Skywire™ Development Kit

Using the USB Qualcomm MSM Interface (QMI) Protocol in Linux

NimbeLink Corp.

Revised: June 2015

Table of Contents

1. Introduction

1.1. Applies to these Part Numbers

Orderable Device	Description	Carrier
NL-SWDK	Skywire™ Development Kit	Any
NL-AB-BBBC	Skywire™ BeagleBone Black Cape	Any
NL-SW-LTE-TSVG	LTE without Fallback, GPS, GLONASS	Verizon
NL-SW-LTE-TNAG	LET with HSPA+ Fallback, GPS, GLONASS	Any GSM (AT&T, T-Mobile, etc.)
NL-SW-LTE-TEUG	LTE with HSPA+ Fallback, GPS, GLONASS, EU	Any EU GSM

1.2. Overview

This document describes using the Qualcomm MSM (Mobile Station Modems) Interface (QMI) protocol to interface a Skywire™ 4G modem over USB to the Linux operating system and access the 4G network as an IP interface.

With the protocol in place, the interface to the modem looks more like an Ethernet device than a serial communication device. It is therefore more efficient in terms of moving data in and out of the Linux network stack.

Examples provided have been tested with:

NL-SW-LTE-TSVG firmware version 17.00.573

NL-SW-LTE-TNAG firmware version 1x.xx.x.x

Hardware setup is the Skywire™ modem plugged into the Skywire Development Kit and USB cable connecting Mini-USB connector J5 into:

BeagleBone Black USB host port or

Raspberry Pi USB host port

Alternative configuration is the Skywire modem plugged into BeagleBone Black Cape (NL-AB-BBBC) and Cape plugged into BeageleBone Black with USB cable connecting the Cape to the BeagleBone USB Host port.

These examples were prepared using Arch Linux (release as of 04/01/2015) on the Raspberry Pi and the Beaglebone.

1.3 Prerequisites

This document assumes you have completed the initial setup of your modem and development kit. If you have not completed those steps, refer to the Skywire™ Development Kit User Manual and complete the modem setup before proceeding. . The modem must already be provisioned or PDP context set before continuing. See the Development Kit User Manual for details on how to accomplish these steps.

Your Linux platform will need kernel version 3.10 or later. There are issues in earlier kernels that prevent the required driver from working correctly and the USB IDs of the Skywire™ 4G modems are not present.

Your kernel will need the "qmi_wwan" driver installed. This driver can be either built-in or a loadable module.

If you are using a Raspberry Pi, Beaglebone, or other embedded platform without keyboard and video console, you will need to be connected to that platform via the serial console since later we will take down the Ethernet interface that may be serving as your console connection.

2. QMI Setup and Use

2.1. Check for Updates

Make sure your Arch Linux system is fully updated by typing the following command:

```
# pacman -Syu
```

followed by the enter key, and downloading the latest updates.

2.2. Install the libqmi Package

The libqmi package is required. To install it, type:

```
# pacman -S libqmi
```

followed by the enter key.

2.3. Configure APN

LTE modems must use specific Access Point Names (APN) provided by the cellular carrier.

The “qmi-network” tool requires that the APN be configured in the file:

```
/etc/qmi-network.conf
```

To create this file, we will use the `nano` text editor; however you can use whatever text editor you prefer. Type the following command:

```
# nano /etc/qmi-network.conf
```

followed by the enter key, and add the following line:

```
APN=yourAPN
```

Where “yourAPN” is the APN your cellular provider has told you should be used for this modem and the SIM it is operating with. Examples would be:

```
APN=VZWINTERNET
APN=broadband
APN=mw01.VZWSTATIC
```


Once you have entered your APN according to the above format, save the file. In `nano` press **CTRL-X.** It will ask if you want to save your changes: press **y**. `nano` will then ask what you would like to name the file. Verify that it is named **/etc/qmi-network.conf** and press the Enter key. You will then be returned to the command line.

2.4. Connect Skywire™ Development Kit

Plug in your USB cable to port J5 on your Skywire™ Development Kit. Then, plug the cable into the computer.

2.5. Power on the Modem

Power on your Skywire™ modem according to the Skywire™ Development Kit User Manual, and after approximately five seconds, you should see syslog entries appear as the system enumerates the new USB device:

```
[61583.248406] sub 1-1.5: new high-speed USB device number 5 using dwc_otg
[61583.385322] usb 1-1.5: New USB device found, idVendor=1bc7,
idProduct=1201
[61583.401475] usb 1-1.5: New USB device strings: Mfr=1, Product=2,
SerialNumber=3
[61583.417954] usb 1-1.5: Product: Android
[61583.430903] usb 1-1.5: Manufacturer: Android
[61583.444012] usb 1-1.5: SerialNumber: 0123456789ABCDEF
[61583.487521] option 1-1.5:1.0: GSM modem (1-port) converter detected
[61583.514644] usb 1-1.5: GSM modem (1-port) converter now attached to
ttyUSB0
[61583.551854] qmi_wwan 1-1.5:1.2: cdc-wdm0: USB WDM device
[61583.592282] qmi_wwan 1-1.5:1.2 wwan0: register 'qmi_wwan' at
usb-bcm2708_usb-1.5, WWAN/QMI device, 2a:d5:e4:1c:f1:82
[61583.621759] option 1-1.5:1.3: GSM modem (1-port) converter detected
[61583.637369] usb 1-1.5: GSM modem (1-port) converter now attached to
ttyUSB1
[61583.654202] option 1-1.5:1.4: GSM modem (1-port) converter detected
[61583.669672] usb 1-1.5: GSM modem (1-port) converter now attached to
ttyUSB2
[61583.686265] option 1-1.5:1.5: GSM modem (1-port) converter detected
[61583.708173] usb 1-1.5: GSM modem (1-port) converter now attached to
ttyUSB3
[61583.739549] option 1-1.5:1.6: GSM modem (1-port) converter detected
[61583.771578] usb 1-1.5: GSM modem (1-port) converter now attached to
ttyUSB4
```

2.6. Verify USB and wwan0 Connections

Type the following command to verify that the USB modem is present:

```
# lsusb
```

Typical response:

Bus 001 Device 005: ID 1bc7:1201 Telit Wireless Solutions

You should also see the `wwan0` device appear in the network stack as shown above or if queried,

```
# ifconfig wwan0
```

Typical Response:

```
wwan0:    flags=4098<BROADCAST,MULTICAST>  mtu 1500 ether
          2a:d5:e4:1c:f1:82  txqueuelen 1000  (Ethernet)
          RX packets 0  bytes 0 (0.0 B) RX errors 0  dropped 0  overruns
                0  frame 0
                TX packets 0  bytes 0 (0.0 B)
                TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

2.7. Take Down the Ethernet Interface

A QMI connection requires that any existing Ethernet connection be taken down. This is required so that the DHCP client serving the cellular link can correctly configure the system gateway and DNS resolver.

To bring down the Ethernet connection, type the following command:

```
# ifconfig eth0 down
```

2.8. Bring up the QMI Connection

To bring up the QMI connection, type the following commands:

```
# ifconfig wwan0 up
# qmicli -d /dev/cdc-wdm0
--dms-set-operating-mode=online
```

Typical response:

[/dev/cdc-wdm0] Operating mode set successfully

```
# qmi-network /dev/cdc-wdm0 start
```

Typical response:

Loading profile...
  APN: broadband
Starting network with 'qmicli -d /dev/cdc-wdm0
--wds-start-network=broadband --client-no-release-cid'...
Saving state... (CID: 10)
Saving state... (PDH: 1205860592)
Network started successfully

2.9. Start DHCP Client

To start the DHCP client, type the following command:

```
# dhcpcd wwan0
```

Typical response:

DUID 00:01:00:01:1d:00:d0:a3:3e:2a:2d:5f:50:26
wwan0: IAID e4:1c:f1:82
wwan0: soliciting a DHCP lease
wwan0: offered 10.124.149.216 from 10.124.149.217
wwan0: leased 10.124.149.216 for 7200 seconds
wwan0: adding route to 10.124.149.216/30
wwan0: adding default route via 10.124.149.217
forked to background, child pid 2465

2.10. Test the QMI Connection

To test the QMI connection, type the following command:

```
# ping -c 3 www.google.com
```

Typical response:

PING www.google.com (74.125.198.99) 56(84) bytes of data.
64 bytes from og-in-f99.1e100.net (74.125.198.99): icmp_seq=1 ttl=41 time=110 ms
64 bytes from og-in-f99.1e100.net (74.125.198.99): icmp_seq=2 ttl=41 time=89.9 ms
64 bytes from og-in-f99.1e100.net (74.125.198.99): icmp_seq=3 ttl=41 time=88.6 ms
--- www.google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 88.625/96.521/110.976/10.235 ms

2.11. Bring Down the QMI Connection

To bring down the QMI connection, type the following command:

```
# qmi-network /dev/cdc-wdm0 stop
```

To stop the DHCP client, type:

```
# kill `cat /var/run/dhcpcd.pid`
```


Finally, to bring down the `wwan0` interface, type:

```
# ifconfig wwan0 down
```

