SIERRA WIRELESS

CMUX Feature
2170036

Author:	Sierra Wireless	Date:	March 7, 2012

APN Content Level	BASIC		INTERMEDIATE		ADVANCED	✓	Confidentiality	Public	✓	Private	

Hardware Compatibility	Product Line	AirPrime	Series	Q26xx	SL60xx
				Q26Extreme	Q24xx
				WMPxx	Q64

Software Compatibility	Series	Q26xx : >6.X	Q24xx :X.5X
		OTHERS : ALL	

1 Version

Application Notes may be updated over their lifetime. To ensure you design with the correct version, please check the application notes page in www.sierrawireless.com for latest versions.

2 Introduction

This Application Note (APN) is provided to Sierra Wireless distributors and clients to aid more rapid development of embedded applications using the Sierra Wireless portfolio of cellular solutions. To request a new application note, contact your regional Sierra Wireless Product Marketing Manager.

3 Overview

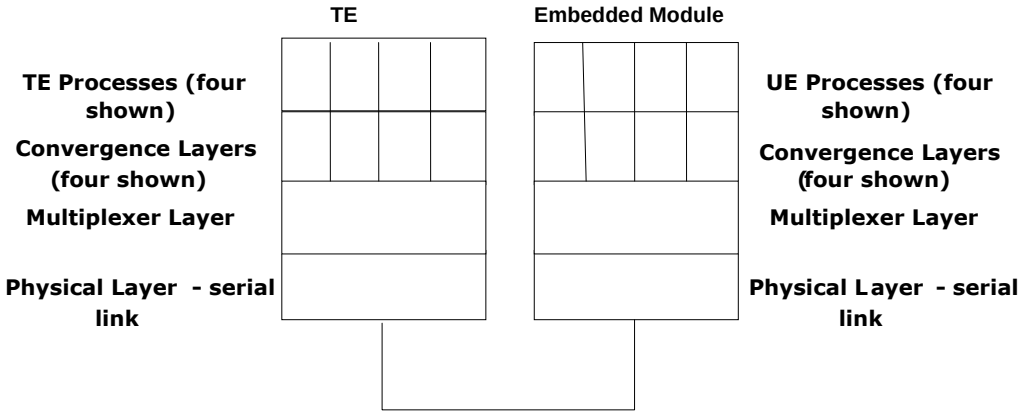
This document gives an overview of the CMUX feature as proposed by 3GPP TS 27.010 for multiplexing data over serial link. This document also describes the Siera Wireless implementation of CMUX.

The CMUX feature allows multiple virtual ports to be created on a physical serial link. These ports can be used for maintaining several simultaneous sessions (SMS, Call, GPRS etc) on the same serial interface. The multiplexing procedure for achieving this feature is described in 3GPP TS 27.010.

4 Glossary

Initials	Definition
DLC	Access Point Name
HDLC	ISO standard for data transfer
TE	Terminal equipment. External application is referred as terminal equipment
UE	Point to Point Protocol

5 Architecture



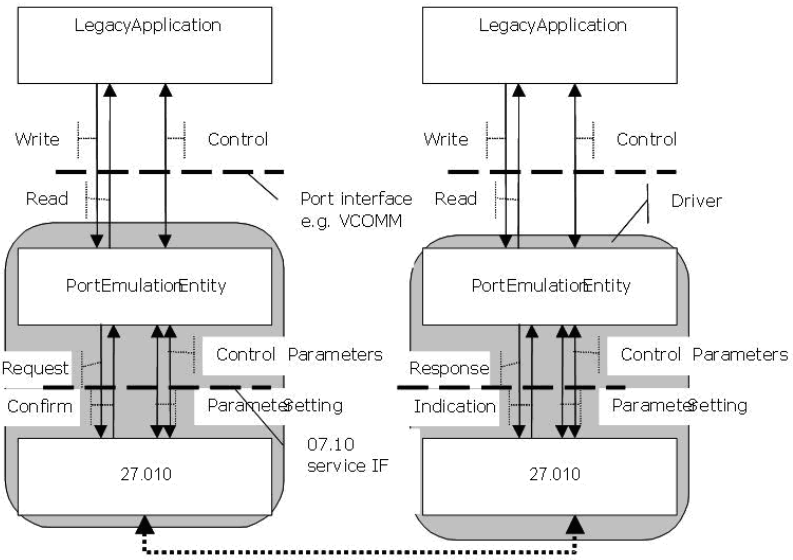
As shown in the figure above, a virtual connection is established between TE and UE processes during a CMUX session. For example, the TE SMS process communicates independently with a UE process having SMS functionality.

Communication is established using a multiplexing layer. This layer opens multiple channels over a single serial link. These channels are called data link connection (DLC). Convergence layers are used to carry the state of V.24 control signals through a DLC.

Note: UE refers to the Sierra Wireless embedded module.

6 CMUX Functionality - 3GPP TS 27.010

6.1. Model Proposed by TS 27.010



As shown in the figure above, the legacy application can send AT commands/Data through a conventional but virtual serial port communication interface. It can also control the Port Interface (open, close, set …). This virtual port maps to a system specific communication interface (Port Emulation Entity) that gets use of 27.010 services. The 27.010 layer provides several transparent data stream channels and a control channel. These channels are referred to as DLCs.

The design of the 27.010 layer is flexible and independent of UE/TE platforms, and allows existing applications to work without any modifications. However, additional upper entity (‘port emulation entity’) is required so that both the UE and TE can get use of CMUX DLCs.

The maximum numbers of channels (DLC) that can be opened are limited to 61 as per 3GPP TS 27.010 modes of operation.

The 27.010 layer has three different modes of operation based on the ISO HDLC standard. These modes can be selected when the session is opened.

6.1.1. Basic Mode

ISO HDLC standard transparency mechanism is not used in basic mode. This mode is used when the link is very good between UE and TE. This mode is also used when the HDLC transparency mechanism cannot be implemented in UE. This mode cannot be used with links which use software flow control (Xon/ Xoff).

6.1.2. Advanced Mode without Error Recovery

This mode uses an ISO HDLC standard transparency mechanism. This mode is used with links which use software flow control (Xon/ Xoff). This mode does not perform any error recovery mechanism.

6.1.3. Advanced Mode with Error Recovery

This mode is the same as advanced mode except that the error recovery mechanism is implemented for this mode.

6.2. CMUX Procedures

The procedures presented in the subsections below are used to:

- Open the Session
- Open a Channel (DLC)
- Transfer data using CMUX.

6.2.1. Session Establishment Procedure

The following steps are required to establish a CMUX connection:

1. The port emulation entity of the TE sends a connection **request** to the ‘27.010 layer’ with a set of configuration parameters. These configuration parameters include port speed, modes of operation etc.
2. The ’27.010 layer’ passes the configuration parameters to the ’27.010’ layer of the target device (UE).
3. The ’27.010 layer of target device’ sends these configuration parameters to the ‘port emulation entity of the target’. This is sent as an **indication.**
4. The ‘port emulation entity of the target’ sets the configuration parameters and opens the session. If the session was successfully opened, it sends a **response** to the ’27.010 layer of the target device’
5. This information is passed to the ’27.010 layer of the TE’. TE sends a **confirm** message to the ‘port emulation entity’ to confirm that the session has been established successfully.

The above steps are also shown in **Error! Reference source not found.**.

6.2.2. Channel (DLC) Establishment Procedure

This procedure is applicable only when the CMUX session is opened using the CMUX session procedure. The following steps are required to establish channels:

1. UE sends a **request** to the 27.010 layer to open a channel. The request contains system parameters such as priority, mode of operation etc. This information is passed to the ’27.010 layer of the target’.
2. The ’27.010 layer of the target’ sends an **indication** to the ‘port emulation entity of the target’.
3. The ‘port emulation entity of the target’ sends a **response** for accepting/rejecting this request to ’27.010 layer’. This is based on the system parameters sent by the UE. If rejected, the 'port emulation entity of the target’ can suggest the modified parameters. The response sent by ‘port emulation entity of the target’ is then passed to the ’27.010 layer of the UE’.
4. The ’27.010 layer of the UE’ sends a **confirm** message to the ‘port emulation entity of the UE’.

6.2.3. Data Transfer

The transmitting unit initiates transmission of data for the chosen channel (DLC) by means of the **request**. The transmitted data is delivered to the upper layer by an **indication** message. There is no **response** or **confirm** during data transfer.

6.2.4. DLC Release Procedure

1. DTE sends a DLC release **request** to the '27.010 layer'.
2. '27.010 layer' sends this request as **indication** to the target device
3. Target device and '27.010 layer' release the DLC.

6.2.5. Session Termination Procedure

1. A close down service **request** is sent from DTE to '27.010 layer' to close the multiplexer session.
2. '27.010 layer' sends this request as **indication** to the target device.
3. Target device terminates the session.

6.3. Frames

As CMUX uses virtual ports over a single serial link, the data sent from virtual ports is encapsulated in a frame and sent to the target device.

These frames are constructed by '27.010 layer of TE' and are decoded by '27.010 layer of UE'. This section describes the different frames managed by '27.010 layer'. This includes frames for session initiation, data transfer, and channel establishment.

6.3.1. Frame for Basic Mode of Operation

Flag	Address	Control	Length Indicator	Information	FCS	Flag
1 byte	1 byte	1 byte	1 or 2 bytes	Unspecified length but integral number of bytes	1 byte	1 byte

- **Flag:** In basic mode, this field is set to '10011111'.
- **Address:** This byte specifies DLC identification for e.g. DLC1 and the direction of the frame. The direction can be read or write. The maximum number of channels that are allowed is implementation specific.
- **Control:** This octet specifies the operations that need to be performed. This can be:
  - Data transfer(AT commands, responses etc) operation
  - Control operations such as connection, disconnection.
- **Length Indicator:** This field specifies the length of information field.
- **Information:** This byte-structured field carries the actual data.
- **FCS:** This field contains the checksum for the information field. The algorithm for calculation of the checksum is mentioned in TS 27.010 section 5.2.1.6.

6.3.2. Frame for Advanced Mode of Operation

Flag	Address	Control	Information	FCS	Flag
1 byte	1 byte	1 byte	Unspecified length but integral number of bytes	1 byte	1 Byte

- **Flag:** In advanced mode, this field is set to '01111110'.
- **Address:** This byte specifies DLC identification for e.g. DLC1 and the direction of the frame. The direction can be read or write. The maximum number of channels that are allowed is implementation specific.
- **Control:** This byte specifies the operations that need to be performed. These can be:
  - Data transfer (AT commands, responses etc) operation
  - Control operations such as connection, disconnection.

  These operations are identified by different types of frames such as SABM, UIH, DISC, etc.
- **Information:** This byte-structured field carries the actual data.
- **FCS:** This field contains the checksum for the information field. The algorithm for calculation of the checksum is mentioned in TS 27.010 section 5.2.1.6

6.3.3. Different Types of Frames

- **Set Asynchronous Balanced Mode (SABM) command:** This command is used to request the target to establish a link so that data can be sent on the link.
- **Unnumbered Acknowledgement (UA) response:** The station acknowledges the SABM and DISC commands using the UA response.
- **Disconnect (DISC) command:** This command is used to terminate a session.
- **Disconnected Mode (DM) response:** This is an acknowledgement to the DISC command.
- **UIH (Unnumbered information with header check) command and response:** This command is used to transfer information without using sequence numbers. The checksum is only calculated for the address part hence this command is used where there is no need to verify the integrity of actual data.
- **UI (Unnumbered information) command and response:** This command is used to transfer information without using sequence numbers. This command uses the checksum for the entire field including data. Support of this frame is optional.

6.4. Convergence Layers

This layer is used to convey the information that is needed to maintain accuracy. This includes the state of the V24 control signal such as DTR, CTS etc. The control information is transmitted along with the information field. Please see frame structure (5.4) for more details.

6.4.1. Type 1 – Unstructured byte stream

This layer is used to transfer data or voice over the channels where there is no need to convey the control signals along with information. This is the default layer for each DLC. This layer can be used when the link is of good quality.

6.4.2. Type 2 – Unstructured octet stream with flow control signals

This layer is used to transfer data along with flow control information such as DTR, RTS etc. The flow control information is transmitted in the first byte of the information field. The coding of the flow control byte is defined in TS 27.010.

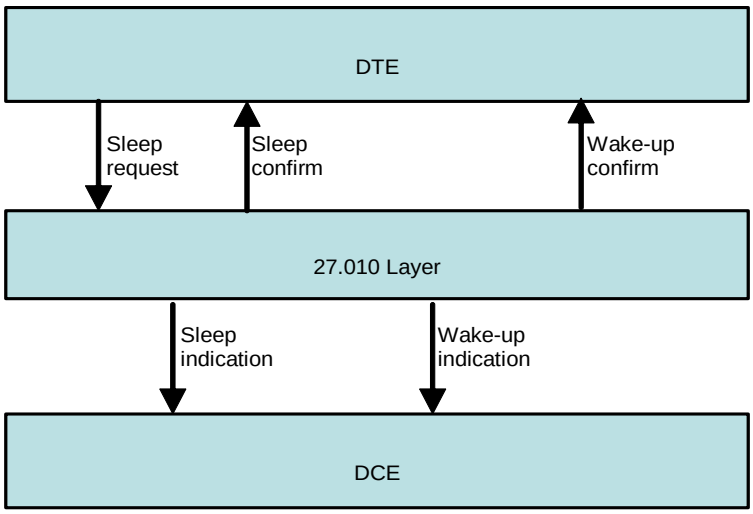
6.4.3. Type 3 - Uninterruptible Framed Data

This layer is used to transmit data in terms of coded voice data. The data is embedded in a single UI or UIH frame. This frame should reach the voice decoder with the shortest possible delay

6.4.4. Type 4 - Interruptible Framed Data

This layer is used to convey information in several frames. This is useful in the case where delay is not important. The several frames sent from the receiver are re-assembled at the receiver.

6.5. Power Control Mechanism



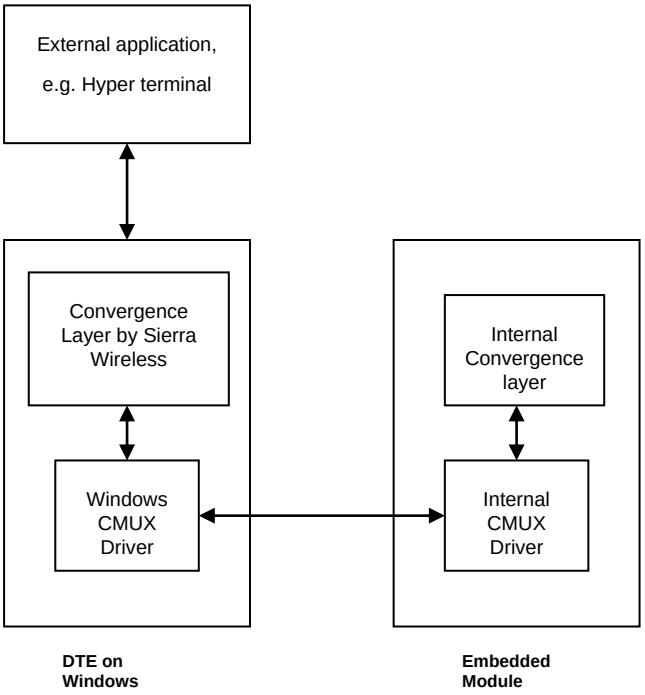
- **Procedure for Sleep Mode:** The DTE can enter sleep mode by sending a sleep request to the 27.010 layer. '27.010 layer' sends indication to the target device to enter sleep mode. No 'response' message is needed from DCE. '27.010 layer of target'. A 'confirm' message is sent directly from the '27.010 layer' to DTE.
- **Wake-up procedure:** DTE requests '27.010 layer' to wakeup from the sleep state. No message (request) needs to be sent for wake-up. A wake-up indication is sent from the '27.010 layer' to DCE and '27.010 layer' confirms the same to DTE.

7 Sierra Wireless Implementation

7.1. Introduction

Sierra Wireless has performed CMUX layer (27.010) implementation in Sierra Wireless embedded module (DCE) and driver implementation in Windows (DTE). The limitation and differences between the proposed 3GPP implementation and Sierra Wireless implementation are listed in the section 6.4.

7.2. Architecture



The figure above shows one of the implementations performed by Sierra Wireless for the CMUX driver. The Sierra Wireless embedded module supports the CMUX driver, but a driver is also needed at the DTE level to support the multiplexing protocol.

As shown above, the external application communicates with '27.010 layer'.

This layer has two components:

- **CMUX driver** – This communicates with CMUX driver of the Sierra Wireless embedded module and exchanges the frames
- **Convergence layer** – This communicates with the convergence layer of the Sierra Wireless embedded module through the CMUX driver and manages control information such as Flow control. These signals are handled virtually by the MSC frame for each DLC.

Sierra Wireless has implemented these two components for a Windows machine. These components can also be implemented for other architectures.

7.3. Software Architecture

7.3.1. Sierra Wireless Embedded CMUX Implementation

1. **Mode of operation:** The 3GPP TS 27.010 specifies three different operating options : basic, advanced without error recovery and advanced with error recovery. Sierra Wireless CMUX implementation operates in basic mode and supports convergence layer type I. Thisrenders the driver extremely rapid. The CMUX management is performed in the ATI and V24 elements of the Sierra Wireless software.
2. **Session establishment procedure:** TE starts the CMUX service with the AT+CMUX command and sends the SABM0 start frame to the UE to initiate the multiplexing service.
3. **Reset strategy**: When the UE embedded module is reset, it reopens the former CMUX session configuration (AT+CMUX parameters and number of opened DLCI). When resetting, the embedded module sends a protocol test frame to check if the TE is still in CMUX mode. If not, the UE will start normally on the physical UART.
4. **DLC establishment / release:** DLC opening/release is initiated by the TE which sends the protocol frame concerned to the embedded module.
5. **Session termination procedure:** The TE should individually close each DLC and finally stop the CMUX. The TE could send a close down protocol frame to end the session. After CMUX session has stopped, the embedded module will return to its normal mode (AT mode) on the physical UART.
6. **Test service**: When the embedded module resets in CMUX mode, the test string payload sent by the embedded module contains "MUX_TEST_STRING", and this means that the embedded module has reset. TE tests the embedded module by sending a protocol test frame and the embedded module replies as specified by the protocol.
7. **Power Control and Wake Up mechanism**: This mechanism is closely related to Sierra Wireless Sleep mode behavior:it uses DTR signal (here with virtual signals) and a TS 27.010 subset (“Power Control and Wake-up Mechanisms”).

7.4. Limitations

Sierra Wireless CMUX implementation specificities are the following:

1. Only a single MUX session, either on UART1 or UART2, is supported.
2. MUX will only handle up to 4 DLCs for the 3 physical ports, instead of 61 per physical port.
3. Flow control is less reactive than on a real hardware UART controller. The application should size its reception buffers correctly and carefully manage the transmit data flow.
4. Services not implemented:
    a. No DLC Service Negotiation service
    b. No Remote DLC parameter negotiation service
5. Miscellaneous:
    a. No DLC Id specified by user for DLC open/close (it is automatically managed by the MUX driver)
    b. No individual N1,N2,T1,T2,T3 parameter management for a DLC (the same parameters are applied to all opened DLCs on the same multiplexer)
    c. No individual DLC rate and framing parameter management (the rate and framing on the physical port is applied to all opened DLCs on it)

7.5. AT Commands

7.5.1. AT+CMUX Command

The CMUX driver session can be started by using the AT command AT+CMUX. The CMUX session can be stopped by sending the protocol frame DISC. This command can also be used to configure CMUX driver parameters such as:

- Mode of operation (Only basic mode is currently supported)
- Type of the convergence layer (only UIH frames are supported)
- Port speed
- Control information related to frames, such as frame size, response timer values etc.

7.5.2. AT+WMFM Command

This command is used to configure CMUX for various flows including:

- GSM data flow
- GPRS data flow
- AT commands

7.5.3. UART Configuration Commands

DLCs can be configured using commands AT+IPR and AT+ICF to virtually set speed and framing. These commands will not have any effects on the physical UART.

7.6. APIs

7.6.1. AT Commands and Responses for CMUX

AT command APIs are available for to use the CMUX feature in the Open AT Framework. The following operations can be achieved using these APIs:

- Send AT commands from virtual CMUX ports
- Capture AT responses for virtual CMUX ports

7.6.2. DATA Transfer for CMUX

Flow control manager APIs provide a mechanism to send/receive data to/from CMUX virtual ports.

7.6.3. Control Information

The AT/FCM IO ports service can be used to monitor the V24 signals from the application. These signals include polling of pins such as DTR, CTS.

7.7. Examples

7.7.1. Using CMUX with a Windows PC and the Sierra Wireless Embedded Module

The CMUX driver comes with the following files:

- **Mux_ConfTool.exe**: Utility to launch CMUX
- **muxgsm2710.reg:** Registry settings for the CMUX driver
- **muxgsm2710.sys:** Driver file for CMUX

This section describes the steps which may be followed to configure the CMUX driver.

7.7.1.1. Step 1: Configuration of the CMUX driver for Windows

Install the CMUX driver by clicking on the setup.exe file provided.

7.7.1.2. Step 2: CMUX Configuration Using AT Command

Use the AT+CMUX command to configure the CMUX session related parameters (Refer to AT commands guide for more information on +CMUX command use).

7.7.1.3. Step 3: Launching CMUX

These steps should be performed within 30 seconds after opening the session as per step 2.

1. Launch Mux_ConfTool.exe
2. Select physical COM port (for e.g. COM1)
3. Open CMUX using the ‘‘Open MUX’’ button. This also sends an AT+CMUX=0 command to open the CMUX session on the embedded module side.
4. Open the DLCs using the “Open DLC” button. This will open virtual COM ports (For e.g. COM3, COM4) on the physical serial port.

These virtual ports can now be used by any external application e.g. Hyperterminal .

7.7.2. CMUX Use with AT Commands

1. Open four DLCs as explained above.
2. The four DLCs can now be used as explained below to take advantage of CMUX feature
    a. DLC1: GPRS data connection (browsing, FTP etc)
    b. DLC2: send/receive
    c. DLC3: Phonebook management

d. DLC4: For configuration using AT commands

7.7.3. CMUX with TCP/IP stack and APIs

1. Open three DLCs as explained above.
2. The three DLCs can now be used as explained below to take advantage of CMUX feature
   a. DLC1: SMTP (sending emails)
   b. DLC2: TCP/IP FTP feature
   c. DLC3: TCP sockets

8 Software Compatibility Matrix

List all current software configurations and compatibility with this application note.

FW	Open AT Framework/SDK	Open AT Framework Libraries
6.5x	Open AT® SDK v3.X	N/A
6.6x	Open AT® SDK v4.X	N/A
R6.5	Open AT® Software Suite v1.X	N/A
R7.x	Open AT Framework v2.X	N/A

9 Reference Documents

	Filename	Comment
[1]	3GPP TS 27.010	CMUX protocol specifications

10 Support

For direct clients: contact your Sierra Wireless FAE

For distributor clients: contact your distributor FAE

For distributors: contact your Sierra Wireless FAE

11 Document History

Version	Date	History
001	August 11 , 2006	Creation
002	January 1, 2007	Incorporated Review comments
003	May 7, 2007	Updated
004	March 3, 2010	Updated
5.0	March 7, 2012	Updated legal boilerplate contents New reference: 2170036 Old reference: WM_DEV_OAT_APN_003

12 Legal Notice

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

Limitations of Liability

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

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

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

Patents

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

This product includes technology licensed from QUALCOMM®.

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

Copyright

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.

wavecom®, , , ®, inSIM®, WAVECOM®, WISMO®, Wireless Microprocessor®, Wireless CPU®, Open AT® are filed or registered trademarks of Sierra Wireless S.A. in France and/or in other countries.

Windows® and Windows Vista® are registered trademarks of Microsoft Corporation.

Macintosh and Mac OS 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 the respective owners.