notification indication over the Direct IP interface's Interrupt endpoint.
2. Send a GET_ENCAPSULATED_RESPONSE request over the USB CDC command endpoint (endpoint 0). The modem returns the control message as the payload of the response.

To send a control message (such as a HIP Sync) to the modem:

1. Send a SEND_ENCAPSULATED_COMMAND request over the USB CDC command endpoint (endpoint 0) with the control message as the payload of the request.

Examples

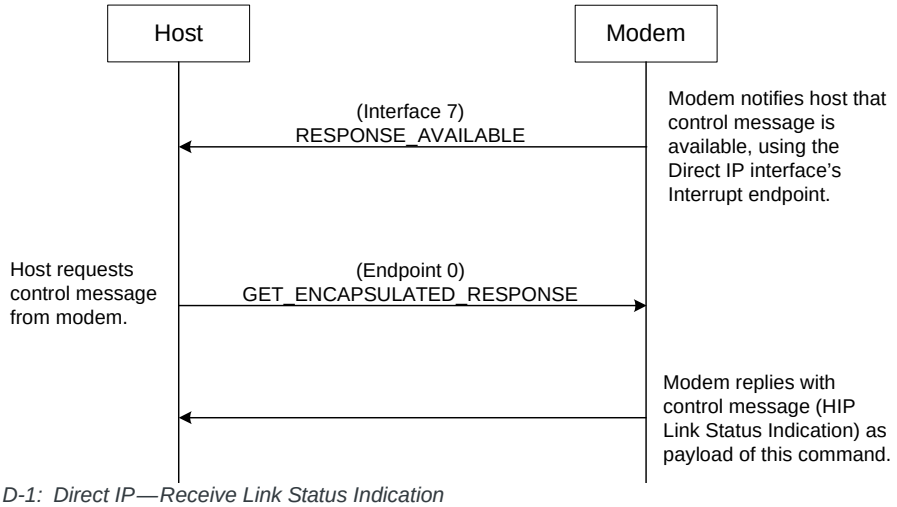

Figure D-1: Direct IP—Receive Link Status Indication

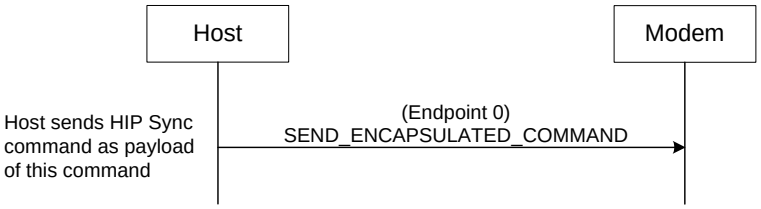

Figure D-2: Direct IP—Host/modem synchronization

Switching between PPP and Direct IP modes

This section applies to modems supporting both PPP and Direct IP modes. (See Table 1-1 on page 15 for devices with Direct IP and legacy PIDs.)

Switch a modem from PPP mode to Direct IP mode

1. Run AT!UDPID=68A3. (0x68A3 is the 'standard' Direct IP PID. You can also use a carrier-specific Direct IP PID if appropriate.)
2. Reset the modem.

Switch a modem from Direct IP mode to PPP mode

1. Run AT!UDPID=<pid> where <pid> is the default PID used by the modem. (For example, 683C is the PID used for MC8790V modems.)
2. Reset the modem.

*Note: To show the device's current PID, use **AT!UDPID?**. To show available PIDs, use **AT!UDPID=?**.*

Configuring USB interface composition in Direct IP mode

To display or customize the set of interfaces presented by the modem in Direct IP mode, use the AT!UDUSBCOMP command:

- AT!UDUSBCOMP=? shows available interface configurations
- AT!UDUSBCOMP? shows the current configuration
- AT!UDUSBCOMP=<xx> sets the current configuration (<xx>). (You must reset the modem after changing the current configuration for the change to take effect.)

The modem's default interface configuration is SKU-dependant—see UMTS devices (DIP VID/PID)—Non-MUX descriptors on page 52 for lists of possible interfaces and typical interface configurations.

E: QMI

This section applies to modems supporting both QMI and Direct IP modes. (See Table 1-1 on page 15 for devices with QMI and Direct IP PIDs.)

For more detailed information about QMI mode, refer to QMI (NET interface) on page 72, and the *Linux QMI SDK Application Developer's Guide* (contact your Sierra Wireless representative).

QMI drivers

To obtain and install Linux serial and network driver files providing QMI support for your modem, refer to the *Linux QMI SDK Application Developer's Guide*.

QMI interface

As shown in Table 2-22 on page 59, the QMI interface (NET, Interface 8) is implemented over three endpoints:

- Bulk (6-IN and 4-OUT) —These endpoints are used to transfer data from the modem to the host (IN), and from the host to the modem (OUT).
- Interrupt (5-IN)—This endpoint is used to notify the host of interface events.

QMI communication / session setup

QMI data sessions are set up and managed using API functions available in the QMI SDK—AT messages are not used for this purpose.

For details on identifying and using QMI API functions for managing data data sessions, refer to the *Linux QMI SDK Application Developer's Guide*.

Switching between QMI and Direct IP modes

Switch a modem from QMI mode to Direct IP mode

1. Run AT!UDPID=68A3. (0x68A3 is the 'standard' Direct IP PID. You can also use a carrier-specific Direct IP PID if appropriate.)
2. Reset the modem.

Switch a modem from Direct IP mode to QMI mode

1. Run AT!UDPID=68A2 (0x68A2 is the QMI PID.)
2. Reset the modem.

*Note: To show the device's current PID, use **AT!UDPID?**. To show available PIDs, use **AT!UDPID=?**.*

Configuring USB interface composition in QMI mode

To display or customize the set of interfaces presented by the modem in QMI mode, use the AT!UDUSBCOMP command:

- AT!UDUSBCOMP=? shows available interface configurations
- AT!UDUSBCOMP? shows the current configuration
- AT!UDUSBCOMP=<xx> sets the current configuration (<xx>). (You must reset the modem after changing the current configuration for the change to take effect.)

The modem's default interface configuration is SKU-dependant—see UMTS devices (QMI VID/PID) non-MUX descriptors on page 61 for lists of possible interfaces and typical interface configurations.

F: User-developed USB Drivers

F

User-developed drivers

This section applies to developers who are creating their own drivers for Sierra Wireless MC- and SL-series embedded modules.

Handshaking

The host must act as a USB host device to interface with the module.

The module uses the USB standard Suspend and Resume functions to control sleep and wakeup states. For details, refer to *Universal Serial Bus Specification, Rev 2.0*.

Suspend

The module supports USB Selective Suspend mode.

When the module enters suspend mode, it shuts down the USB clock to save power.

While in the suspend state:

- The module provides power to the D+/- interface to signal its current state to the host device.
- The host must maintain input voltage (VCC)

Refer to *Universal Serial Bus Specification, Rev 2.0* for critical timing parameters for the suspend state.

Resume

USB activity may be resumed by either the USB host or by the module.

If the host initiates USB activity:

1. USB transceiver detects the change in bus activity and triggers the USB_RESUME interrupt to the module's processor.
2. Module enables its USB clock and responds to the host.

If the module initiates USB communication (Remote Wakeup):

1. Module enables its USB clock.
2. Module enables the USB transceiver.
3. Module sends the resume signal for at least 20 ms.

Refer to *Universal Serial Bus Specification, Rev 2.0* for critical timing parameters for the resume state.

Host USB driver requirements

The USB driver on the host device must meet these critical requirements:

- The host USB driver must support remote wakeup, resume, and suspend operations as described in *Universal Serial Bus Specification, Rev 2.0*.
- The host USB driver must support serial port emulation. The module implements both 27.010 multiplexing and USB-CDC.

When the host doesn't have any valid data to send, the host USB driver should NOT send any SOF tokens (start-of-frames) to the module. These tokens keep the module awake and cause unnecessary power consumption.

Index

Symbols

%%Get TRU-Install Information, 95

Numerics

27.010 MUX protocol
    query (Get), 79
    set up, 77
    tear down, 76

A

Access Point Name (APN), 70
advanced power management, 19, 84
APN (Access Point Name), 70
architecture, supported driver, 20
architecture, USB, 19
AT commands, 66
AT/PPP data channel, 66

B

boot-and-hold mode
    supported commands, 74

C

CDC USB commands, 101
CDC-ACM drivers, 19
class-specific notifications, 102
CLEAR_COMM_FEATURE command, 101
CnS (Control and Status), 67
commands
    CDC USB, 101
    vendor-specific, 73
Communication Interface Class, 65
configuration items, get, 91
configuration items, set, 93
CONNECTION_SPEED_CHANGE notification, 102
Control and Status (CnS), 67
control channel, standard, 65

D

Data 1, Data 2, Data 3 data channels, 70
data channel
    AT/PPP, 66
    Data 1, Data 2, Data 3, 70
    DM, 68
    HIP (Host Interface Protocol), 67
    NMEA, 69
data stream, NMEA
    query status, 86
    start or stop, 85
debugging, 68
device power state, 75
Diagnostic Monitoring (QUALCOMM DM), 68, 103
Direct IP
    control message, receive, 106
    control message, send, 106
    data session, setup, 106
    description, 105
    drivers, 105
    HIP sync, process, 107
    interface, endpoints, 105, 109
    Link Status Indication, process, 106
DLCI control channel assignments, hybrid-MUX mode
    MDM6085, 30
    MSM6800, 30
    MSM6800A, 30
    QSC6085, 30
DLCI control channel assignments, MUX mode
    MDM6085, 28
    MSM6800, 28
    MSM6800A, 28
    QSC6085, 28
DM (QUALCOMM Diagnostic Monitoring), 68, 103
documents, reference, 18
driver
    architecture, supported, 20
    DM data, handling, 103
    NDIS
        enabled or disabled, 82
        supported or not, 81
drivers, CDC-ACM, 19

E

endpoints, hybrid-MUX mode
    MDM6085, 30
    MSM6800, 30
    MSM6800A, 30
    QSC6085, 30
endpoints, MUX mode
    MSM6800, 28
    MSM6800A, 28
    QSC6085, 28
endpoints, non-MUX mode
    MDM6085, 23
    QSC6085, 23
endpoints, USB
    overview, 19

F

firmware image attributes, 83
full-speed standard, compliance, 19

G

Get Config Item, 91
Get Mode MUX, 79
Get Mode NMEA, 86
Get NDIS Preference, 82
Get NDIS Support, 81
Get TRU-Install Information, 88
GET_COMM_FEATURE command, 101
GET_ENCAPSULATED_RESPONSE command, 101
GET_LINE_CODING command, 101
GPS, 69

H

handshaking
    USB, 111
Heatherington escape method, 66
high-speed standard, compliance, 19
HIP (Host Interface Protocol)
    data channel, 67
    description, 67
Host Interface Protocol (HIP)
    data channel, 67
    description, 67
host power state
    set, 87
hybrid-MUX mode, endpoints, 30

I

interrupt channels, 65

L

LBS (Location-Based Services), 69
Location-Based Services (LBS), 69
logging of QUALCOMM DM Data, 103
logical protocols, 65
low-speed standard, compliance, 19

M

Mac package details (TRU-Install), 89
management element, 65
mass storage device, 71
    See also TRU-Install.
MDM6085
    DLCI control channel assignments
        hybrid-MUX mode, 30
        MUX mode, 28
    endpoints
        hybrid-MUX mode, 30
        MUX mode, 28
        non-MUX mode, 23
    USB descriptors, 25
MDM6270
    USB descriptors
        legacy VID/PID, 36, 61
MDM8200
    USB descriptors, 52– 63
modem power state, 75
MSM6200
    USB descriptors
        legacy VID/PID, 36, 61
MSM6290
    USB descriptors
        legacy VID/PID, 36, 61
        TRU-Install mode, 42
MSM6290 (DIP VID/PID)
    USB descriptors, 52– 63
MSM6800
    DLCI control channel assignments
        hybrid-MUX mode, 30
        MUX mode, 28
    endpoints
        hybrid-MUX mode, 30
        MUX mode, 28
MSM6800A
    DLCI control channel assignments
        hybrid-MUX mode, 30
        MUX mode, 28
    endpoints
        hybrid-MUX mode, 30
        MUX mode, 28
MSM7200A
    USB descriptors
        TRU-Install mode, 42
MSM7201A
    USB descriptors
        TRU-Install mode, 42
MUX mode
    activating/deactivating, 21
    get, 79
    set, 77
MUX protocol, 27.010
    query (Get), 79
    set up, 77
    tear down, 76

N

National Marine Electronics Association (NMEA)
    data stream
        query status, 86
        start or stop, 85
    description, 69
NDIS
    preference, 82
    supported or not, 81
NETWORK_CONNECTION notification, 102

NMEA (National Marine Electronics Association)
  data stream
    query status, 86
    start or stop, 85
  description, 69
non-MUX mode
  endpoints
    MDM6085, 23
    MDM6270, 33
    Mdm6270, 59
    MDM8200 (DIP VID/PID), 49
    MSM6200, 33, 59
    MSM6290, 33, 59
    MSM6290 (DIP VID/PID), 49
    QSC6085, 23
    QSC6270, 33, 59
  set, 76
notification elements, 65
notifications, class-specific, 102

P

package details (TRU-Install), 89
power modem on/off, 75
power state
  host, set, 87
  modem, set, 75
PPP
  Data 1, Data 2, Data 3 data channels, 70
  data channel, 66
protocol
  27.010 MUX
    query (Get), 79
    set up, 77
    tear down, 76
  HIP (Host Interface Protocol). See HIP (Host Interface Protocol).
  logical, 65
protocol stack, wireless, 68

Q

QMI
  data session, setup, 109
  description, 109
  drivers, 109
QoS (quality of service), 70
QSC6085
  DLCI control channel assignments
    hybrid-MUX mode, 30
    MUX mode, 28
  endpoints
    hybrid-MUX mode, 30
    MUX mode, 28
    non-MUX mode, 23
  USB descriptors, 25
QSC6270
  USB descriptors
    legacy VID/PID, 36, 61
QUALCOMM Diagnostic Monitoring (DM), 68, 103
quality of service (QoS), 70
QXDM, 103

R

reference documents, 18
RESPONSE_AVAILABLE notification, 102
resume
  mode, USB, 111

S

SEND_BREAK command, 101
SEND_ENCAPSULATED_COMMAND command, 101
SERIAL_STATE notification, 102
services, 65
Set Config Item, 93
Set Device Power State, 75
Set Host Information, 96
Set Host Power State, 87
Set Mode MUX, 77
Set Mode NMEA, 85
Set Mode Non-MUX, 76
Set TRU-Install Mode, 90
SET_COMM_FEATURE command, 101
SET_CONTROL_LINE_STATE command, 101
SET_LINE_CODING command, 101
Spirent's UDM, 103
stack, wireless protocol, 68
standard control channel, 65
standards compliance, USB, 19
suspend
  mode, USB, 111
SwiLog Plus, 103

T

TCP/IP, 66
TIES, 66
TRU-Install
  endpoints used, 71
  query, 88, 95
  set mode, 90

U

UDM (Spirent), 103
USB
  handshaking, 111
  host driver requirements, 112
  resume mode, 111
  suspend mode, 111
  user-developed drivers, 111

USB commands, CDC, 101
USB descriptors
MDM6085, 25
MDM6200
legacy VID/PID, 36, 61
MDM6270
legacy VID/PID, 36, 61
MDM8200, 52– 63
MSM6290
legacy VID/PID, 36, 61
TRU-Install mode, 42
MSM6290 (DIP VID/PID), 52– 63
MSM7200A
TRU-Install mode, 42
MSM7201A
TRU-Install mode, 42
QSC6085, 25
QSC6270
legacy VID/PID, 36, 61

V

vendor-specific commands, 73

W

Windows package details (TRU-Install), 89
wireless protocol stack, 68

SIERRA
WIRELESS