AirPrime WP76xx/WP77xx

USB Driver Developer's Guide

SIERRA WIRELESS®

41110895
Rev 1

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

Copyright

Trademarks

Sierra Wireless®, AirPrime®, AirLink®, AirVantage® and the Sierra Wireless logo are registered 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.

Other trademarks are the property of their respective owners.

Contact Information

Sales information and technical support, including warranty and returns	Web: sierrawireless.com/company/contact-us/ Global toll-free number: 1-877-687-7795 6:00 am to 5:00 pm PST
Corporate and product information	Web: sierrawireless.com

Revision History

Revision number	Release date	Changes
1	July 2018	Initial release

Contents

List of Tables

List of Figures

1: Introduction

Purpose of this Guide

This guide is intended for use when designing non-Windows drivers for the following modules:

- AirPrime WP76xx Series
- AirPrime WP77xx Series

The guide describes the following information:

- Services (protocols) available over the USB connection
- Physical USB interface (device and endpoint descriptors)

Related documents

Sierra Wireless documents

[1] AirPrime WP8548/WP75xx/WP76xx/WP77xx AT Command Reference

For additional Sierra Wireless documents, refer to product-specific pages on source.sierrawireless.com.

Industry/other documents

Related and supporting documents include:

[2] Terminal Equipment to User Equipment (TE-UE) multiplexer protocol (Release 6)
Website: www.3gpp.org

[3] Universal Serial Bus Class Definitions for Communication Devices, Version 1.1
Website: www.usb.org

[4] Universal Serial Bus Specification, Rev 2.0
Website: www.usb.org

2: USB Architecture

This chapter describes the WP Series module's driver architecture for data transfer, and its physical USB interfaces.

It is assumed that the developer has a good understanding of USB principles and architecture. For detailed information on USB specifications, refer to [3] Universal Serial Bus Class Definitions for Communication Devices, Version 1.1. This (and other resources) are available at www.usb.org.

USB Standards Compliance

AirPrime WP Series modules comply with USB 2.0 standards, including the following:

- USB OTG (Host and Device)
- USB 2.0
- High-speed (480 Mb/s) data transfer
- Standard USB flow control
- Standard USB power management—Suspends the USB bus when idle, to conserve power.

For consistency across module types, Sierra Wireless modules employ a static Interface numbering strategy for enabled (concurrent) interfaces, whereas the USB 2.0 specification dynamically numbers interfaces based on the number of enabled interfaces.

USB Interfaces

The architecture used for AirPrime WP Series modules is derived from the Abstract Control Model (ACM) described in [3] Universal Serial Bus Class Definitions for Communication Devices, Version 1.1.

These modules do not claim any CDC classes in the descriptors, do not support functional descriptors, and therefore are not normally compatible with native CDC-ACM drivers.

Host drivers must distinguish Sierra Wireless modules by VID/PID and interface number.

Interface numbering

Interfaces on Sierra Wireless modules are numbered consistently across all module types, as detailed in Table 2-1, which . For example, RMNET0 is always Interface 8, regardless of which interfaces (from 0–7) are supported and enabled.

Therefore, the host driver must separate services by the Interface Descriptor field bInterfaceNumber.

Table 2-1: Sierra Wireless USB Interface Support

Interface Name	Interface Number[a]	WP76xx[b]	WP77xx
DIAG (DM)	0 (0x00)	Y	Y
ADB	1 (0x01)	N	N
NMEA	2 (0x02)	Y	Y
Modem	3 (0x03)	Y	Y
AT	4 (0x04)	Y	Y
Raw Data	5 (0x05)	Y	Y
OSA	6 (0x06)	N	N
RMNET0	8 (0x08)	Y	Y
RMNET1	10 (0x0A)	N	N
RMNET2	11 (0x0B)	N	N
RNDIS	14 (0x0D)	N	N
Audio	16 (0x10) 17 (0x11) 18 (0x12)	Y	Y
ECM	19 (0x13) 20 (0x14)	Y	Y
UBIST	21 (0x15)	N	N
NCM	22 (0x16)	N	N
EEM	24 (0x18)	N	N

a. Interface numbers not listed are reserved for future use
b. For descriptions of supported interfaces, see Supported Interfaces—Service Descriptions on page 10.

Enabling/Disabling USB Interfaces

The USB interface configuration can be modified by enabling/disabling supported interfaces using AT commands (for usage details, refer to [1] AirPrime WP8548/WP75xx/WP76xx/WP77xx AT Command Reference):

1. Establish a serial connection:
   - (Linux) # microcom /dev/ttyAT
   - (Windows) Use a terminal emulator.
2. Unlock commands:

   AT!ENTERCND=”A710”
3. Display current USB composition:

   AT!USBCOMP?

4. Display parameter details to select the correct interfaces to enable:
   AT!USBCOMP=?
5. Set the desired configuration:
   AT!USBCOMP=<Config Index>,<Config Type>,<Interface bitmask>

Important: *By default, the DIAG (DM) interface is enabled. !USBCOMP can be used (only on WP76xx/WP77xx) to disable DIAG (DM), but cannot re-enable it.*

6. Reset the module to use the new configuration:
   AT!RESET

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 (Input) or from the host to the modem (Output).
- Maximum number of assignable endpoints—15 input, 15 output (Endpoint 0x00 (Input/Output) is reserved for USB Control.)
- Endpoint numbering—Inputs (0x81–0x8F); Outputs (0x01–0x0F)
- Endpoint numbers are assigned dynamically as the USB interfaces are enumerated, starting at 0x81/0x01.

Note: Figure 2-1 on page 11 and Table 2-2 on page 12 illustrate the architecture of a WP Series module with all supported interfaces enabled. Because endpoints are dynamically allocated, the endpoint addresses shown in the table will vary depending on which interfaces are enabled or disabled. When developing your drivers, you must the actual physical endpoints for enumerated interfaces.

Supported Interfaces—Service Descriptions

The following services are available over the supported interfaces described in Table 2-1 on page 9:

- DIAG (DM)—Diagnostic (or DM) interface intended for use with Qualcomm tools, such as QXDM, during product development.
- NMEA—Provides external application with standard NMEA messages.
- MODEM/AT—Main command & control interface and supports Data calls. Note that AT commands cannot be used while a PPP session is established (the session must be terminated).
- AT—Interface for AT command entry.
- Raw Data—Provides an interface for communication between external host and embedded Linux application.
- RMNET0—Intended for wireless data transfer between host and wireless network through device modem stack.

  If data path routing support is required (e.g. high QoS traffic on one interface, regular traffic on another), the USB driver must implement QMAP mode, which enables multiple virtual channels over the physical RMNET0 channel. (Older Sierra Wireless modules supported only 2–3 channels using dedicated

physical channels (RMNET0, RMNET1, RMNET2)). Contact Sierra Wireless for details.

- Audio—Provides interfaces for using Audio over USB.
- ECM—CDC Ethernet Control Model (ECM) interface providing Ethernet-style networking over USB.

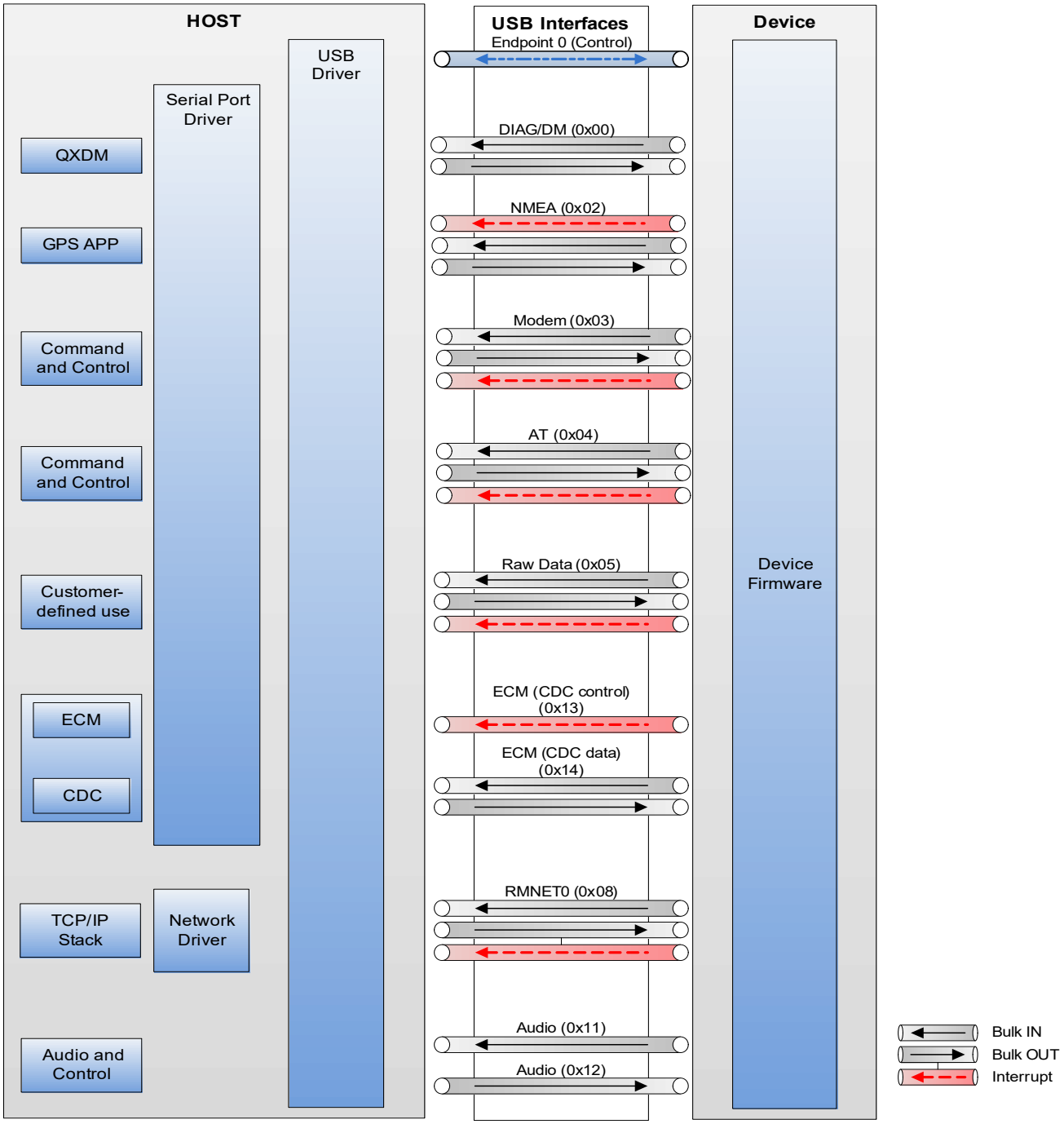

Figure 2-1: Supported USB Interfaces with Endpoint Details

Descriptors

Table 2-2 presents the Device, Configuration, and Endpoint descriptors for the full suite of available USB interfaces supported by WP76xx/WP77xx modules. The available USB interfaces (as identified in Table 2-1 on page 9) are presented in the order that they typically enumerate and have endpoint addresses assigned:

- Audio (Interfaces 0x10, 0x11, 0x12)
- DIAG/DM (Interface 0x00)
- NMEA (Interface 0x02)
- Modem (Interface 0x03)
- AT (Interface 0x04)
- Raw Data (Interface 0x05)
- RMNET0 (Interface 0x08)
- ECM Control (Interface 0x13)
- ECM Data (Interface 0x14)

Table 2-2: USB descriptors

Descriptor	Field identifier	Value	Description/Notes
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 bytes
	idVendor	0x1199	Sierra Wireless Inc.
	idProduct	0x68C0	WP76xx, WP77xx
	bcdDevice	0x0318	Module variant
	iManufacturer	0x01	Sierra Wireless, Incorporated
	iProduct	0x02	Product name e.g. "WP7702", "Sierra Wireless WP7603", etc.
	iSerialNumber	0x03	*Note: Value contained in USBSERIALENABLE customization (accesible via !CUSTOM AT command).*
	bNumConfigurations	0x01	1 configuration
Configuration Descriptor (Index 0x00)	bLength	0x09	
	bDescriptorType	0x02	
	wTotalLength	0x010A	*Note: Length varies depending on number of interfaces and endpoints.*
	bNumInterfaces	0x0B	*Note: Number of interfaces varies depending on provisioning, and use of !USBCOMP to enable/disable interfaces.*
	bConfigurationValue	0x01	
	iConfiguration	0x00	
	bmAttributes	0xA0	Bus powered—remote wakeup
	MaxPower	0xFA	500 mA

Table 2-2: USB descriptors (Continued)

Descriptor	Field identifier	Value	Description / Notes
Interface Descriptor (Interface # 0x10) Audio	bLength	0x09	
	bDescriptorType	0x04	
	bInterfaceNumber	0x10	
	bAlternateSetting	0x00	
	bNumEndpoints	0x00	
	bInterfaceClass	0x01	Audio
	bInterfaceSubClass	0x01	Audio Control
	bInterfaceProtocol	0x00	
	iInterface	0x0D	“AC Interface”
Interface Descriptor (Interface # 0x11) Audio	bLength	0x09	
	bDescriptorType	0x04	
	bInterfaceNumber	0x11	
	bAlternateSetting	0x01	
	bNumEndpoints	0x01	
	bInterfaceClass	0x01	Audio
	bInterfaceSubClass	0x02	Audio Streaming
	bInterfaceProtocol	0x00	
	iInterface	0x00	
Endpoint Descriptor (Endpoint address 0x81)	bLength	0x09	
	bDescriptorType	0x05	
	bEndpointAddress	0x81	Endpoint ID (1); Direction (IN) *Note: Dynamically allocated. Actual endpoint addresses will vary depending on enabled/disabled interfaces.*
	bmAttributes	0x05	Transfer type: Isochronous; Synchronization type: Asynchronous; Usage type: Data Endpoint
	wMaxPacketSize	0x0020	1 transaction per microframe, max 32 bytes
	wInterval	0x0004	
	bSyncAddress	0x00	
Interface Descriptor (Interface # 0x12) Audio	bLength	0x09	
	bDescriptorType	0x04	
	bInterfaceNumber	0x12	
	bAlternateSetting	0x01	
	bNumEndpoints	0x01	
	bInterfaceClass	0x01	Audio
	bInterfaceSubClass	0x02	Audio Streaming
	bInterfaceProtocol	0x00	
	iInterface	0x13	“AS Interface”

Table 2-2: USB descriptors (Continued)

Descriptor	Field identifier	Value	Description/Notes
Endpoint Descriptor (Endpoint address 0x01)	bLength	0x09	
	bDescriptorType	0x05	
	bEndpointAddress	0x01	Endpoint ID (1); Direction (OUT) *Note: Dynamically allocated. Actual endpoint addresses will vary depending on enabled/disabled interfaces.*
	bmAttributes	0x09	Transfer type: Isochronous; Synchronization type: Adaptive; Usage type: Data Endpoint
	wMaxPacketSize	0x0020	1 transaction per microframe, max 32 bytes
	wInterval	0x0004	
	wSyncAddress	0x00	
Interface Descriptor (Interface # 0x00) DIAG/DM	bLength	0x09	
	bDescriptorType	0x04	
	bInterfaceNumber	0x00	
	bAlternateSetting	0x00	
	bNumEndpoints	0x02	
	bInterfaceClass	0xFF	Vendor-specific device class
	bInterfaceSubClass	0xFF	Vendor-specific device class
	bInterfaceProtocol	0xFF	Vendor-specific
	iInterface	0x00	
Endpoint Descriptor (Endpoint address 0x82)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x82	Endpoint ID (2); Direction (IN) *Note: Dynamically allocated. Actual endpoint addresses will vary depending on enabled/disabled interfaces.*
	bmAttributes	0x02	Transfer type: Bulk
	wMaxPacketSize	0x0200	Max 512 bytes; Must be high speed device
	bInterval	0x00	
Endpoint Descriptor (Endpoint address 0x02)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x02	Endpoint ID (2); Direction (OUT) *Note: Dynamically allocated. Actual endpoint addresses will vary depending on enabled/disabled interfaces.*
	bmAttributes	0x02	Transfer type: Bulk
	wMaxPacketSize	0x0200	Max 512 bytes; Must be high speed device
	bInterval	0x00	
Interface Descriptor (Interface # 0x02) NMEA	bLength	0x09	
	bDescriptorType	0x04	
	bInterfaceNumber	0x02	
	bAlternateSetting	0x00	
	bNumEndpoints	0x03	
	bInterfaceClass	0xFF	Vendor-specific device class
	bInterfaceSubClass	0x00	
	bInterfaceProtocol	0x00	
	iInterface	0x00	

Table 2-2: USB descriptors (Continued)

Descriptor	Field identifier	Value	Description / Notes
Endpoint Descriptor (Endpoint address 0x84)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x84	Endpoint ID (4); Direction (IN) *Note: Dynamically allocated. Actual endpoint addresses will vary depending on enabled/disabled interfaces.*
	bmAttributes	0x03	Transfer type: Interrupt
	wMaxPacketSize	0x000A	1 transaction per microframe, max 10 bytes
	bInterval	0x09	
Endpoint Descriptor (Endpoint address 0x83)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x83	Endpoint ID (3); Direction (IN) *Note: Dynamically allocated. Actual endpoint addresses will vary depending on enabled/disabled interfaces.*
	bmAttributes	0x02	Transfer type: Bulk
	wMaxPacketSize	0x0200	Max 512 bytes; Must be high speed device
	bInterval	0x00	
Endpoint Descriptor (Endpoint address 0x03)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x03	Endpoint ID (3); Direction (OUT) *Note: Dynamically allocated. Actual endpoint addresses will vary depending on enabled/disabled interfaces.*
	bmAttributes	0x02	Transfer type: Bulk
	wMaxPacketSize	0x0200	Max 512 bytes; Must be high speed device
	bInterval	0x00	
Interface Descriptor (Interface # 0x03) Modem	bLength	0x09	
	bDescriptorType	0x04	
	bInterfaceNumber	0x03	
	bAlternateSetting	0x00	
	bNumEndpoints	0x03	
	bInterfaceClass	0xFF	Vendor-specific device class
	bInterfaceSubClass	0x00	
	bInterfaceProtocol	0x00	
	iInterface	0x00	
Endpoint Descriptor (Endpoint address 0x86)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x86	Endpoint ID (6); Direction (IN) *Note: Dynamically allocated. Actual endpoint addresses will vary depending on enabled/disabled interfaces.*
	bmAttributes	0x03	Transfer type: Interrupt
	wMaxPacketSize	0x000A	1 transaction per microframe, max 10 bytes
	bInterval	0x09	

Table 2-2: USB descriptors (Continued)

Descriptor	Field identifier	Value	Description/Notes
Endpoint Descriptor (Endpoint address 0x85)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x85	Endpoint ID (5); Direction (IN) *Note: Dynamically allocated. Actual endpoint addresses will vary depending on enabled/disabled interfaces.*
	bmAttributes	0x02	Transfer type: Bulk
	wMaxPacketSize	0x0200	Max 512 bytes; Must be high speed device
	bInterval	0x00	
Endpoint Descriptor (Endpoint address 0x04)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x04	Endpoint ID (4); Direction (OUT) *Note: Dynamically allocated. Actual endpoint addresses will vary depending on enabled/disabled interfaces.*
	bmAttributes	0x02	Transfer type: Bulk
	wMaxPacketSize	0x0200	Max 512 bytes; Must be high speed device
	bInterval	0x00	
Interface Descriptor (Interface # 0x04) AT	bLength	0x09	
	bDescriptorType	0x04	
	bInterfaceNumber	0x04	
	bAlternateSetting	0x00	
	bNumEndpoints	0x03	
	bInterfaceClass	0xFF	Vendor-specific device class
	bInterfaceSubClass	0x00	
	bInterfaceProtocol	0x00	
	iInterface	0x00	
Endpoint Descriptor (Endpoint address 0x86)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x88	Endpoint ID (8); Direction (IN) *Note: Dynamically allocated. Actual endpoint addresses will vary depending on enabled/disabled interfaces.*
	bmAttributes	0x03	Transfer type: Interrupt
	wMaxPacketSize	0x000C	1 transaction per microframe, max 12 bytes
	bInterval	0x09	
Endpoint Descriptor (Endpoint address 0x85)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x87	Endpoint ID (7); Direction (IN) *Note: Dynamically allocated. Actual endpoint addresses will vary depending on enabled/disabled interfaces.*
	bmAttributes	0x02	Transfer type: Bulk
	wMaxPacketSize	0x0200	Max 512 bytes; Must be high speed device
	bInterval	0x00	

Table 2-2: USB descriptors (Continued)

Descriptor	Field identifier	Value	Description / Notes
Endpoint Descriptor (Endpoint address 0x04)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x05	Endpoint ID (5); Direction (OUT) *Note: Dynamically allocated. Actual endpoint addresses will vary depending on enabled/disabled interfaces.*
	bmAttributes	0x02	Transfer type: Bulk
	wMaxPacketSize	0x0200	Max 512 bytes; Must be high speed device
	bInterval	0x00	
Interface Descriptor (Interface # 0x05) Raw Data	bLength	0x09	
	bDescriptorType	0x04	
	bInterfaceNumber	0x05	
	bAlternateSetting	0x00	
	bNumEndpoints	0x03	
	bInterfaceClass	0xFF	Vendor-specific device class
	bInterfaceSubClass	0x00	
	bInterfaceProtocol	0x00	
	iInterface	0x00	
Endpoint Descriptor (Endpoint address 0x88)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x8A	Endpoint ID (A); Direction (IN) *Note: Dynamically allocated. Actual endpoint addresses will vary depending on enabled/disabled interfaces.*
	bmAttributes	0x03	Transfer type: Interrupt
	wMaxPacketSize	0x000A	1 transaction per microframe, max 10 bytes
	bInterval	0x09	
Endpoint Descriptor (Endpoint address 0x87)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x89	Endpoint ID (9); Direction (IN) *Note: Dynamically allocated. Actual endpoint addresses will vary depending on enabled/disabled interfaces.*
	bmAttributes	0x02	Transfer type: Bulk
	wMaxPacketSize	0x0200	Max 512 bytes; Must be high speed device
	bInterval	0x00	
Endpoint Descriptor (Endpoint address 0x05)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x06	*Note:* Endpoint ID (6); Direction (OUT) *Dynamically allocated. Actual endpoint addresses will vary depending on enabled/disabled interfaces.*
	bmAttributes	0x02	Transfer type: Bulk
	wMaxPacketSize	0x0200	Max 512 bytes; Must be high speed device
	bInterval	0x00	

Table 2-2: USB descriptors (Continued)

Descriptor	Field identifier	Value	Description/Notes
Interface Descriptor (Interface # 0x08) RMNET0	bLength	0x09	
	bDescriptorType	0x04	
	bInterfaceNumber	0x08	
	bAlternateSetting	0x00	
	bNumEndpoints	0x03	
	bInterfaceClass	0xFF	Vendor-specific device class
	bInterfaceSubClass	0xFF	
	bInterfaceProtocol	0xFF	
	iInterface	0x07	"rmnet-qmap-0"
Endpoint Descriptor (Endpoint address 0x8A)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x8C	Endpoint ID (C); Direction (IN) *Note: Dynamically allocated. Actual endpoint addresses will vary depending on enabled/disabled interfaces.*
	bmAttributes	0x03	Transfer type: Interrupt
	wMaxPacketSize	0x0008	1 transaction per microframe, max 10 bytes
	bInterval	0x09	
Endpoint Descriptor (Endpoint address 0x89)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x8B	Endpoint ID (B); Direction (IN) *Note: Dynamically allocated. Actual endpoint addresses will vary depending on enabled/disabled interfaces.*
	bmAttributes	0x02	Transfer type: Bulk
	wMaxPacketSize	0x0200	Max 512 bytes; Must be high speed device
	bInterval	0x00	
Endpoint Descriptor (Endpoint address 0x06)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x07	Endpoint ID (7); Direction (OUT) *Note: Dynamically allocated. Actual endpoint addresses will vary depending on enabled/disabled interfaces.*
	bmAttributes	0x02	Transfer type: Bulk
	wMaxPacketSize	0x0200	Max 512 bytes; Must be high speed device
	bInterval	0x00	
Interface Descriptor (Interface # 0x13) ECM Control	bLength	0x09	
	bDescriptorType	0x04	
	bInterfaceNumber	0x13	
	bAlternateSetting	0x00	
	bNumEndpoints	0x01	
	bInterfaceClass	0x02	Communications (CDC Control) USB Device Interface Class
	bInterfaceSubClass	0x06	
	bInterfaceProtocol	0x00	
	iInterface	0x08	"CDC Ethernet Control Model (ECM)"

Table 2-2: USB descriptors (Continued)

Descriptor	Field identifier	Value	Description / Notes
Endpoint Descriptor (Endpoint address 0x8C)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x8D	Endpoint ID (D); Direction (IN) *Note: Dynamically allocated. Actual endpoint addresses will vary depending on enabled/disabled interfaces.*
	bmAttributes	0x03	Transfer type: Interrupt
	wMaxPacketSize	0x0010	1 transaction per microframe, max 10 bytes
	bInterval	0x09	
Interface Descriptor (Interface # 0x14) ECM Data	bLength	0x09	
	bDescriptorType	0x04	
	bInterfaceNumber	0x14	
	bAlternateSetting	0x01	
	bNumEndpoints	0x02	
	bInterfaceClass	0x0A	CDC Data USB Device Interface Class
	bInterfaceSubClass	0x00	
	bInterfaceProtocol	0x00	
	iInterface	0x0A	“CDC Ethernet Data”
Endpoint Descriptor (Endpoint address 0x8B)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x8E	Endpoint ID (E); Direction (IN) *Note: Dynamically allocated. Actual endpoint addresses will vary depending on enabled/disabled interfaces.*
	bmAttributes	0x02	Transfer type: Bulk
	wMaxPacketSize	0x0200	Max 512 bytes; Must be high speed device
	bInterval	0x00	
Endpoint Descriptor (Endpoint address 0x07)	bLength	0x07	
	bDescriptorType	0x05	
	bEndpointAddress	0x08	Endpoint ID (8); Direction (OUT) *Note: Dynamically allocated. Actual endpoint addresses will vary depending on enabled/disabled interfaces.*
	bmAttributes	0x02	Transfer type: Bulk
	wMaxPacketSize	0x0200	Max 512 bytes; Must be high speed device
	bInterval	0x00	

USB Enumeration Examples

As noted earlier, endpoint addresses are dynamically allocated as each USB interface enumerates.

The following table shows the endpoint allocations for a module that has all supported USB interfaces enabled, and for a module that does not have Raw Data (Interface 0x05) or Audio (Interfaces 0x10, 0x11, 0x12) enabled.

Table 2-3: Endpoint Allocation Examples

Direction: IN				Direction: OUT		
Endpoint Address	Interfaces (0x##)			Endpoint Address	Interfaces (0x##)	
	00, 02, 03, 04, 05, 08, 10, 11, 12, 13, 14	00, 02, 03, 08, 13, 14			00, 02, 03, 04, 05, 08, 10, 11, 12, 13, 14	00, 02, 03, 08, 13, 14
x81	Audio (0x11) Isochronous	DIAG (0x00) Bulk		x01	Audio (0x12) Isochronous	DIAG (0x00) Bulk
x82	DIAG (0x00) Bulk	NMEA (0x02) Bulk		x02	DIAG (0x00) Bulk	NMEA (0x02) Bulk
x83	NMEA (0x02) Bulk	NMEA (0x02) Interrupt		x03	NMEA (0x02) Bulk	Modem (0x03) Bulk
x84	NMEA (0x02) Interrupt	Modem (0x03) Bulk		x04	Modem (0x03) Bulk	RMNET0 (0x08) Bulk
x85	Modem (0x03) Bulk	Modem (0x03) Interrupt		x05	AT (0x04) Bulk	ECM (CDC Data) (0x14) Bulk
x86	Modem (0x03) Interrupt	RMNET0 (0x08) Bulk		x06	Raw Data (0x05) Bulk	
x87	AT (0x04) Bulk	RMNET0 (0x08) Interrupt		x07	RMNET0 (0x08) Bulk	
x88	AT (0x04) Interrupt	ECM (CDC Data) (0x14) Bulk		x08	ECM (CDC Data) (0x14) Bulk	
x89	Raw Data (0x05) Bulk	ECM (CDC Control) (0x13) Interrupt		x09		
x8A	Raw Data (0x05) Interrupt			x0A		
x8B	RMNET0 (0x08) Bulk			x0B		
x8C	RMNET0 (0x08) Interrupt			x0C		
x8D	ECM (CDC Data) (0x14) Bulk			x0D		
x8E	ECM (CDC Control) (0x13) Interrupt			x0E		
x8F				x0F		