SIERRA WIRELESS
HEART OF THE WIRELESS MACHINE®

# CMUX Feature for Q26 Elite

## WI_DEV_Q26EL_APN_003-001

| Author: | Sierra Wireless | Date: | August 21, 2009 |
|---|---|---|---|

| APN Content Level | BASIC | | INTERMEDIATE | | ADVANCED | ✓ | Confidentiality | Public | ✓ | Private | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|

| Software Compatibility* | FW: | R1A | Open AT® SW Suite: | N/A | Plug-Ins: | TCP/IP | | C-GPS | | Compiler Used: | ADS GCC | Reference Hardware | No |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | Internet | | Bluetooth | | | | Reference Software | Yes |
| | | | M2mpower | | | MQTT | | M2MC | | | | | |

** refer to software compatibility matrix section for more detail*

| Wireless CPU Compatibility | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Plug & Play: | M1306B | | M2106B | | | | | | | | | | | | | | | |
| | Quik (CDMA): | Q2438F | | Q2438R | | CM52 | | | | | | | | | | | | | |
| | | Q2400A | | Q2406A | | Q2406B | | Q2426B | | Q2501B | | | | | | | | | |
| | Quik (GSM): | GR64 | | GS64 | | Q2686H | | Q2687H | | Q26 Elite | ✓ | | | | | | | | |
| | | Q24 Classic | | Q24 Plus | | Q24 Extended | | Q24 Auto | | | | | | | | | | | |
| | Wireless Microprocessor | WMP100 | | WMP150 | | | | | | | | | | | | | | | |

# 1 Overview

This document gives an overview of the CMUX feature as proposed by 3GPP TS 27.010 for multiplexing data over a serial link. This document also describes the Wavecom implementation of CMUX for Q26 Elite.

# 2 Glossary

| Term | Definition |
|---|---|
| DLC | Data link connection. Channels which are used as virtual serial port |
| FCS | Frame checksum |
| GPS | Global Positioning System |
| NMEA | National Marine Electronics Association. NMEA-0183 defines the interface standard used by most GPS manufacturers albeit with extensions to the standard. |
| HDLC | ISO standard for data transfer |
| TE | Terminal equipment. External application is referred as terminal equipment |
| UE | User equipment. Wireless CPU is referred as UE |
| UIH | Unnumbered Information with Header check |

# 3 References

The reference number in brackets is used in this document to identify the reference, for example, [1].

| | |
|---|---|
| 1 | 3GPP TS 27.010 V8.0.0 (2008-12) Technical Specification 3rd Generation Partnership Project; Technical Specification Group Terminals; Terminal Equipment to User Equipment (TE-UE) multiplexer protocol (Release 8). |

# 4 Introduction

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

# 5 Architecture

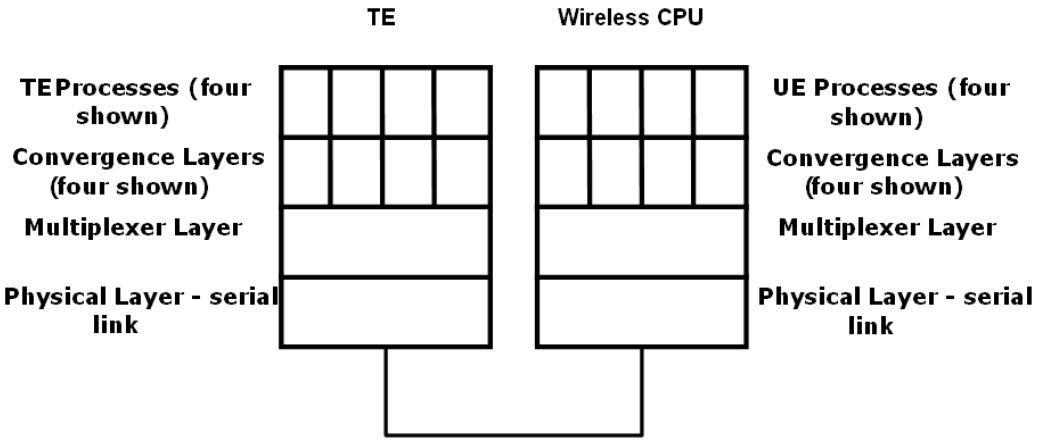


*Figure 1. CMUX Architecture*

The figure above shows a virtual connection is established between TE and UE processes during a CMUX session. For example, a TE GPS process communicates independently with a wireless CPU (the UE) process providing GPS NMEA positioning information.

Communication is established using a multiplexing layer. This layer provides multiple channels, or data link connections (DLCs), over a single serial link. Convergence layers are used to carry the state of V.24 control signals through a DLC.

# 6 CMUX functionality - 3GPP TS 27.010

## Model proposed by TS 27.010

The figure below shows the service model proposed by 27.010. A 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 DLC's and a control DLC.

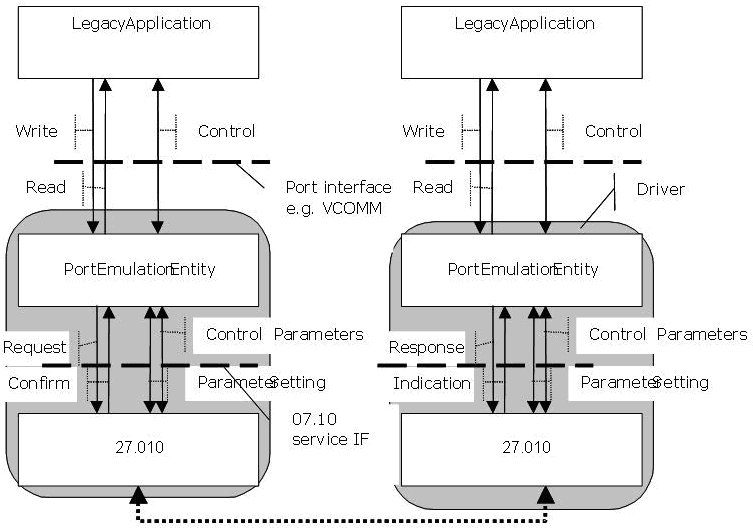


*Figure 2. 27.010 Service Model*

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

## 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.

### 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). Basic mode is implemented in CMUX for the Q26 Elite.

### Advanced mode without error recovery

This mode uses an ISO HDLC standard transparency mechanism. This mode is used with links which uses software flow control (Xon/ Xoff). This mode does not perform any error recovery mechanism. Advanced mode is not implemented in CMUX for the Q26 Elite.

### Advanced mode with error recovery

This mode is the same as advanced mode except that the error recovery mechanism is implemented for this mode. Advanced mode is not implemented in CMUX for the Q26 Elite.

## CMUX Procedures

This section provides a brief overview of the procedures involved in using the CMUX. These procedures are used to

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

### 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 Figure 2.

### Channel (DLC) establishment procedure

This procedure is applicable only when the CMUX session is opened using the CMUX session procedure as described in section 6.3.1. 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'.

### Data transfer procedure

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.

### 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.

### 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

## Frames

As CMUX uses virtual ports over a single serial link, data sent from virtual ports are encapsulated in frames and sent to the target device. The reader is referred to section 5.2 of [1] for a complete frame specification.

*Note: Bit positions in the 27.010 specification are labeled least-significant (bit 1) to most-significant from left to right. In the examples that follow, hexadecimal bytes are displayed with the most-significant bit (bit 8) in the left-most position. This can be confusing when attempting to reconcile the examples with the specification.*

Frames are constructed by '27.010 layer of TE' and are decoded by '27.010 layer of UE'. This section gives a brief overview the different frames managed by '27.010 layer'. This includes frames for session initiation, data transfer, and channel establishment.

### 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. A maximum of 4 channels are allowed in the CMUX implementation for Q26 Elite.
- **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 defined in section 5.2.1.6 of [1].

### 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. UI frames are not supported in CMUX for the Q26 Elite.

## Convergence layers

This layer conveys the information that is needed to maintain accuracy. This includes the state of the V24 control signal such as DTR, CTS etc. The convergence layer type is negotiated when the DLC is opened. The control information is transmitted along with the information field. Please see frame structure in 6.4.1 for more details.

### 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. Type 1 is supported by the CMUX implementation for Q26 Elite.

### 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. Type 2 is not supported by the CMUX implementation for Q26 Elite.

### 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 Type 3 is not supported by the CMUX implementation for Q26 Elite.

### 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. Type 4 is not supported by the CMUX implementation for Q26 Elite.

### Power control mechanism

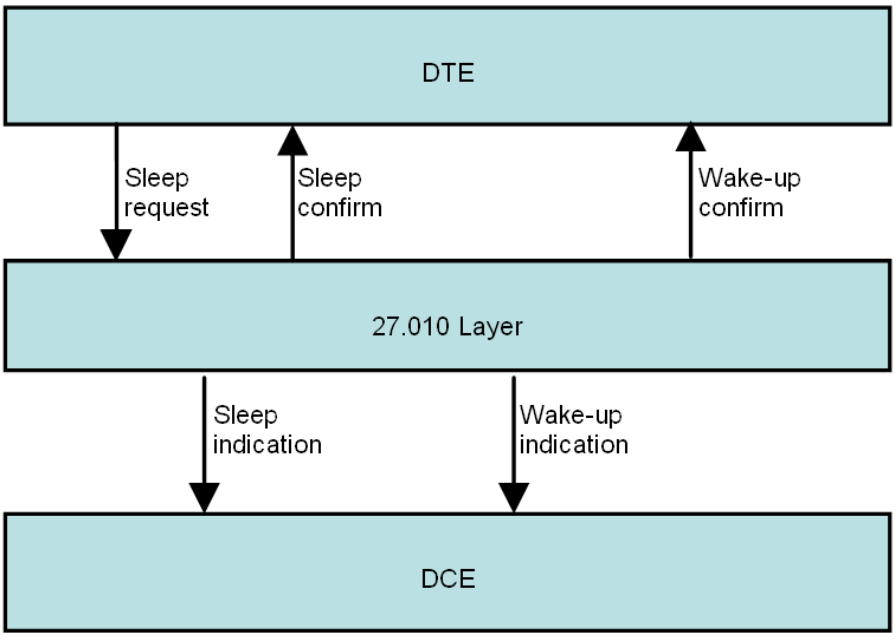


*Figure 3. Power control procedure*

- **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

### Introduction

Sierra Wireless provides a CMUX layer (27.010) implementation in the Q26 Elite Wireless CPU (DCE) and a driver implementation for Microsoft Windows XP (DTE). The limitation and differences between the proposed 3GPP implementation and Sierra Wireless implementations are listed in this application note.

## Architecture

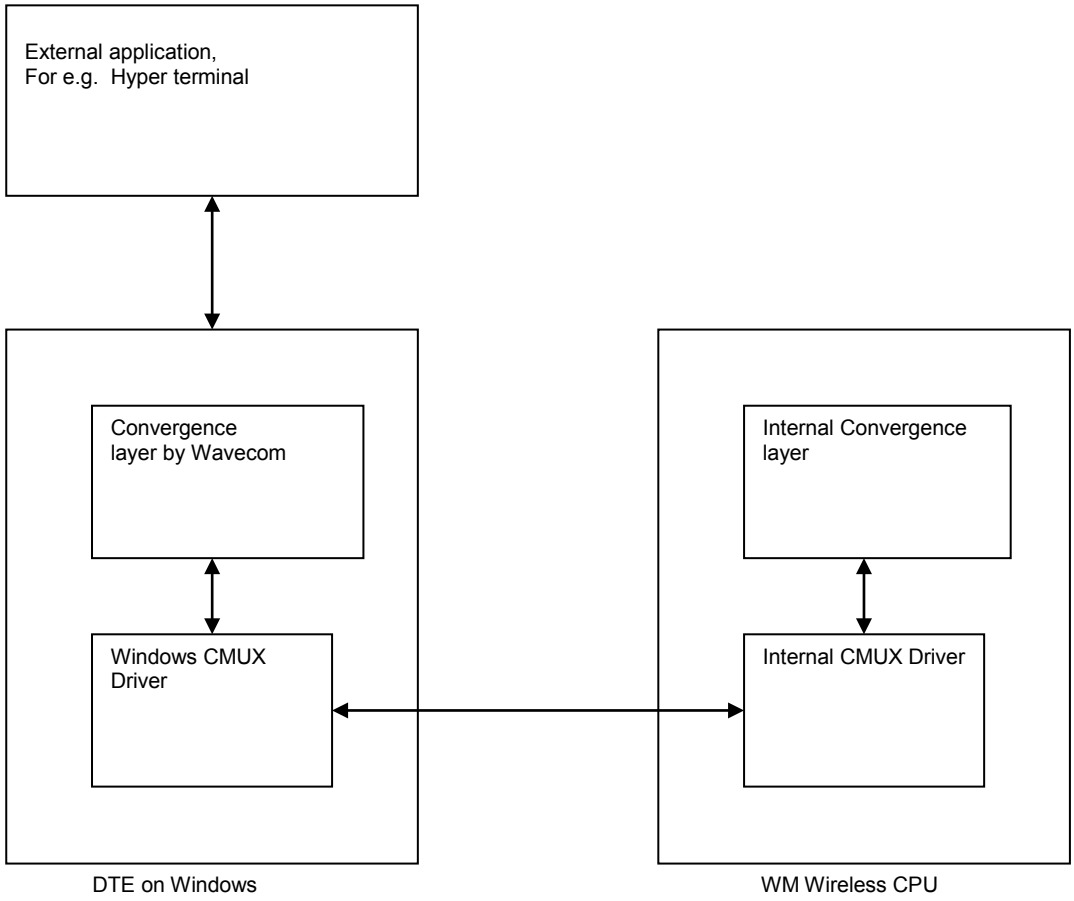


*Figure 4. Architecture of Wavecom implementation of CMUX*

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

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

This layer has two components

- **CMUX driver** – This communicates with CMUX driver of the Wavecom Wireless CPU and exchanges the frames.
- **Convergence layer** – This communicates with the convergence layer of the Wavecom Wireless CPU 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 Microsoft Windows XP environment. These components can also be implemented for other architectures.

## Software Architecture

### 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. This renders the driver extremely rapid.
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 Wireless CPU is reset, it reopens the former CMUX session configuration (AT+CMUX parameters and number of opened DLCI). When resetting, the Wireless CPU 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 Wireless CPU.
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 Wireless CPU will return to its normal mode (AT mode) on the physical UART.
6. **Test service**: When the Wireless CPU resets in CMUX mode, the test string payload sent by the Wireless CPU contains "MUX_TEST_STRING", and this means that the Wireless CPU has reset. TE tests the Wireless CPU by sending a protocol test frame and the Wireless CPU replies as specified by the protocol.

### 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”).

## Limitations

The following limitations apply to the Sierra Wireless CMUX implementation for Q26 Elite:

1. Only a single MUX session on UART1 is supported.
2. MUX will only handle up to 3 DLC’s. Each DLC carries traffic as follows:
   a. DLC 1 – AT commands and unsolicited reports
   b. DLC 2 – data that would normally be transferred during an online data session (ATD#777).
   c. DLC 3 – GPS NMEA sentences
3. TCP/UDP Socket Commands should not be used while CMUX is active.
4. 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.
5. Services not implemented:
   a. No DLC Service Negotiation service
   b. No Remote DLC parameter negotiation service
6. 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)

### AT commands

#### 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 (See 6.4.2). 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.

## 8 Building Applications using CMUX

We strongly encourage you read and understand the 27.010 MUX protocol specification [1] before attempting to design your application. To help you get familiar with CMUX, Sierra Wireless provides the PC Mux Pack that allows you to exercise the Q26 Elite CMUX capabilities.

### Testing with the PC Mux Pack

The PC Mux Pack contains the Windows CMUX driver, utilities and installation files. When installed and configured, the CMUX driver will create virtual COM ports that allow you to use multiple Windows applications such as HyperTerminal to communicate with the Q26 Elite over a single UART.

#### Installing the PC Mux Pack

This section describes the steps to install and configure the CMUX driver and utilities.

1. Unzip the PC Mux Pack distribution to a temporary directory. Be sure to read the ReadmeCmux.txt file as it may contain important information. Double-click on the setup.exe program to begin the installation. We recommend accepting the default installation parameters.
2. The installation procedure will install the Windows CMUX driver and the MUX configuration tool, MuxConfTool.
3. When the installation is complete, run MuxConfTool from the Start menu.
4. Select the COM port to which your Q26 Elite UART1 is connected from the pulldown list then click on “Select Com.”
5. Click on “Open Mux,” keep the suggested settings and click on “OK.” Click on “OK” to dismiss the success dialog.

The CMUX driver has successfully communicated with the Q26 Elite CMUX implementation and DLC 0 is established.

#### AT Commands

AT command entries and responses are transferred over DLC 1 when the MUX is active. Use these steps to open DLC 1 and establish a terminal session for AT commands.

1. Click on “Open DLC,” select 1 from the DLCI pulldown list, then click on “OK.” Click on “OK” to dismiss the success dialog.
2. Run HyperTerminal or other suitable terminal emulator program and connect to the COM port associated with DLC 1. By default, DLC 1 is associated with COM11. You should be able to enter AT commands to the Q26 Elite and receive responses.

In most cases, the default COM ports are suitable for your PC configuration. However, if you have difficulty communicating opening or communicating over DLC 1, you should examine the COM ports defined for your PC. Select

Start..Settings..Control Panel..System, then the Hardware tab and Device Manager. Examine the Ports to find any that conflict with those defined by MuxConfTool, by default COM11 through COM 14. You can redefine which COM ports that MuxConfTool uses by editing the values of ComForDLC in C:\Program Files\Wavecom\MuxConfTool\MuxConfTool.ini.

### Data stream

During non-MUX operation, the Q26 Elite enters online data mode when a data connection is established, using the ATD#777 command. In online data mode, all traffic on UART1 of the Q26 Elite is protocol data packets, typically PPP for 1xRTT data.

When the MUX is active, data traffic is present over DLC 2. To establish the connection, the ATD#777 command is entered on DLC 1 (or DLC 2 to support Windows Dialup Network interface), the "CONNECT" response will appear on both DLC 1 and 2, at which point data can be transmitted or received on DLC 2.

Demonstrating the use of the data stream over DLC 2 is beyond the scope of this application note.

### GPS position information

GPS positioning information in NMEA format is available on DLC 3.

Ensure that the Q26 Elite is configured with a GPS antenna and is receiving GPS signals. You can use the AT+WPOS command to obtain a single GPS position and verify that GPS is working properly. Then follow these steps:

1. Click on "Open DLC," select 3 from the DLCI pulldown list, then click on "OK." Click on "OK" to dismiss the success dialog.
2. Run HyperTerminal or other suitable terminal emulator program and connect to the COM port associated with DLC 3. By default, DLC 3 is associated with COM13.
3. Enter "AT+WNMEA=1,3" on the terminal emulator attached to DLC 1. After a few seconds, you should see the GPS position data in NMEA format continuously displayed on the terminal emulator associated with DLC 3. To stop the display, enter "AT+WNMEA=0" on DLC 1.

## Observing the MUX protocol

Once you have successfully communicated with the Q26 Elite using the PC Mux Pack, you can use one of the publicly available Windows serial port analyzers such as Free Serial Port Monitor to examine and analyze the 27.010 protocol packets exchanged between the PC (TE) and the Q26 Elite (ME). This exercise gives you valuable insight into how the Q26 Elite MUX implementation communicates using the 27.010 protocol. Install and configure the serial port analyzer to monitor the PC serial port connected to your Q26 Elite UART1.

If your application does not use the Windows environment, there may be a serial port monitor that is supported in your application's environment. Alternatively, there are hardware-based serial port monitors or low-cost logic analyzers that can monitor the RS232 signal lines of UART1.

## Example MUX transactions

In the following examples, MuxConfTool is used to open and close the CMUX and open and close DLC's. HyperTerminal is used to issue AT commands and view responses over a MUX channel. A Windows serial port analyzer is used to monitor the UART1 traffic. The contents of the DLC frames are displayed as a series of hexadecimal bytes. The direction arrow points right for TE-originated frames and left for ME-originated frames.

### Starting the MUX

When "Open Mux" is selected on the MuxConfTool dialog, MuxConfTool queries the Q26 Elite for its model identifier and issues the command AT+CMUX=0,0,5,31,10,3,15. When the WCPU returns OK, the CMUX driver and Q26 Elite are in MUX protocol mode.

```
-> F9 03 FF 15 83 11 00 00 00 14 1F 00 03 01 FB F9 (Parameter Negotiation command for DLC 0)
-> F9 03 3F 01 1C F9 (Set Asynchronous Balanced Mode command for DLC 0)
```

```
<- F9 03 73 01 D7 F9 (Unnumbered Acknowledgement response for DLC 0)
```

The Parameter negotiation specified convergence layer type 1, a max frame size of 31 bytes and a window size of 1. At this point, DLC 0, the control channel, has been opened.

### Opening a DLC

When "Open DLC" for DLC 1 is selected on the MuxConfTool dialog, the following transaction takes place:

```
-> F9 03 FF 15 83 11 01 00 01 0A 1F 00 03 01 FB F9 (Parameter Negotiation command for DLC 1)
-> F9 07 3F 01 DE F9 (Set Asynchronous Balanced Mode command for DLC 1)
<- F9 03 73 01 D7 F9 (Unnumbered Acknowledgement response for DLC 1)
<- F9 01 EF 09 E3 05 07 09 9A F9 (Modem Status Command, RTR asserted)
-> F9 03 FF 09 E1 05 07 09 EE F9 (Modem Status Response)
```

DLC 1 has been opened.

### Opening an application on the DLC 1 COM port

When a terminal emulator is opened on the COM port associated with DLC 1, the following transaction takes place:

```
-> F9 03 FF 09 E3 05 07 09 EE F9 (Modem Status Command, RTR asserted)
-> F9 03 FF 09 E3 05 07 0D EE F9 (Modem Status Command, RTR and RTC asserted)
<- F9 01 EF 09 E1 05 07 09 9A F9 (Modem Status Response)
<- F9 01 EF 09 E1 05  07 0D 9A F9 (Modem Status Response)
```

DLC 1 has been opened.

### Sending and receiving application data

When "AT+GMM<cr>" is entered in the terminal emulator and the Q26 Elite responds with

```
AT+GMM
 MULTIBAND 800 1900 Model 114
OK
```

the following transaction takes place:

```
-> F9 05 EF 0F 41 54 2B 47 4D 4D 0D BC F9 (Unnumbered Information with Header check on DLC 1)
<- F9 05 EF 0F 41 54 2B 47 4D 4D 0D BC F9 (Unnumbered Information with Header check on DLC 1)
<- F9 05 EF 3F 0D 0A 20 4D 55 4C 54 49 42 41 4E 44 20 38 30 20 31 39 30 30 20 4D 6F 64 65 6C
20 31 31 98 F9 (Unnumbered Information with Header check on DLC 1)
<- F9 05 EF 11 0D 0A 0D 0A 4F 4B 0D 0A 4A F9 (Unnumbered Information with Header check on DLC
1)
```

### Entering low power state

The Q26 Elite can be placed in a low power state during periods of inactivity. However, there are a number of conditions, all of which must be true before the Q26 Elite will enter low power state.

1. The Q26 Elite must not be searching for a network.
2. A GPS fix must not be in progress.
3. QXDM must not be communicating with the Q26 Elite. If QXDM is using the USB connection, it must be exited before disconnecting the USB cable.
4. The USB cable must not connected to the Q26 Elite.
5. The Q26 must be in the idle call state.
6. AT+W32K=1 must be issued to the Q26 Elite to enable low power state.

7. The DTE signal DTR must be negated for each DLC.

When all power saving state conditions are met, the Q26 will issue a Power Saving Control command to the TE. When the TE responds, the Q26 Elite will enter low power state.

```
-> F9 05 EF 15 61 74 2B 77 33 32 6B 3D 31 0D 4D F9 (AT command "AT+W32K=1" on DLC1)
<- F9 05 EF 15 61 74 2B 77 33 32 6B 3D 31 0D 4D F9
<- F9 05 EF 0D 0D 0A 4F 4B 0D 0A 5F F9 (DCE echoes command and reponds with "OK")
-> F9 03 FF 09 E3 05 07 09 EE F9 (Modem Status command for DLC 1 with DTR negated)
<- F9 01 EF 09 E1 05 07 09 9A F9 (Modem Status command for DLC 1 with DSR negated)
-> F9 03 FF 09 E1 05 07 87 EE F9 (Modem Status response)
<- F9 01 EF 05 43 01 93 F9 (Power Saving Control command)
-> F9 01 FF 05 41 01 86 F9 (Power Saving Control response)
```

### Exiting low power state

The Q26 Elite will exit the low power state when activity is detected on UART1. Since the TE knows that the Q26 Elite is in low power state, it sends flag characters according to 27.010 until flag characters are received from the Q26 Elite. When flag characters are received, the TE stops sending flag characters and transitions to normal operation and sends its commands.

In this example, when the TE DTR signal is asserted, the Q26 Elite exits low power state.

```
-> F9 F9 F9 F9 F9 F9 F9 F9… (flag characters)
<- F9 F9 F9 F9 F9 F9 (flag characters)
-> F9 03 FF 09 E3 05 07 0D EE F9 (Modem Status command for DLC1 with DTR,RTS asserted)
<- F9 01 EF 09 E1 05 07 0D 9A F9 (Modem Status response)
<- F9 01 EF 09 E3 05 07 8D 9A F9 (Modem Status command for DLC1 with DSR,CTS asserted)
-> F9 03 FF 09 E1 05 07 8D EE F9 (Modem Status response)
```

### Closing a DLC

When "Close DLC" for DLC 1 is selected on the MuxConfTool dialog, the following transaction takes place:

```
-> F9 07 53 01 3F F9 (Disconnect command for DLC 1)
```

DLC 1 has been closed.

### Closing the MUX

When "Close Mux" is selected on the MuxConfTool dialog, the following transaction takes place:

```
-> F9 03 53 01 FD F9 (Disconnect command for DLC 0)
<- F9 03 73 01 D7 F9 ((Unnumbered Acknowledgement response for DLC 0)
```

The MUX has been closed. AT commands may again be issued on the Q26 Elite UART1.

## 9 Software Compatability Matrix

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

| Core | Open AT® | IP | Compatibility |
|---|---|---|---|
| R1A | N/A | N/A | YES |

## 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

| Level | Date | History |
|---|---|---|
| 001 | August 21, 2009 | Creation |

# 12 Legal Notice

**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.

**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.

**Copyright**



**Trademarks**

AirCard® and "Heart of the Wireless Machine®" are filed or registered trademarks of Sierra Wireless. Watcher® is a trademark of Sierra Wireless, registered in the European Community. Sierra Wireless, the Sierra Wireless logo, the red wave design, and the red-tipped antenna are trademarks of Sierra Wireless.

wavecom®, inSIM®, "YOU MAKE IT, WE MAKE IT WIRELESS®", WAVECOM®, WISMO®, Wireless Microprocessor®, Wireless CPU®, Open AT® are filed or registered trademarks of Wavecom S.A. in France and/or in other countries.

Windows® is a registered trademark of Microsoft Corporation.

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

Other trademarks are the property of the respective owners.