



QUICK START GUIDE

QSM-8Q60

Linux EVK v3.0.2

**Copyright**

Copyright © 2017-2018 VIA Technologies Incorporated. All rights reserved.

No part of this document may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual or otherwise without the prior written permission of VIA Technologies, Incorporated.

Trademarks

All brands, product names, company names, trademarks and service marks are the property of their respective holders.

Disclaimer

VIA Technologies makes no warranties, implied or otherwise, in regard to this document and to the products described in this document. The information provided in this document is believed to be accurate and reliable as of the publication date of this document. However, VIA Technologies assumes no responsibility for the use or misuse of the information (including use or connection of extra device/ equipment/add-on card) in this document and for any patent infringements that may arise from the use of this document. The information and product specifications within this document are subject to change at any time, without notice and without obligation to notify any person of such change.

VIA Technologies, Inc. reserves the right to make changes to the products described in this manual at any time without prior notice.



Revision History

Version	Date	Remarks
1.00	03/05/2018	Initial release



Table of Contents

1. Introduction	1
1.1 EVK Package Contents	1
1.1.1 Firmware Folder Contents	1
1.1.2 Document Folder Contents	1
1.1.3 Tools Folder Contents	1
1.2 Version Information and Supported Features	2
2. Image Installation	3
2.1 Booting from a Micro SD Card	3
2.2 Booting from the SPI ROM with eMMC	4
3. Hardware Function	6
3.1 Setting Up U-Boot Parameters	6
3.2 Restoring Default U-Boot Parameters	8
3.3 Using the OpenEmbedded Console	8
3.4 Setting Up Display Device	9
3.4.1 Setting Up the U-Boot Display Parameters for HDMI	9
3.4.2 Setting Up the U-Boot Display Parameters for LVDS	9
3.5 Setting Up COM 4, COM 5, COM 6 and COM 7	11
3.6 Configuring FlexCAN	13
3.7 Configuring Watchdog Timer	15
3.8 Setting Up GPIO	16
3.8.1 Setting Up GPI Mode	17
3.8.2 Setting Up GPO Mode	17
3.9 Configuring RTC	18
4. Accessories	19
4.1 Configuring the VNT9271 USB Wi-Fi Dongle	19
4.2 Configuring the EMIO-1541 miniPCle Wi-Fi Module	20
4.3 Configuring the EMIO-2531 miniPCle Wi-Fi & Bluetooth Module	21
4.3.1 Connecting to the Internet	21
4.3.2 Enabling Bluetooth	22
4.3.2.1 Setting Up Bluetooth A2DP Profile	22
4.3.2.2 Setting Up Bluetooth SPP Profile	23
4.4 Configuring the EMIO-2550 miniPCle Mobile Broadband Module	25
4.4.1 Connecting to the Internet	25
4.4.2 Enabling GPS	26

1. Introduction

This Quick Start Guide provides an overview on how to boot the Linux EVK system image on the QSM-8Q60 module and configure the supported hardware functions and I/O defined on the QSMDB2 carrier board.

The QSM-8Q60 Linux BSP v3.0.2 is developed based on the NXP fsl-yocto-L4.1.15_1.1.0-ga (Yocto 2.0 Jethro) and it enables hardware features that are defined on the QSM-8Q60 and QSMDB2 carrier board.

1.1 EVK Package Contents

There are three folders in the package listed as below.

Firmware folder	Description
QSM-8Q60_Yocto2.0_BIN_v3.0.2.tar.gz	Yocto EVK system image and installation script files
Document folder	Description
QSM-8Q60_Linux_EVK_v3.0.2_Quick_Start_Guide_v1.00_20180305.pdf	Quick Start Guide
Tools folder	Description
BT_Config.zip	Contain Bluetooth configuring files

QSM-8Q60 Linux EVK contents

1.1.1 Firmware Folder Contents

QSM-8Q60_Yocto2.0_BIN_v3.0.2.tar.gz: contains installation script files and the precompiled U-boot and image for evaluating the QSM-8Q60 and QSMDB2 carrier board with an OpenEmbedded filesystem.

1.1.2 Document Folder Contents

QSM-8Q60_Linux_EVK_v3.0.2_Quick_Start_Guide_v1.00_20180305.pdf: This Quick Start Guide provides an overview on how to boot the Linux EVK system image on the QSM-8Q60 module and configure the supported hardware functions and I/O defined on the QSMDB2 carrier board.

1.1.3 Tools Folder Contents

BT_Config.zip: contains the three Bluetooth configuration files for A2DP support.

1.2 Version Information and Supported Features

- U-Boot version: 2015.04
- Kernel version: 4.1.15
- Evaluation image: OpenEmbedded-core built with Yocto 2.0 Jethro
- Development based on NXP fsl-yocto-L4.1.15_1.1.0-ga (Yocto 2.0 Jethro)
- Supports SPI with eMMC or Micro SD boot (default)
- Supports HDMI and LVDS display
- Supports HDMI audio output
- Supports AUO LVDS capacitive touch panels (through USB interface)
 - AUO 10.4" G104XVN01.0 LVDS panel (1024×768)
 - AUO 7" G070VW01 V0 LVDS panel (800×480)
- Supports COM 1, COM 3 (TX/RX) and COM 2 as a debug port
- Supports COM 4, COM 5, COM 6 and COM 7 (RS-232/RS-485/RS-422)
- Supports two FlexCAN TX/RX
- Supports Gigabit Ethernet (LAN1)
- Supports 10/100Mbps Ethernet (LAN2)
- Supports Line-in, Line-out, and Mic-in
- Supports VNT9271 USB Wi-Fi dongle
- Supports EMIO-1541 miniPCle Wi-Fi module
- Supports EMIO-2531 miniPCle Wi-Fi & Bluetooth module
 - Supports Bluetooth A2DP and SPP profile
- Supports EMIO-2550 miniPCle Mobile Broadband module
- Supports Watchdog Timer, GPIO and RTC

2. Image Installation

This section explains the setup requirements for booting from a Micro SD card or the SPI ROM and eMMC.

The installation script files, the precompiled U-boot and image are provided in the “Firmware” folder.

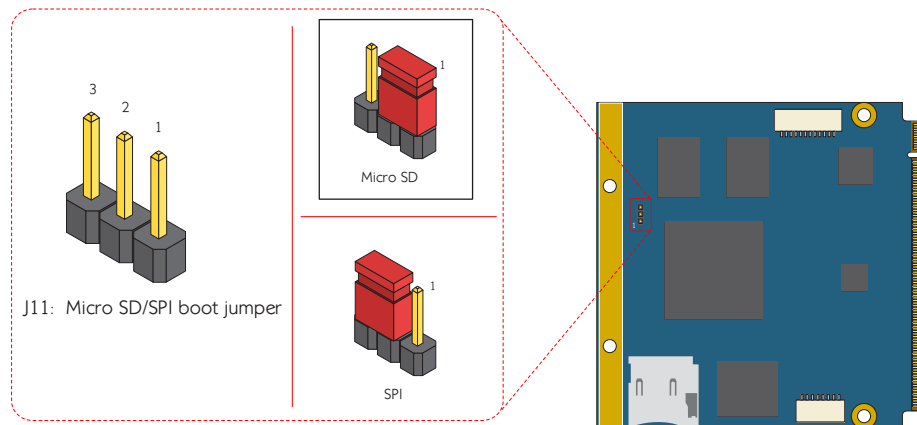
2.1 Booting from a Micro SD Card

The first step is to install this image onto a Micro SD card. Insert a Micro SD card into your Linux host machine and make sure it is not mounted. Install the SD card image onto the Micro SD card with the commands below, replacing <device name> with the correct value for the card.

***Important:** Make sure you are writing to the correct device or the host system environment could be damaged.

```
$ tar xvf QSM8Q60_Yocto2.0_BIN_v3.0.2.tar.gz
$ cd sd_installer/
$ sudo ./mk_sd_installer.sh /dev/<device name> --yocto
```

Next, on the QSM-8Q60, set the Micro SD/SPI boot jumper (J11) to the Micro SD position as shown below.



Micro SD/SPI boot jumper diagram

Insert the prepared Micro SD card into the QSM-8Q60, connect the HDMI display to the QSMDB2 carrier board, and power on the device to initiate the boot process.

When the boot process is completed, you will see the OpenEmbedded desktop.

2.2 Booting from the SPI ROM with eMMC

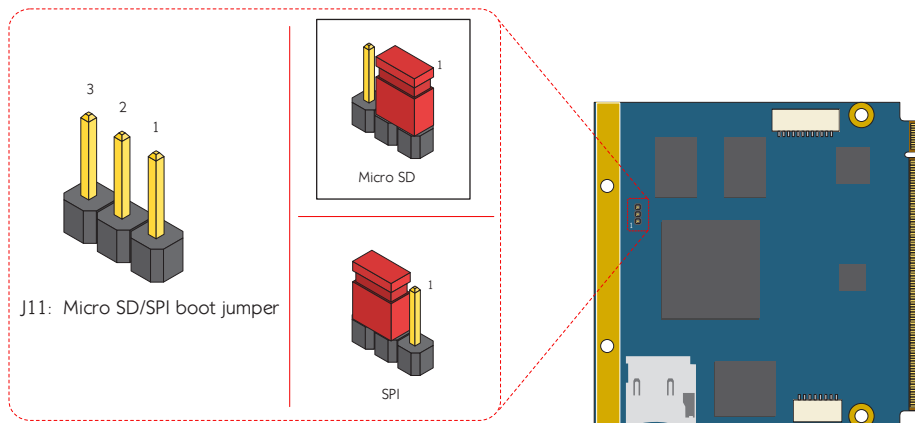
The QSM-8Q60 and QSMDB2 carrier board supports booting from the SPI ROM while loading the kernel and root filesystem from the eMMC.

The first step is to prepare a Micro SD card as the system installer. Insert a Micro SD card into your Linux host machine and make sure it is not mounted. Install the system installer onto the Micro SD card with the command below, replacing <device name> with the correct value for the card.

***Important:** Make sure you are writing to the correct device or the host system environment could be damaged.

```
$ tar xvf QSM8Q60_Yocto2.0_BIN_v3.0.0.tar.gz
$ cd sd_installer/
$ sudo ./mk_sd_installer.sh /dev/<device name>
```

Next, on the QSM-8Q60, set the Micro SD/SPI boot jumper (J11) to the Micro SD position as shown below.

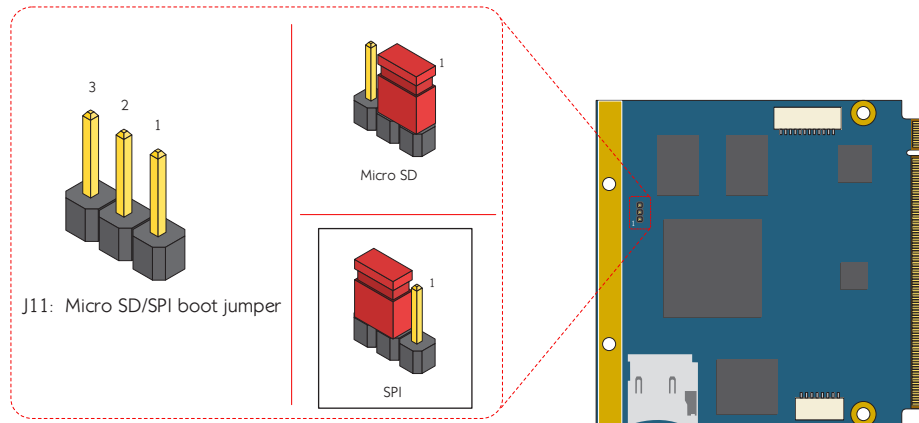


Micro SD/SPI boot jumper diagram

Insert the prepared Micro SD card into the QSM-8Q60, connect the HDMI display to the QSMDB2 carrier board, and power on the device. The system will automatically start to install U-boot and image onto the SPI ROM and eMMC.

When the install process is completed, unplug the power cable and remove the Micro SD card.

In order to boot from the SPI ROM make sure the Micro SD/SPI boot jumper (J11) is set to SPI ROM.



Micro SD/SPI boot jumper diagram

Next, connect an HDMI display, and power on the device to initiate the boot process. When the boot process is completed, you will see the OpenEmbedded desktop.

3. Hardware Function

This section explains how to enable and test the hardware functions precompiled in the Linux EVK including setting u-boot parameters, restoring default u-boot parameters, setting up display device, setting up COM 4, COM 5, COM 6 and COM 7, configuring flexCAN, configuring watchdog timer, setting up GPIO, and configuring RTC.

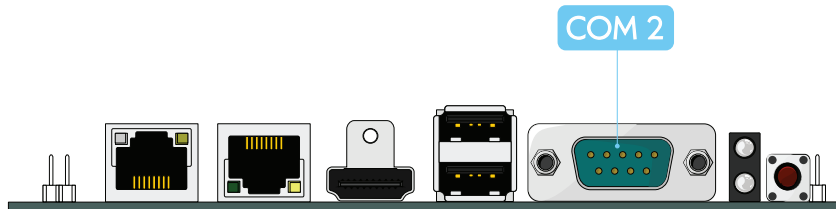
3.1 Setting Up U-Boot Parameters

The first step is to connect the host machine and the QSM-8Q60 through the COM 2 port of the QSMD B2 carrier board. Use a serial port communication program such as PuTTY, GtkTerm, or Minicom, to configure the serial port setting and connect to the debug console. There you will be able to see the U-Boot boot log and adjust settings in the U-Boot console.

```

+-----+
| A -   Serial Device       : /dev/rfcomm0 |
| B -   Lockfile Location   : /var/lock   |
| C -   Callin Program      :             |
| D -   Callout Program     :             |
| E -   Bps/Par/Bits        : 115200 8N1  |
| F -   Hardware Flow Control : No        |
| G -   Software Flow Control : No        |
+-----+
  
```

Serial port setting of host machine



COM 2 port diagram

Next, power on the QSM-8Q60 and QSMD B2 carrier board to initiate the boot process.

When prompted, press any key to stop the boot process and enter the U-Boot console as illustrated by the screenshot below.

```

U-Boot 2015.04-imx_v2015.04_4.1.15_1.0.0_ga+gd7d7c43 (Feb
15 2017 - 17:10:38)

CPU:   Freescale i.MX6DL rev1.3 at 996 MHz
CPU:   Temperature 34 C
Reset  cause: POR
Board: MX6DL-QSM-8Q60
I2C:   ready
DRAM:  2 GiB
MMC:   FSL_SDHC: 0, FSL_SDHC: 1
SF:    Detected W25Q32BV with page size 256 Bytes, erase size
4 KiB, total 4 MiB
Unable to allocate framebuffer memory
In:    serial
Out:   serial
Err:   serial
Net:   FEC [PRIME]
Normal Boot
Hit any key to stop autoboot:  0
=>
  
```

Debug console view of boot process

To list the current U-Boot parameters, use the following command:

```
=> printenv
```

Make sure the printout message includes “bootcmd=run bootcmd_auto”.

The default “bootcmd” parameter is set to auto detect the Micro SD card by attempting to load the kernel. If it is not detected, it continues to boot from the eMMC.

To load the kernel from the Micro SD card or eMMC, refer to the two examples below.

To load the kernel from the Micro SD card, use the following command:

```
=> setenv bootcmd 'run bootcmd_sd'
=> saveenv
=> boot
```

To load the kernel from the eMMC, use the following command:

```
=> setenv bootcmd 'run bootcmd_mmc'
=> saveenv
=> boot
```

3.2 Restoring Default U-Boot Parameters

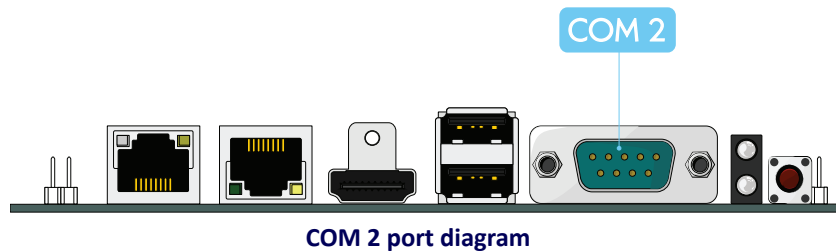
If the U-Boot parameters have been modified, the “destroyenv” command in the U-Boot console can restore the default settings.

To restart the device, use the “reset” command.

```
=> destroyenv
=> reset
```

3.3 Using the OpenEmbedded Console

The first step is to connect the host machine and the QSM-8Q60 through the COM 2 port of the QSMDB2 carrier board. Use a serial port communication program such as PuTTY, GtKTerm, or Minicom, to connect to the debug console. There you will be able to see the U-Boot boot log and adjust settings in the U-Boot console. Next, power on the QSM-8Q60 and QSMDB2 carrier board to initiate the boot process. When the boot process is completed you will be prompted to login.



The default account is “root”, with no password set (just press Enter when prompted for password).

```
...
Freescale i.MX Release Distro 4.1.15-1.1.1 imx6qsm8q60 dev/ttymxcl
imx6qsm8q60 login: root
```

3.4 Setting Up Display Device

The QSM-8Q60 Linux BSP v3.0.2 supports the following display devices:

- HDMI monitor
- AUO 10.4" G104XVN01.0 LVDS panel (1024x768)
- AUO 7" G070VW01 V0 LVDS panel (800x480)

3.4.1 Setting Up the U-Boot Display Parameters for HDMI

To set the HDMI monitor as the display output, use the following command:

```
=> run o_hdmi
=> saveenv
=> reset
```

In order to confirm the settings are correct use the following command to list the new U-Boot parameters:

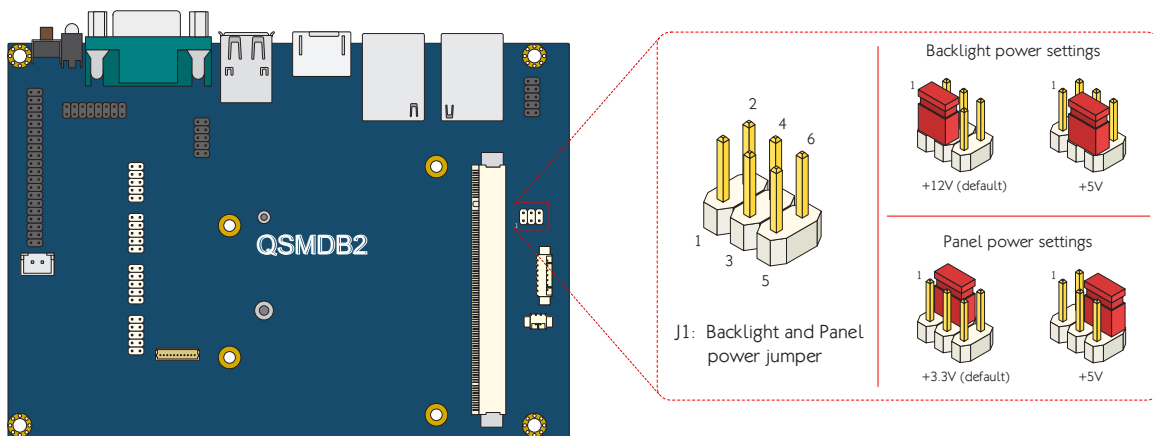
```
=> printenv hdmi

hdmi=video=mxcfb0:dev=hdmi,1920x1080M@60, bpp=32
```

Make sure that the display device is HDMI and the resolution matches your monitor's resolution.

3.4.2 Setting Up the U-Boot Display Parameters for LVDS

Before setting an LVDS panel as the display output make sure the LVDS backlight power jumper (J1) is set to +12V and the LVDS panel power jumper is set to +3.3V as shown below.



LVDS power jumper diagram

To set the AUO 10.4" G104XVN01.0 LVDS panel as the display output, use the following command:

```
=> run o_lvdsg104  
=> saveenv
```

In order to confirm the settings are correct use the following command to list the new U-Boot parameters:

```
=> printenv lvds_auo_g104  
lvds_auo_g104=video=mxcfb0:dev=ldb,LDB-XGA, if=RGB24 ldb=sin0
```

To set the AUO 7" G070VW01 V0 LVDS panel as the display output, use the following command:

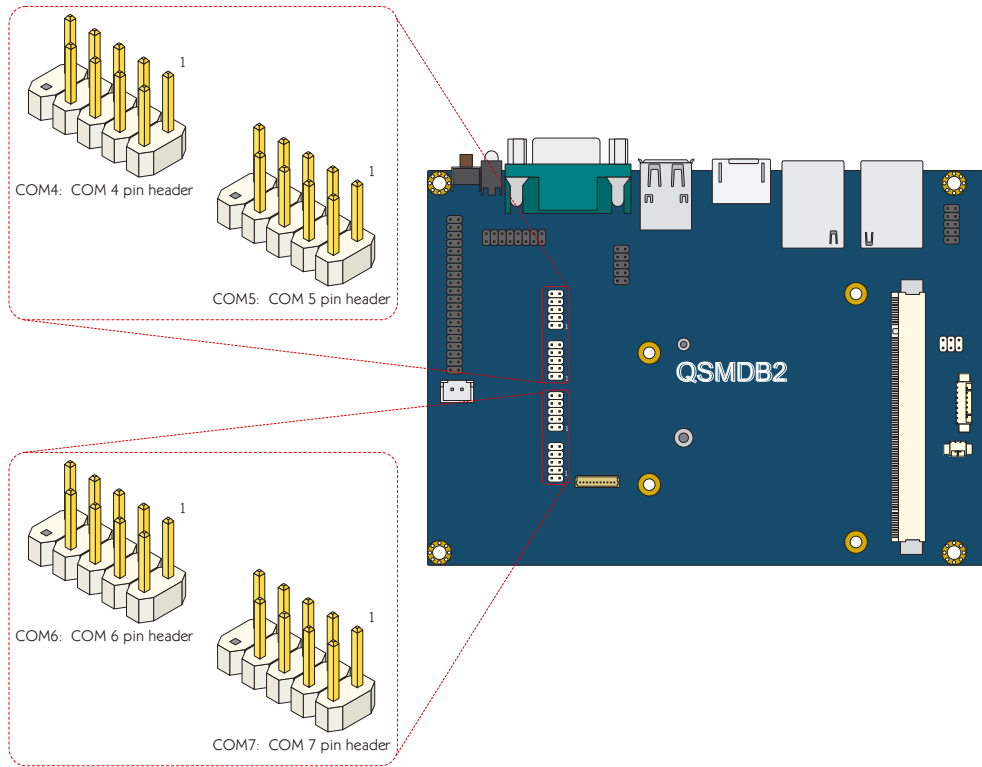
```
=> run o_lvdsg070  
=> saveenv
```

In order to confirm the settings are correct use the following command to list the new U-Boot parameters:

```
=> printenv lvds_auo_g070  
lvds_auo_g070=video=mxcfb0:dev=ldb,480C60,if=RGB24 ldb=sep0
```

3.5 Setting Up COM 4, COM 5, COM 6 and COM 7

The COM 4, COM 5, COM 6 and COM 7 pin headers can support RS-232/RS-422/RS-485 modes. For RS-232 mode the COM ports can use all pins of the pin header. For RS-422 and RS-485 modes, they can use only pins no. 2 and 3 (TX and RX signals).



COM 4, COM 5, COM 6 and COM 7 pin header diagram

To check that the QSMDB2 carrier board has correctly detected the Exar USB UART module, use the following command:

```
$ lsusb
```

Make sure the printout message includes “xr_usb_serial_common 19814 0” and “Bus 002 Device 003: ID 04e2:1414 Exar Corp.”.

To list all Exar USB serial devices, use the following command:

```
$ ls /dev/ttyXRUSB*
/dev/ttyXRUSB0 /dev/ttyXRUSB1 /dev/ttyXRUSB2 /dev/ttyXRUSB3
```

COM port	Device name
COM 4	ttyXRUSB0
COM 5	ttyXRUSB1
COM 6	ttyXRUSB2
COM 7	ttyXRUSB3

COM port and Device name definition table

Important: Please set up the serial devices before starting the communication program.

Connect the COM cable (P/N: **99G33-080127**) to the COM 4/COM 5/COM 6/COM 7 pin header on the QSMDDB2 carrier board. Then connect the host machine to the QSMDDB2 through COM cable.

To set up the serial device to RS-232 mode, use the following command:

```
$ exar_setmode <Device name> 0
```

To set up the serial device to RS-422 mode, use the following command:

```
$ exar_setmode <Device name> 1
```

To set up the serial device to RS-485 mode, use the following command:

```
$ exar_setmode <Device name> 2
```

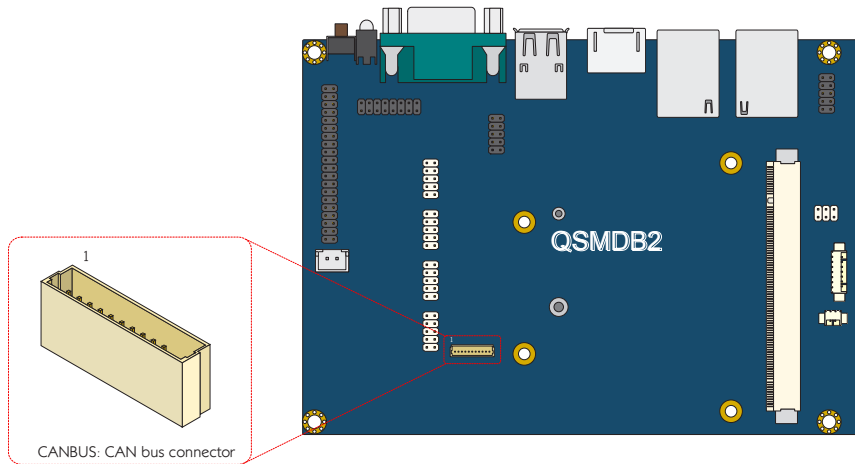
To start the Minicom serial communication program on the QSM-8Q60 and QSMDDB2 carrier board, use the following command:

```
$ minicom -s
```

Start a serial communication program on the host machine with the same communication settings using the appropriate serial device. Afterwards the host machine and the QSM-8Q60 and QSMDDB2 carrier board will be able to communicate through the programs (for example sending a keypress on one machine will be shown on the other machine).

3.6 Configuring FlexCAN

The CAN bus connector of the QSM-8Q60 and QSMDB2 carrier board supports two CAN bus ports.



CAN bus connector diagram

Pin	Signal
1	NC
2	NC
3	NC
4	CANH1
5	NC
6	GND
7	CANL2
8	CANH2
9	NC
10	CANL1

CAN bus connector pinouts table

Please note that you need to first disable the CAN bus then set the CAN bus bitrate. The CAN bus bitrate should be between $5000 < [BRT_value] < 1000000$.

The CAN bus 1 is can0 and the CAN bus 2 is can1.

To set the CAN bus 1 bitrate [BRT_value], use the following command:

```
# ip link set can0 type can bitrate [BRT_value]
```

To enable the CAN bus 1, use the following command:

```
# ifconfig can0 up
```

To disable the CAN bus 1, use the following command:

```
# ifconfig can0 down
```

Use the cansend command to send CAN bus 1 data:

```
# cansend can0 <ID>#<Data>
```

<ID>: Device ID of CAN bus

<Data>: Send out data

Use the candump command to receive data from CAN bus 1:

```
# candump can0 &
```

3.7 Configuring Watchdog Timer

Watchdog Timer is an electronic timer that is used to detect and recover from system malfunctions.

The `wdt_driver_test.out` provides an application to configure the Watchdog timer.

To enable Watchdog Timer with a default timeout of 60 seconds, use the following command:

```
$ /unit_tests/wdt_driver_test.out enable
```

To change the timeout `<value>` and the board reboot `<value>` (seconds), use the following command:

```
$ /unit_tests/wdt_driver_test.out set_timeout <value>
```

To send keep-alive requests to the Watchdog `<value>` (seconds), use the following command:

```
$ /unit_tests/wdt_driver_test.out set_keep_alive <value>
```

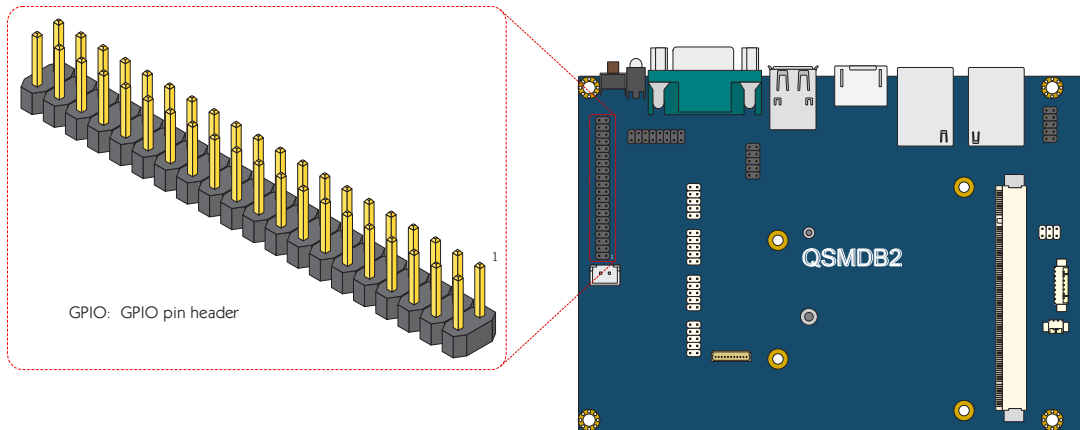
When the value of “`set_keep_alive`” is smaller than the value of “`set_timeout`”, the board will not reboot.

To disable the Watchdog timer, use the following command:

```
$ /unit_tests/wdt_driver_test.out disable
```

3.8 Setting Up GPIO

The GPIO pin header on the QSMDB2 carrier board consists of 40 pins. The following section explains setting up these pins for input/output communication.



GPIO pin header diagram

The echo value of GPIO pins are listed below:

Pin	Signal	GPIO echo value	Pin	Signal	GPIO echo value
1	GPIO0	208	2	GPIO4	212
3	GPIO1	209	4	GPIO5	213
5	GPIO2	210	6	GPIO6	214
7	GPIO3	211	8	GPIO7	215
9	GND		10	GND	
11	GPIO10	216	12	GPIO14	220
13	GPIO11	217	14	GPIO15	221
15	GPIO12	218	16	GPIO16	222
17	GPIO13	219	18	GPIO17	223
19	GND		20	GND	
21	GPIO20	224	22	GPIO24	228
23	GPIO21	225	24	GPIO25	229
25	GPIO22	226	26	GPIO26	230
27	GPIO23	227	28	GPIO27	231
29	GND		30	GND	
31	GPIO30	232	32	GPIO34	236
33	GPIO31	233	34	GPIO35	237
35	GPIO32	234	36	GPIO36	238
37	GPIO33	235	38	GPIO37	239
39	GND		40	GND	

GPIO pin header pinout and GPIO echo value

Enable the multifunction pin as a GPIO pin. Here is the example for GPIO pin 1:

```
$ echo 208 > /sys/class/gpio/export
```

Disable the multifunction pin as a GPIO pin. Here is the example for GPIO pin 1:

```
$ echo 208 > /sys/class/gpio/unexport
```

3.8.1 Setting Up GPI Mode

To set a pin to “input” mode (GPI), use the following command (here is the example for GPIO pin 1):

```
$ echo in > /sys/class/gpio/gpio208/direction
```

To read the GPI input value, use the following command:

```
$ cat /sys/class/gpio/gpio208/value
```

3.8.2 Setting Up GPO Mode

To set a pin to “output” mode (GPO), use the following command (here is the example for GPIO pin 1):

```
$ echo out > /sys/class/gpio/gpio208/direction
```

To output a “Low” value on the GPO pin, use the following command:

```
$ echo 0 > /sys/class/gpio/gpio208/value
```

To output a “High” value on the GPO pin, use the following command:

```
$ echo 1 > /sys/class/gpio/gpio209/value
```

3.9 Configuring RTC

The RTC (Real-Time Clock) keeps track of the current time values. The time values include the year, month, date, hours, minutes and seconds. To configure the RTC on the QSM-8Q60, set the system time with the Linux date command shown below:

```
$ date mmddHMMYYYY
```

Time	Value
mm	2-digit month
dd	2-digit day
HH	2-digit hours in 24h system
MM	2-digit minutes
YYYY	4-digit year

Time value table

For example, the value 061110072013 means June 11 2013 10:07. The date command also accepts other time formats. For more information, use the following command:

```
$ date --help
```

To write the system time to the RTC, use the following command:

```
$ hwclock -w -f /dev/rtc0
```

To read the RTC time, use the following command:

```
$ hwclock -r
```

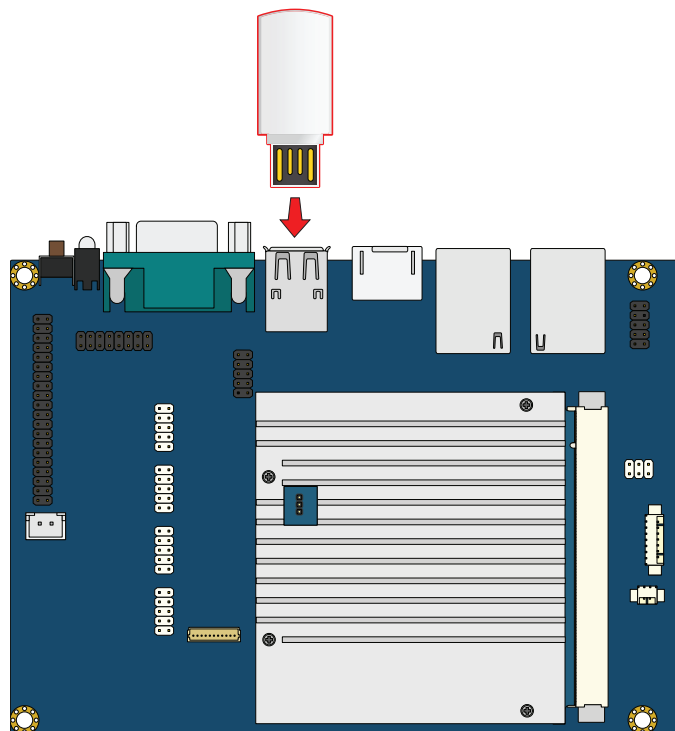
4. Accessories

This section explains how to install and configure the various EMIO wireless modules available for the QSM-8Q60 and QSMDB2 carrier board.

4.1 Configuring the VNT9271 USB Wi-Fi Dongle

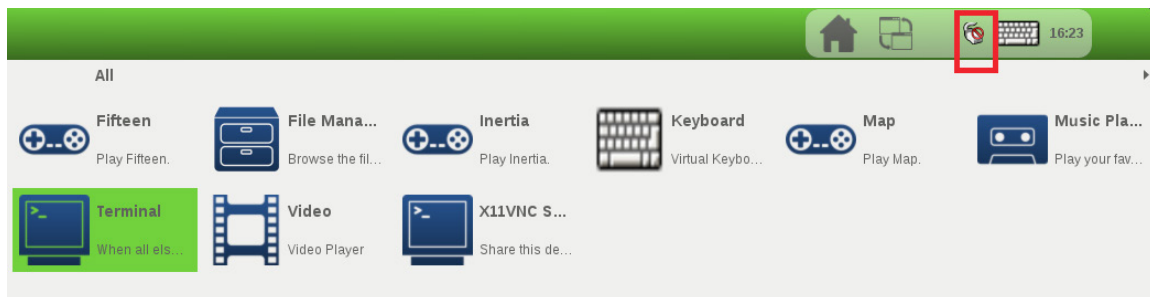
The VNT9271 USB Wi-Fi dongle supports Wi-Fi functionality through USB port connection.

The first step is to insert the VNT9271 USB Wi-Fi dongle into a USB port. Next, make sure to unplug any LAN cables or other Wi-Fi/3G modules you have installed. Finally, power on the QSM-8Q60 and QSMDB2 carrier board.



Inserting the VNT9271 USB Wi-Fi dongle

When the boot process is completed, click on the “Connection Manager” icon to configure WLAN.



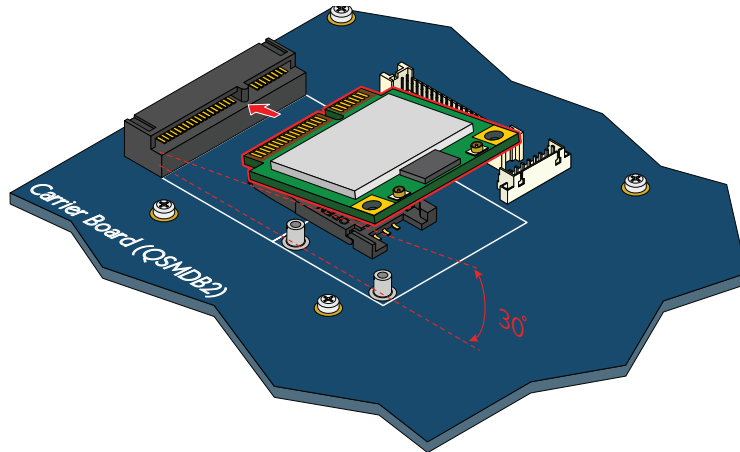
Connection Manager icon diagram

When the connection is created, connect to the internet through your web browser.

4.2 Configuring the EMIO-1541 miniPCIe Wi-Fi Module

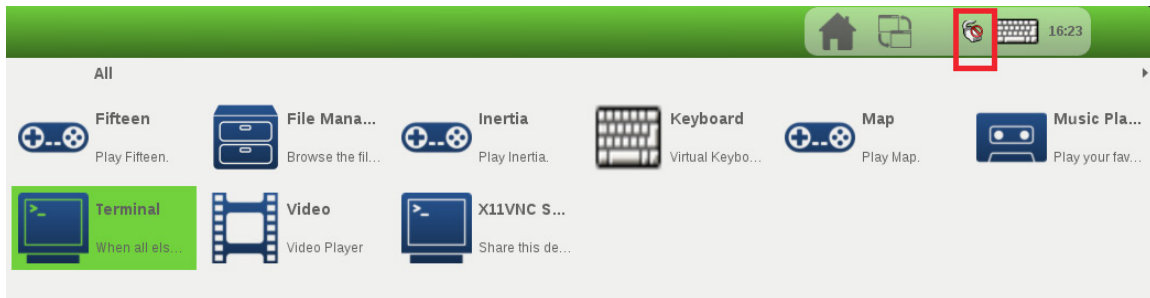
The EMIO-1541 module supports Wi-Fi through the onboard miniPCIe slot.

The first step is to insert the EMIO-1541 module into the miniPCIe slot. After installing the module connect the provided antenna to the module. Next, make sure to unplug any LAN cables or USB Wi-Fi dongles you have installed. Finally, power on the QSM-8Q60 and QSMDB2 carrier board.



Inserting the EMIO-1541 module

When the boot process is completed, click on the “Connection Manager” icon to configure WLAN.



Connection Manager icon diagram

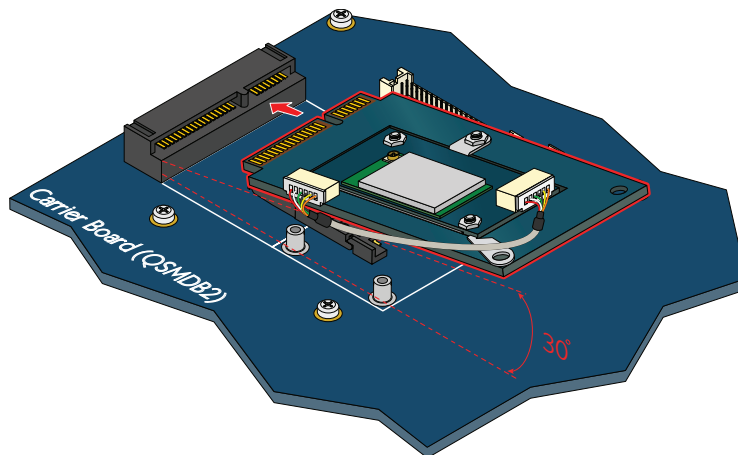
When the connection is created, connect to the internet through your web browser.

4.3 Configuring the EMIO-2531 miniPCle Wi-Fi & Bluetooth Module

The EMIO-2531 module supports Wi-Fi and Bluetooth functionality through the onboard miniPCle slot.

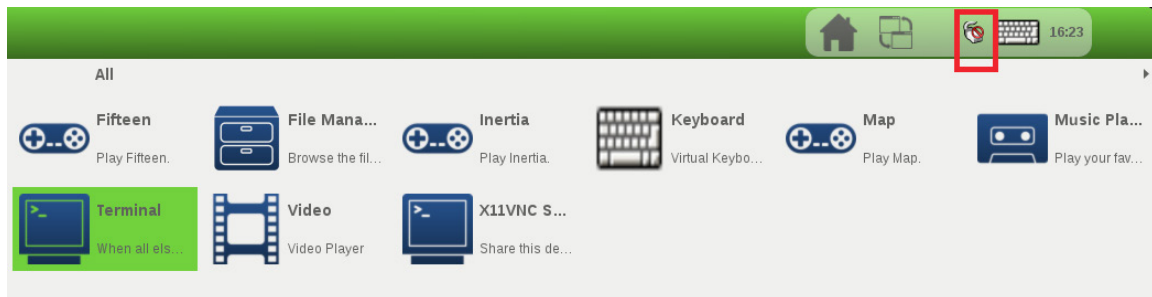
4.3.1 Connecting to the Internet

The first step is to insert the EMIO-2531 module into the miniPCle slot. After installing the EMIO-2531 module connect the provided antenna to the module. Next, make sure to unplug any LAN cables or other Wi-Fi/3G modules you have installed. Finally, power on the QSM-8Q60 and QSMDB2 carrier board.



Inserting the EMIO-2531 module

When the boot process is completed, click on the “Connection Manager” icon to configure WLAN.



Connection Manager icon diagram

When the connection is created, connect to the internet through your web browser.

4.3.2 Enabling Bluetooth

The following section explains setting up the Advanced Audio Distribution Profile (A2DP) and Serial Port Profile (SPP).

4.3.2.1 Setting Up Bluetooth A2DP Profile

The first step is to extract to **BT_Config.zip** file. Next, copy the **pulseaudio-system.conf**, **bluetooth.conf** and **system.pa** files to **/etc/dbus-1/system.d/** folder to replace the original **pulseaudio-system.conf**, **Bluetooth.conf** and **system.pa** files in order to add the A2DP support into the system.

To enable the Bluetooth service, use the following command:

```
$ rfkill unblock bluetooth
$ hciconfig hci0 up
$ /usr/lib/bluez5/bluetooth/bluetoothd -C -d &
$ hciconfig hci0 piscan
```

Use the following commands to run the “bluetoothctl” to show the EMIO-2531 Bluetooth MAC address and set up pairing mode.

```
$ bluetoothctl
[NEW] Controller 5C:F3:70:24:4B:2E (EMIO-2531 Bluetooth MAC address)
[bluetooth]#
[bluetooth]# show
Controller 5C:F3:70:24:4B:2E
    Name: BlueZ 5.25
    Alias: BlueZ 5.25
    ...
[bluetooth]# select 5C:F3:70:24:4B:2E (select controller_mac_address)
[bluetooth]# power on
Changing power on succeeded
[bluetooth]# agent on
Agent registered
[bluetooth]# default-agent
Default agent request successful
[bluetooth]# discoverable on
Changing discoverable on succeeded
[CHG] Controller 5C:F3:70:24:4B:2E Discoverable: yes
[bluetooth]# pairable on
Changing pairable on succeeded
[bluetooth]# scan on
Discovery started
[CHG] Controller 5C:F3:70:24:4B:2E Discovering: yes
[NEW] Device 00:1D:82:BC:C1:C4 Jabra BT-530
...
[bluetooth]# scan off
[bluetooth]# devices
Device 00:1D:82:BC:C1:C4 Jabra BT-530
Device ...
[bluetooth]# pair 00:1D:82:BC:C1:C4 (pair device_mac_address)
[bluetooth]# trust 00:1D:82:BC:C1:C4 (trust device_mac_address)
```

To enable Bluetooth device, use the following commands:

```
[bluetooth]# connect 00:1D:82:BC:C1:C4 (connect device_mac_address)
[bluetooth]# info 00:1D:82:BC:C1:C4 (info device_mac_address)
[bluetooth]# quit
```

The default audio output setting for the QSM-8Q60 is HDMI device.

To list all the available audio output devices, use the following command:

```
$ pactl list short sinks
0 alsa_output.platform-sound-hdmi.23.analog-stereo module-alsa-card.c s16le 2ch 48000Hz
SUSPENDED
1 alsa_output.platform-sound.22.analog-stereo module-alsa-card.c s16le 2ch 48000Hz
SUSPENDED
2 bluez_sink.00_1D_82_BC_C1_C4 module-bluez5-device.c s16le 2ch 48000Hz IDLE
```

From the output, we can see the following:

- 0 = HDMI device
- 1 = Line-out
- 2 = Bluetooth device

To set up the audio output to a Bluetooth device, use the following command:

```
$ pacmd set-default-sink 2
```

***Note:** If you would like to change audio output after enabling a Bluetooth device, use the commands below:

To set up the audio output to Line-out, use the following command:

```
$ pacmd set-default-sink 1
```

To set up the audio output to HDMI device, use the following command:

```
$ pacmd set-default-sink 0
```

4.3.2.2 Setting Up Bluetooth SPP Profile

The AMOS-820 supports SPP server mode and SPP client mode.

To enable the Bluetooth service, use the following command:

```
$ rfkill unblock bluetooth
$ hciconfig hci0 up
$ /usr/lib/bluez5/bluetooth/bluetoothd -C -d &
$ hciconfig hci0 piscan
```

Use the following commands to run the “bluetoothctl” to show the Bluetooth MAC address and set up pairing mode for the EMIO-2531 module.

```
$ bluetoothctl
[NEW] Controller 5C:F3:70:24:4B:2E
[bluetooth]# scan on
Discovery started
[CHG] Controller 5C:F3:70:24:4B:2E Discovering: yes
[NEW] Device 5C:F3:70:25:DD:33 BlueZ 5.25
...
[bluetooth]# scan off
[bluetooth]# devices
Device 5C:F3:70:25:DD:33 BlueZ 5.25
Device ...
[bluetooth]# pair 5C:F3:70:25:DD:33 (pair device_mac_address)
[bluetooth]# trust 5C:F3:70:25:DD:33 (pair device_mac_address)
[bluetooth]# quit
```

SPP server mode:

If you do not change the listen channel number, the service will automatically use the default channel number. Make sure the server and client mode are using the same bluetooth channel number

To set up the SPP server mode and allow other devices to connect, use the following commands:

```
$ sdptool add SP
Serial Port service registered
$ rfcomm listen hci0
Waiting for connection on channel 1
Connection from 5C:F3:70:25:DD:33 to /dev/rfcomm0
Press CTRL-C for hangup
```

SPP client mode:

If you do not change the listen channel number, the service will automatically use the default channel number. Make sure the server or client mode is using the same listen channel number.

Use the following commands to set up the SPP client mode and request a connection to SPP server.

```
$ sdptool add SP
Serial Port service registered
$ rfcomm connect hci0 5C:F3:70:24:4B:2E
Connected /dev/rfcomm0 to 5C:F3:70:24:4B:2E on channel 1
Press CTRL-C for hangup
```

To open the minicom serial communication program, use the following command:

```
$ minicom -s
```

Connect to the `/dev/rfcomm0` serial device with the following settings:

```
+-----+
| A -      Serial Device           : /dev/rfcomm0 |
| B -      Lockfile Location       : /var/lock   |
| C -      Callin Program          :             |
| D -      Callout Program         :             |
| E -      Bps/Par/Bits            : 115200 8N1  |
| F -      Hardware Flow Control   : No         |
| G -      Software Flow Control   : No         |
+-----+
```

Serial port setting

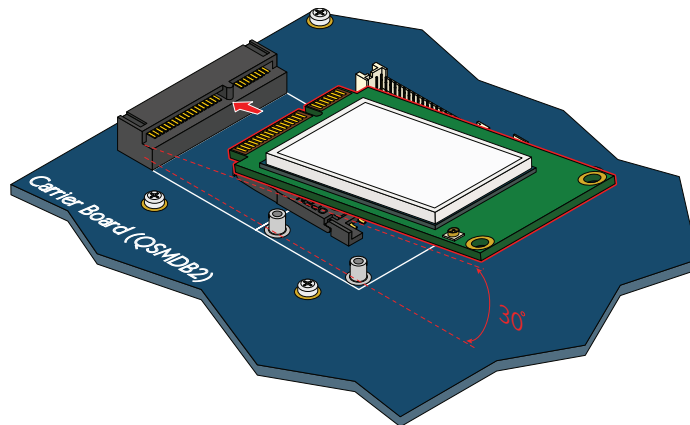
Start a serial communication program on the host machine with the same communication settings using the appropriate Bluetooth SPP device. Afterwards the Bluetooth SPP device and the QSM-8Q60 will be able to communicate through the programs (for example sending a keypress on one machine should be shown on the other machine).

4.4 Configuring the EMIO-2550 miniPCIe Mobile Broadband Module

The EMIO-2550 miniPCIe Mobile Broadband module supports 3G and GPS functions.

4.4.1 Connecting to the Internet

The first step is to insert an active SIM card into the EMIO-2550 module, and then insert the EMIO-2550 module into the miniPCIe slot. After installing the module connect the provided antenna to the module. Next, make sure to unplug any LAN cables or USB Wi-Fi dongles you have installed. Finally, power on the QSM-8Q60 and QSMDB2 carrier board.



Inserting the EMIO-2550 module

To check that the board has correctly detected the EMIO-2550 module, use the following command:

```
$ lsusb
```

Make sure the printout message includes “Bus 001 Device 004: ID 1545:1102 U-Blox AG”

Configure the 3G function by creating a PPP connection with root privilege.

```
$ pppd call Module-ZU200
```

Once the PPP connection has been created, open the browser to ensure connection to the Internet.

4.4.2 Enabling GPS

To configure the GPS functions of the module requires the use of the Minicom serial communication program. Open the Minicom serial communication program with the following command:

```
$ minicom -s
```

Connect to the **/dev/ttyUSB1** serial device with the following settings:

```
+-----+
| A -   Serial Device       : /dev/ttyUSB1 |
| B -   Lockfile Location   : /var/lock   |
| C -   Callin Program      :             |
| D -   Callout Program     :             |
| E -   Bps/Par/Bits        : 115200 8N1  |
| F -   Hardware Flow Control : No       |
| G -   Software Flow Control : No       |
+-----+
```

Serial port setting of QSM-8Q60 and QSMDB2 carrier board

When the setting is completed, type the following GPS AT commands into Minicom: AT+UGPRF=1, AT+UGRMC=1, AT+UGGSV=1, AT+UGZDA=1, and AT+UGPS=1,0

Wait for the “OK” message before sending the next AT command.

```
AT+UGPRF=1
OK
AT+UGRMC=1
OK
AT+UGGSV=1
OK
AT+UGZDA=1
OK
AT+UGPS=1,0
OK
```

Use the Minicom to set the serial device **/dev/ttyUSB7** in another terminal and it will output the data received from the GPS module as illustrated in the screenshot below.

```
$GPRMC,142015.00,A,2501.93961,N,12133.66111,E,0.068,,030314,,A*73
$GPGSV,4,1,13,01,34,184,29,03,49,023,47,06,26,042,41,07,44,317,26*75
$GPGSV,4,2,13,08,13,323,,11,60,192,,13,29,242,38,16,33,071,16*79
$GPGSV,4,3,13,19,65,356,44,23,18,208,12,27,35,034,27,30,37,145,29*7B
$GPGSV,4,4,13,32,01,154,*4B
$GPZDA,142015.00,03,03,2014,00,00*62
.....
```

GPS NMEA message



Taiwan Headquarters

1F, 531 Zhong-zheng Road,
Xindian Dist., New Taipei City 231
Taiwan

Tel: 886-2-2218-5452
Fax: 886-2-2218-9860
Email: embedded@via.com.tw



USA

940 Mission Court
Fremont, CA 94539,
USA

Tel: 1-510-687-4688
Fax: 1-510-687-4654
Email: embedded@viatech.com



Japan

3-15-7 Ebisu MT Bldg. 6F,
Higashi, Shibuya-ku
Tokyo 150-0011
Japan

Tel: 81-3-5466-1637
Fax: 81-3-5466-1638
Email: embedded@viatech.co.jp



China

Tsinghua Science Park Bldg. 7
No. 1 Zongguancun East Road,
Haidian Dist., Beijing, 100084
China

Tel: 86-10-59852288
Fax: 86-10-59852299
Email: embedded@viatech.com.cn



Europe

Email: embedded@via-tech.eu