



Qualcomm Technologies, Inc.

Qualcomm IQ-X5xxx Series Evaluation Kit - Linux

Bringup Guide

80-80022-297 Rev. AD

July 2, 2026

About Qualcomm

Qualcomm relentlessly innovates to deliver intelligent computing everywhere, helping the world tackle some of its most important challenges. Building on our 40 years of technology leadership in creating era-defining breakthroughs, we deliver a broad portfolio of solutions built with our leading-edge AI, high-performance, low-power computing, and unrivaled connectivity. Our Snapdragon® platforms power extraordinary consumer experiences, and our Qualcomm Dragonwing™ products empower businesses and industries to scale to new heights. Together with our ecosystem partners, we enable next-generation digital transformation to enrich lives, improve businesses, and advance societies. At Qualcomm, we are engineering human progress.

Revision history

Revision	Date	Description
AA	May 18, 2026	Initial release
AB	May 19, 2026	Updated the CCI level to white
AC	May 27, 2026	Added the following chapters <ul style="list-style-type: none">▪ Chapter 5: Configure Wi-Fi▪ Chapter 6: Perform Bluetooth GAP functions▪ Chapter 7: Bring up Ethernet▪ Chapter 8: Setup the Display Port
AD	July 2026	Numerous changes were made to the document. Read the document in its entirety.

Contents

1	Introduction	7
1.1	Conventions	7
1.2	Technical assistance	7
2	IQ-X5xxx series carrier board–peripheral interface connectors	8
3	Configure DIP switch settings	10
4	Download and flash software	12
4.1	Power up and power cycle	12
4.1.1	Configure boot mode	13
4.1.2	Power cycle the EVK	13
4.2	Flash the software images to the Dragonwing IQ-X5xxx EVK	14
4.2.1	Prerequisites	14
4.2.2	Set up the Ubuntu host	14
4.2.3	Download flashable binaries	15
5	Configure Wi-Fi	17
5.1	Start the Wi-Fi subsystem	17
5.2	Scan for Wi-Fi APs	17
5.3	Set up a Wi-Fi station	18
6	Perform Bluetooth GAP functions	21
6.1.1	Enable Bluetooth	21
6.2	Run Bluetooth inquiry scan	22
6.3	Stop Bluetooth inquiry scan	23
6.4	Pair with a remote Bluetooth device	23
6.5	Get the bonded/paired device list	24
6.6	Unpair a device	25
6.7	Enable device discovery	25
6.8	Disable Bluetooth	25
7	Bring up Ethernet	26
8	Set up the DisplayPort	28
9	FAQs	31
9.1	How do you perform configuration data table (CDT) flashing?	31

9.2 How do you provision universal flash storage (UFS)? 31

9.3 Further support 32

A References 33

 A.1 Related documents 33

 A.2 Acronyms and terms 33

Figures

Figure 2-1 Top view of IQ-X5xxx series carrier board	8
Figure 2-2 Bottom view of IQ-X5xxx series carrier board	9
Figure 3-1 DIP switch settings for default SPINOR + UFS and Bluetooth setup.....	10
Figure 4-1 Power up options on IQ-X5xxx series carrier board	12
Figure 4-2 Connecting the IQ-X5xxx board to the host computer	14
Figure 7-1 Ethernet bringup workflow.....	26
Figure 8-1 DisplayPort setup workflow.....	28

Tables

Table 3-1 DIP switch settings	10
Table 3-2 S2, S3, and S4 DIP switches	11
Table 4-1 DIP switch settings	13
Table 4-2 DIP switch setting for Boot mode configuration.....	13

1 Introduction

The IQ-X5xxx series evaluation kit (EVK) includes a carrier board and a module. This bring-up guide explains how to set up the IQ-X5xxx series EVK, understand the peripheral interfaces, and perform software flashing.

NOTE: This document is a preview (beta) version. The production version will be released later. Features such as audio, video, and camera are not included in this guide and will be added in future revisions.

For information about the IQ-X5xxx series module, see the documents listed in [Related documents](#).

1.1 Conventions

Function declarations, function names, type declarations, attributes, and code samples appear in a different font, for example, `cp armcc armcpp`.

Code variables appear in angle brackets, for example, `<number>`.

Commands to be entered appear in a different font, for example, **copy a:.* b:**. Button and key names appear in bold font, for example, click **Save** or press **Enter**.

1.2 Technical assistance

For assistance or clarification on information in this document, go to www.qualcomm.com/support

2 IQ-X5xxx series carrier board–peripheral interface connectors

The following figure shows the location of various peripheral interface connectors from the top view of the IQ-X5xxx series carrier board.

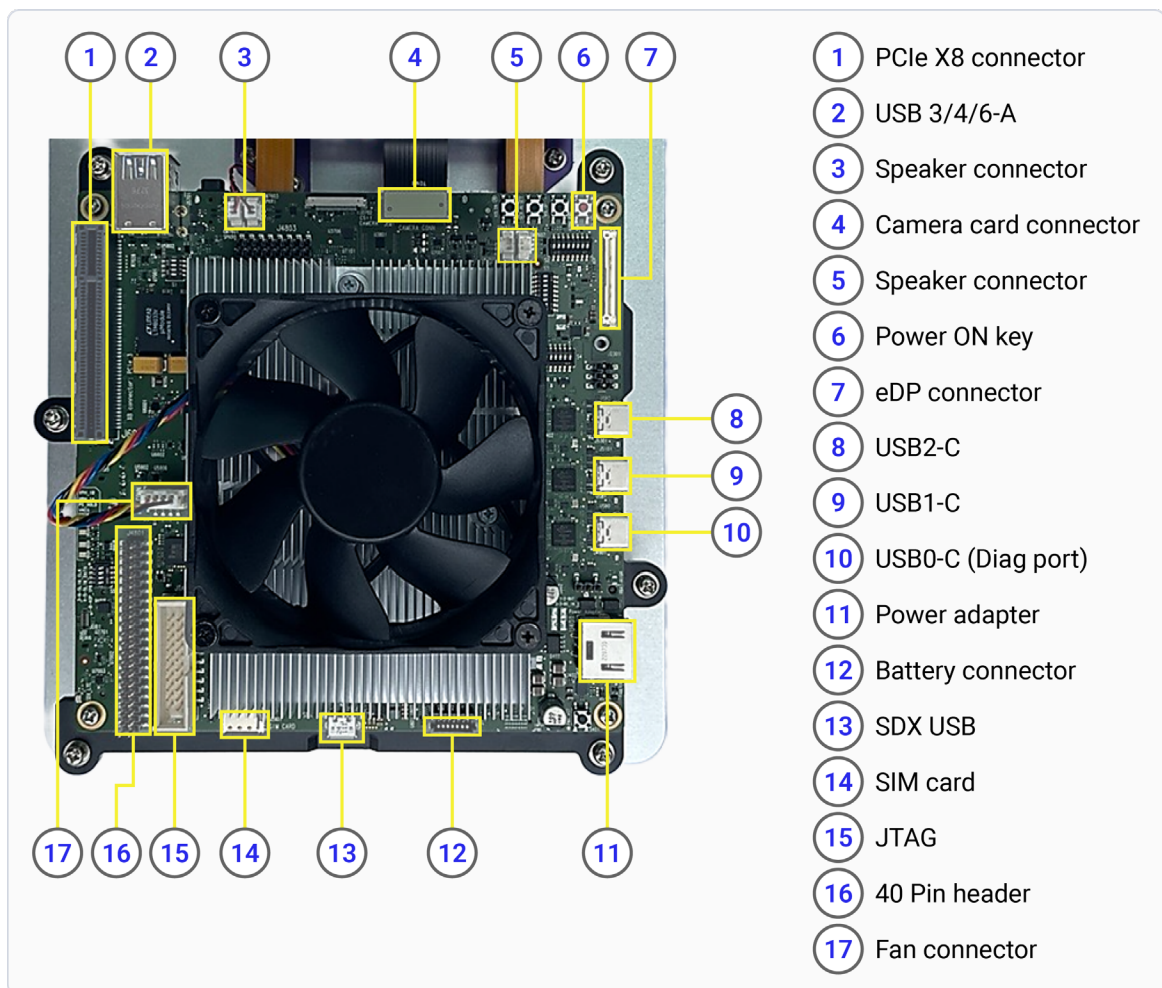


Figure 2-1 Top view of IQ-X5xxx series carrier board

The following figure shows the location of various peripheral interface connectors from the bottom view of the IQ-X5xxx series carrier board.

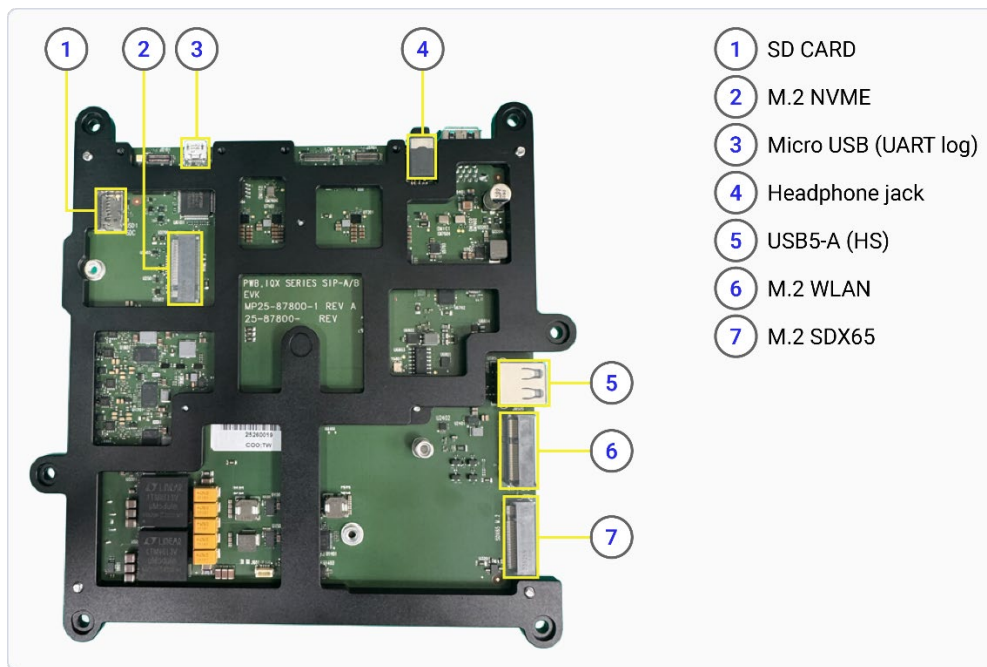


Figure 2-2 Bottom view of IQ-X5xxx series carrier board

3 Configure DIP switch settings

The following figure shows the dual in-line package (DIP) switch settings to set up the default SPINOR and universal flash storage (UFS) mode on the IQ-X5xxx series carrier board.

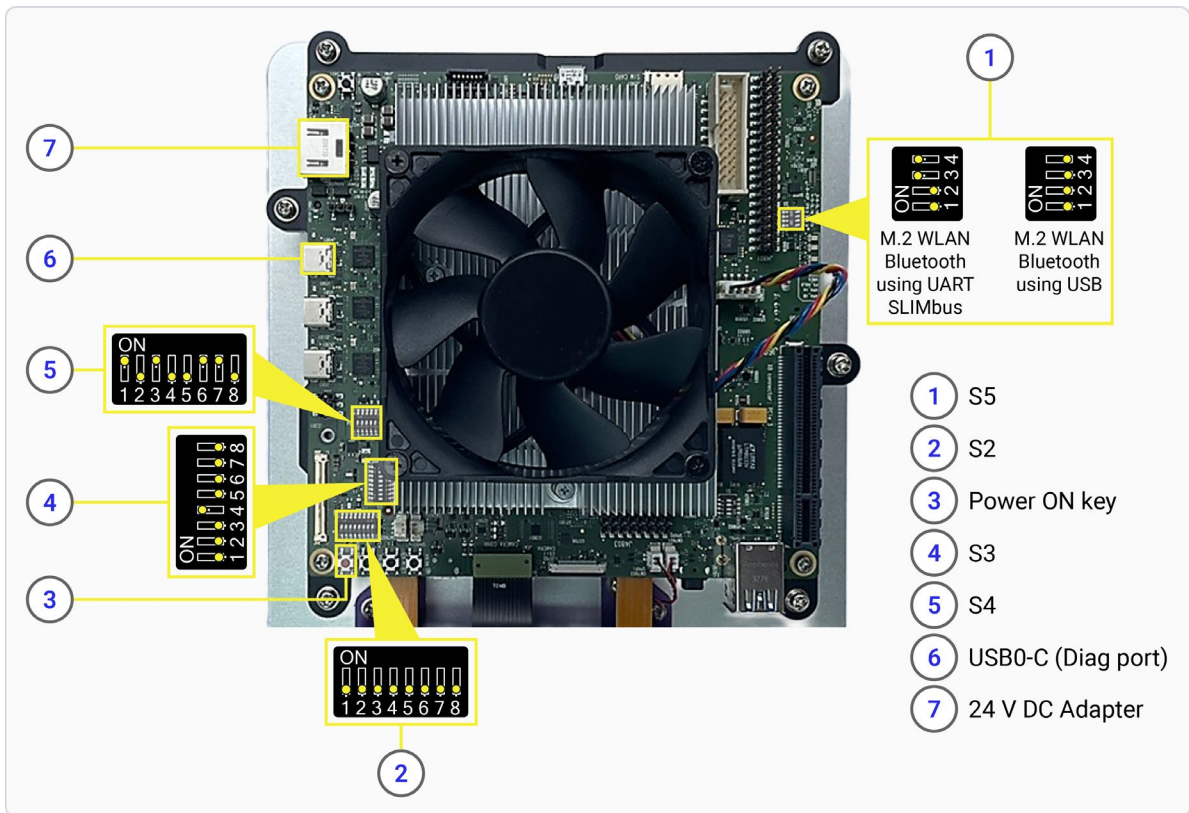


Figure 3-1 DIP switch settings for default SPINOR + UFS and Bluetooth setup

The following table describes the ON and OFF positions of the DIP switches for the SPINOR and UFS mode.

Table 3-1 DIP switch settings

Switch	Description	
	ON	OFF
S2-4	Emergency Download (EDL) mode	Normal mode
S3-4	FAST BOOT [0] → 1	FAST BOOT [0] → 0
S4-1	FAST BOOT [2] → 1	FAST BOOT [2] → 0
S4-3	FAST BOOT [1] → 1	FAST BOOT [1] → 0

Switch	Description	
	ON	OFF
S4-5	BATT_THERM for fake battery	NC
S4-6	Enables BAT_ID resistor on board	Uses BAT_ID resistor in battery
S4-7		
S5-3	SLIMBUS_SEL_1	Disconnect SLIMBUS CLK, Bluetooth over USB
S5-4	SLIMBUS_SEL_2	Disconnect SLIMBUS DATA, Bluetooth over USB

The following table lists the description of the S2, S3, and S4 DIP switches:

Table 3-2 S2, S3, and S4 DIP switches

Pins	Function
DIP switch S2	
8-9	Enumerator enable
7-10	Enumerator enable
6-11	CSI1 selection, select 22 pin camera
5-12	APPS PBL Boot SPEED (1)
4-13	FORCE USB BOOT
3-14	APPS PBL BOOT SPEED (0)
2-15	SYSTEM_THROTTLE
1-16	SDX_FORCE_USB
DIP switch S3	
8-9	DDR3_CH_SEL
7-10	SDX_PCIE_INITPCIE, early init bypass
6-11	Disable input power regulator
5-12	WDOG_DISABLE
4-13	FAST BOOT [0]→1
3-14	FORCE_PS_HOLD
2-15	Disable input power regulator
1-16	AUTO_PWR_EN
DIP switch S4	
8-9	UART_SEL
7-10	BATT_ID
6-11	BATT_ID
5-12	BATT_THERM
4-13	DEAD_BATT_CHG
3-14	FAST BOOT [1]→1
2-15	FAST BOOT [3]→1
1-16	FAST BOOT [2]→1

4 Download and flash software

This chapter describes how to download and flash the software on the IQ-X5xxx series carrier board.

4.1 Power up and power cycle

The IQ-X5xxx series carrier board is preloaded with software to set up the device. If no image is loaded, the device enters into EDL mode when powered on. The following figure shows the location of the interfaces used to power up the device:

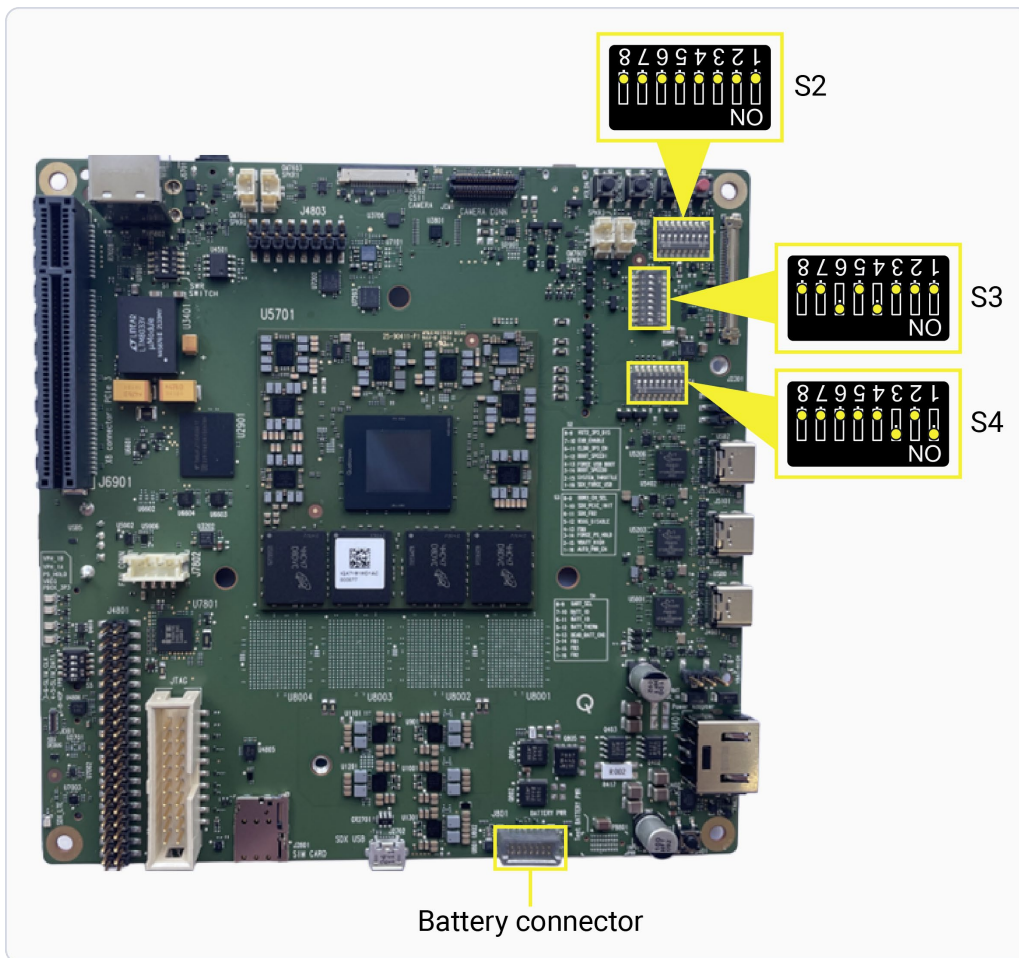


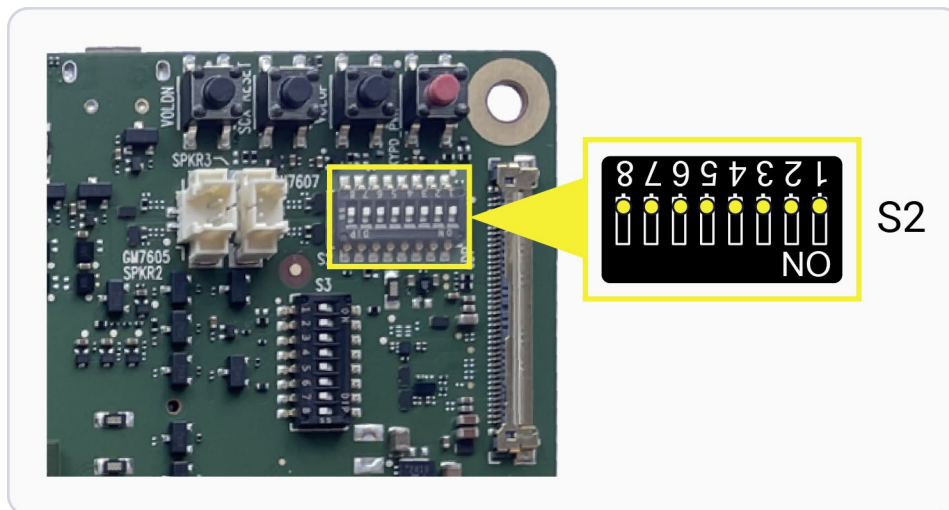
Figure 4-1 Power up options on IQ-X5xxx series carrier board

Table 4-1 DIP switch settings

Switch	Description	
	ON	OFF
S2-4	EDL mode	Normal mode
S3-4	FAST BOOT [0] → 1	FAST BOOT [0] → 0
S4-1	FAST BOOT [2] → 1	FAST BOOT [2] → 0
S4-3	FAST BOOT [1] → 1	FAST BOOT [1] → 0
S4-2	FAST BOOT [3] → 1	FAST BOOT [3] → 0
S4-6	Enables BAT_ID resistor on board	Uses BAT_ID resistor in battery
S4-7		

4.1.1 Configure boot mode

Boot mode configuration of the S2-4 DIP switch: For the EDL mode, turn ON S2-4 and for Normal mode, turn OFF S2-4.

**Table 4-2 DIP switch setting for Boot mode configuration**

Switch	Description	
	ON	OFF
S2-4	EDL mode	Normal mode

4.1.2 Power cycle the EVK

To power cycle the EVK, do the following:

1. Disconnect the power source (DC adapter, battery, Type C PD).
2. Disconnect the USB Type-C cable (USB0-C, USB1-C, USB2-C) (if connected).
3. Reconnect the power source.

4.2 Flash the software images to the Dragonwing IQ-X5xxx EVK

Download and integrate the software images, and flash them onto the Dragonwing IQ-X5xxx EVK.

4.2.1 Prerequisites

Ensure that you have the following required hardware and peripherals ready:

Requirement	Description
Host computer OS	Ubuntu OS
Development kit	Dragonwing IQ-X5xxx EVK
USB cable	One USB Type-C to USB Type-A cable for flashing the image to the Dragonwing IQ-X5xxx EVK
Serial Cable	One Micro-USB to USB Type-A cable for the UART debug port

Follow the setup steps based on your host computer's operating system (OS).

4.2.2 Set up the Ubuntu host

To set up the environment in the Ubuntu OS, set up the debug UART.

1. Connect one end of a Micro-USB cable to the Micro-USB port on the Dragonwing IQ-X5xxx EVK.



Figure 4-2 Connecting the IQ-X5xxx board to the host computer

2. Connect the other end of the Micro-USB cable to the host computer.
3. Install the `Screen` application on the Linux host computer:


```
$ sudo apt update
$ sudo apt install screen
```
4. Verify the USB port:


```
$ ls /dev/ttyUSB*
```

Sample output

```
$ /dev/ttyUSB0
```

5. Open the debug UART:


```
$ sudo screen <serial_port> <baud_rate>
```

Example command:

```
$ sudo screen /dev/ttyUSB0 115200
```

4.2.3 Download flashable binaries

To download the no-login binaries for the Dragonwing IQ-X5xxx EVK, and to set up the recommended directory structure and extraction process, do the following:

1. Create a workspace and then go to that directory.

```
$ mkdir iqx_workspace
$ cd iqx_workspace
```

2. Download and unzip the binaries.

```
$ wget https://artifacts.codelinearo.org/artifactory/qli-ci/flashable-
binaries/meta-qcom/iq-x5121-evk/qli-2.0-qcom-multimedia-proprietary-
image.zip
$ unzip qli-2.0-qcom-multimedia-proprietary-image.zip
```

4.2.3.1 Flash the Dragonwing IQ-X5xxx EVK integrated image on Ubuntu host

To flash the Dragonwing IQ-X5xxx EVK integrated image using Ubutnu host, do the following:

1. Move the Dragonwing IQ-X5xxx EVK into the [Emergency Download \(EDL\)](#) mode.
 - a. Connect the Dragonwing IQ-X5xxx EVK to a 12 V wall power supply.
 - b. Connect the Dragonwing IQ-X5xxx EVK to the host computer through the USB Type-C connector.
 - c. Verify whether the Dragonwing IQ-X5xxx EVK has entered the EDL mode by running the following command on the host computer:

```
$ lsusb
```

Sample output

```
Bus 002 Device 014: ID 05c6:9008 Qualcomm, Inc. Gobi Wireless Modem
(QDL mode).
```

2. Update the udev rules.
 - a. Go to the udev configuration directory:

```
$ cd /etc/udev/rules.d
```
 - b. List the contents of the directory:

```
$ ls
```

 - i If the `51-qcom-usb.rules` file isn't present, run the following command to create it.

```
$ sudo vi 51-qcom-usb.rules
```
 - ii Add the following content to the file `51-qcom-usb.rules`:

```
SUBSYSTEMS=="usb", ATTRS{idVendor}=="05c6",
ATTRS{idProduct}=="9008", MODE="0664", GROUP="plugdev"
```
 - iii If the file exists, check for the earlier content. Run the following command to view the content:

```
$ cat 51-qcom-usb.rules
```
 - c. Restart udev:

```
$ sudo systemctl restart udev
```

If the USB cable is already connected to the host computer, disconnect and reconnect it to apply the updated rules.

3. Use the **QDL** tool to flash the software onto the Dragonwing IQ-X5xxx EVK:

NOTE: Use QDL version 2.4.0 or later to flash the Dragonwing IQ-X5xxx EVK.

- a. Download the [QDL tool](#) and unzip the contents of the downloaded folder.
- b. Provide executable permission:

```
$ chmod +x ./qdl
```

NOTE: For more information about QDL tool usage, see the **QDL_User_Guide** in your unzipped `qdl` directory.

NOTE: To perform UFS flashing, switch on SW4-1.

- c. Perform [configuration data table \(CDT\) flashing](#)
- d. Perform [universal flash storage \(UFS\) provisioning](#)
- e. Flash the boot binaries to spinor.
 - i. Go to the IQ-X5xxx workspace

```
iqx_workspace/images/ iq-x5121-evk/qcom-multimedia-proprietary-image-iq-x5121-evk/spinor
```
 - ii. Flash Spinor using the command:

```
<qdl_tool_path>/qdl --storage spinor xbl_s_devprg_ns.melf rawprogram0.xml patch0.xml
```
- f. Flash the RC3 firmware
 - i. Go to the IQ-X5xxx workspace

```
iqx_workspace/iq-x5121-evk/qcom-multimedia-proprietary-image-iq-x5121-evk
```
 - ii. Flash the RC3 firmware using the command:

```
<qdl_tool_path>/ qdl --storage ufs xbl_s_devprg_ns.melf rawprogram0.xml rawprogram1.xml patch0.xml patch1.xml
```

NOTE: Turn off the SW2-4 DIP switch after flashing the software

5 Configure Wi-Fi

This information describes the setup and configuration procedures required to establish a Wi-Fi connection.

5.1 Start the Wi-Fi subsystem

If the device is started in STA mode, Wi-Fi is active by default, that is, the Wi-Fi host driver and `wpa_supplicant` are launched during the boot process of the device.

1. Ensure that the device is connected to the host over UART.
2. To confirm if the Wi-Fi host driver is loaded, run the following command on the debug UART console:
`iw dev`
3. Search for the `wlp4p1s0` interface in the output:

Sample output

```
root@iq-x5121-evk:~# iw dev
phy#0
Interface wlp4p1s0
ifindex 2
wdev 0x1
addr 00:03:7f:58:74:92
type managed
multicast TXQ:
qsz-byt qsz-pkt flows drops marks overlmt hashcol tx-bytes tx-packets
0 0 0 0 0 0 0 0 0
```

5.2 Scan for Wi-Fi APs

A Wi-Fi scan finds the APs listed in the vicinity that must be initiated through the `nmcli` command line tool.

To initiate a Wi-Fi scan using `nmcli` run the following command:

```
nmcli dev wifi list 2>&1 | less
```

Sample output

IN-USE	BSSID	SSID	MODE	CHAN	RATE	SIGNAL
BARS	SECURITY					
****	20:DB:AB:98:57:EE	JioPrivateNet	Infra	9	130 Mbit/s	89
		WPA2 802.1X				

```

20:DB:AB:98:57:EF JioNet Infra 9 130 Mbit/s 89
**** --
20:DB:AB:98:57:E1 JioPrivateNet Infra 44 270 Mbit/s 80
*** WPA2 802.1X
20:DB:AB:98:57:E0 JioNet Infra 44 270 Mbit/s 77
*** --
F0:61:C0:FB:96:A1 QGuest Infra 11 260 Mbit/s 75
*** --
F0:61:C0:FB:96:A0 Hydra Infra 11 260 Mbit/s 75
*** WPA2
20:DB:AB:9E:CA:CF JioNet Infra 5 130 Mbit/s 72
*** --
F0:61:C0:FB:96:B0 Hydra Infra 140 540 Mbit/s 72
*** WPA2
F0:61:C0:FB:96:B1 Pandora Infra 140 540 Mbit/s 72
*** WPA2 802.1X
F0:61:C0:FB:96:B2 QGuest Infra 140 540 Mbit/s 72
*** --
20:DB:AB:9E:CA:CE JioPrivateNet Infra 5 130 Mbit/s 69
***

```

NOTE: To exit from the standard input shell, press CTRL+C.

5.3 Set up a Wi-Fi station

In the STA mode, the IQ-X5xxx device operates as a client and it can connect to a wireless hotspot or router.

Establish a wireless connection through the `nmcli` command line tool.

To set up the Wi-Fi in STA mode, perform the following steps:

1. Ensure that the device is connected to the host over UART.
2. To establish a connection to an AP, run the following command:
`nmcli dev wifi connect <WiFi-SSID> password <WiFi-password>`

For example, run the command

```
root@iq-x5121-evk:~# nmcli dev wifi connect QualcommWiFi password
1234567890
```

where, the Wi-Fi SSID is `QualcommWiFi` and the Wi-Fi password is `1234567890`.

Sample output

```
Device 'wlP4p1s0' successfully activated with 'df4250eb-45f6-4ce2-bd90-
a2513e016536'
```

NOTE: If you see a network error message while running the command, then run one of the following commands to trigger the Wi-Fi scan and verify the intended AP.

```
nmcli dev wifi list
iw dev wlP4p1s0 scan
```

3. To verify the connection state, run the following command:
`root@iq-x5121-evk:~# nmcli general status`

Sample output

```
STATE CONNECTIVITY WIFI-HW WIFI WWAN-HW WWAN
connected full enabled enabled enabled enabled
```

4. To verify the connection status, perform the following steps:

a. To view the device status, run the following command:

```
root@iq-x5121-evk:~# nmcli dev status
```

Sample output

```
DEVICE TYPE STATE CONNECTION
wlp1s0 wifi connected QualcommWiFi
eth0 ethernet unavailable --
eth1 ethernet unavailable --
can0 can unmanaged --
lo loopback unmanaged --
```

b. To view the additional connection information, run the following command:

```
nmcli device show wlp4pls0
```

Sample output

```
GENERAL.DEVICE: wlp4pls0
GENERAL.TYPE: wifi
GENERAL.HWADDR: 00:03:7F:12:F7:F7
GENERAL.MTU: 1500
GENERAL.STATE: 100 (connected)
GENERAL.CONNECTION: QualcommWiFi
GENERAL.CON-PATH: /org/freedesktop/NetworkManager/ActiveConnection/5
IP4.ADDRESS[1]: 192.168.117.130/24
IP4.ADDRESS[2]: 192.168.117.131/24
IP4.GATEWAY: 192.168.117.126
IP4.ROUTE[1]: dst = 192.168.117.0/24, nh = 0.0.0.0, mt = 3005
IP4.ROUTE[2]: dst = 192.168.117.0/24, nh = 0.0.0.0, mt = 600
IP4.ROUTE[3]: dst = 0.0.0.0/0, nh = 192.168.117.126, mt = 3005
IP4.DNS[1]: 192.168.117.126
IP6.ADDRESS[1]: 2401:4900:658c:d8b0:3a86:b071:fd59:7ade/64
IP6.ADDRESS[2]: 2401:4900:658c:d8b0:37d5:d37f:675c:3313/64
IP6.ADDRESS[3]: fe80::c930:1be0:3ac0:496c/64
IP6.ADDRESS[4]: fe80::28a6:3dc0:f535:75f1/64
IP6.GATEWAY: fe80::14c1:74ff:feef:f40f
IP6.ROUTE[1]: dst = fe80::/64, nh = ::, mt = 256
IP6.ROUTE[2]: dst = fe80::/64, nh = ::, mt = 1024
IP6.ROUTE[3]: dst = 2401:4900:658c:d8b0::/64, nh = ::, mt = 3005
```

```
IP6.ROUTE[4]:                               dst = ::/0, nh =
fe80::14c1:74ff:feef:f40f, mt = 3005
IP6.DNS[1]:                                 2401:4900:658c:d8b0::dc
```

- c. Verify if the IP address is assigned on the wlp4p1s0 interface by running the `ip addr show wlp4p1s0` command in the UART console.
- ```
root@iq-x5121-evk:~# ip addr show wlp4p1s0
```

#### Sample output

```
root@iq-x5121-evk:~# ip addr show wlp4p1s0
2: wlp4p1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
state UP group default qlen 1000
 link/ether 00:03:7f:12:41:41 brd ff:ff:ff:ff:ff:ff
 altname wlx00037f124141
 inet 10.92.142.49/21 brd 10.92.143.255 scope global dynamic
noprofixroute wlp4p1s0
 valid_lft 1788sec preferred_lft 1788sec
 inet6 fe80::6726:c37b:bd85:278b/64 scope link noprefixroute
 valid_lft forever preferred_lft forever
```

- d. Verify if the AP or router is connected to the Internet by pinging the following public DNS IP:
- ```
root@iq-x5121-evk:~# ping 8.8.8.8
```

Sample output

```
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=55 time=174 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=55 time=96.9 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=55 time=84.8 ms
^C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 84.809/118.472/173.684/39.352 ms
```

NOTE: If you are connected to one network, but want to use a different connection, you can disconnect by switching the connection to down by specifying the SSID `nmcli con down SSID/UUID`. Alternatively, if you have multiple connections with the same SSID, use the UUID.

NOTE: To connect to another saved connection, simply pass the up option in the `nmcli` command line tool by running the following command.

```
nmcli con up SSID/UUID
```

Ensure that you specify the SSID or UUID of the new network that you want to connect to.

NOTE: To exit from the standard input shell, press CTRL+C.

For more information about `nmcli`, see <https://www.linux.org/docs/man1/nmcli.html> and <https://networkmanager.dev/docs/api/latest/nmcli.html>.

6 Perform Bluetooth GAP functions

You can perform various Bluetooth GAP functions using the commands provided in the main menu of `bluetoothctl`. Bluetooth GAP functions include enabling and disabling Bluetooth, running and stopping inquiry scans to discover nearby devices, and enabling device discovery mode. GAP also supports pairing with remote Bluetooth devices, viewing the list of bonded or paired devices, and unpairing of devices.

6.1.1 Enable Bluetooth

1. Open the Bluetooth test application by running the following command:

```
bluetoothctl
```

Sample output:

```
root@iq-x5121-evk:~# bluetoothctl
```

```
[NEW] Media /org/bluez/hci1
```

```
    SupportedUUIDs: 0000110a-0000-1000-8000-00805f9b34fb
```

```
    SupportedUUIDs: 0000110b-0000-1000-8000-00805f9b34fb
```

```
Agent registered
```

```
[CHG] Controller 8C:FD:F0:58:82:03 Pairable: yes
```

```
hci1 new_settings: powered bondable ssp br/edr le secure-conn wide-band-  
speech cis-central cis-peripheral iso-broadcaster sync-receiver ll-  
privacy
```

2. To view GAP functions, run the following command:\

```
Help
```

This command provides the main menu of `bluetoothctl`. The main menu includes the submenus and GAP functions.

Sample output

```
[bluetoothh]# help
```

```
Menu main:
```

```
Available commands:
```

```
-----
```

```
advertise
```

```
Advertise Options Submenu
```

```
monitor
```

```
Advertisement Monitor Options Submenu
```

```
scan
```

```
Scan Options Submenu
```

```
gatt
```

```
Generic Attribute Submenu
```

```
admin
```

```
Admin Policy Submenu
```

```
player
```

```
Media Player Submenu
```

```
endpoint
```

```
Media Endpoint Submenu
```

```
transport
```

```
Media Transport Submenu
```

```
list
```

```
List available controllers
```

```
show [ctrl]
```

```
Controller information
```

select <ctrl>	Select default controller
devices [Paired/Bonded/Trusted/Connected]	List available
devices, with an optional property as the filter	
system-alias <name>	Set controller alias
reset-alias	Reset controller alias
power <on/off>	Set controller power
pairable <on/off>	Set controller pairable mode
discoverable <on/off>	Set controller discoverable mode
discoverable-timeout [value]	Set discoverable timeout
agent <on/off/capability>	Enable/disable agent with given capability
default-agent	Set agent as the default one
advertise <on/off/type>	Enable/disable advertising with given type
set-alias <alias>	Set device alias
scan <on/off/bredr/le>	Scan for devices
info [dev]	Device information
pair [dev]	Pair with device
cancel-pairing [dev]	Cancel pairing with device
trust [dev]	Trust device
untrust [dev]	Untrust device
block [dev]	Block device
unblock [dev]	Unblock device
remove <dev>	Remove device
connect <dev>	Connect device
disconnect [dev]	Disconnect device
menu <name>	Select submenu
version	Display version
quit	Quit program
exit	Quit program
help	Display help about this program
export	Print environment variables

- To enable Bluetooth on the device, run the following command:

```
power on
```

Sample output

```
[bluetoothctl]> power on
Changing power on succeeded
```

6.2 Run Bluetooth inquiry scan

- To start an inquiry for nearby devices, run the following command:

```
scan on
```

Sample output

```
[bluetooth]# scan on
Discovery started
[CHG] Controller 22:22:9B:2C:79:1E Discovering: yes
[NEW] Device A4:30:7A:EE:AF:EF [TV] MyDeviceA 8 Series (43)
[NEW] Device 7E:08:AE:BC:66:58 7E-08-AE-BC-66-58
```

```

[CHG] Device A4:30:7A:EE:AF:EF RSSI: -91
[CHG] Device A4:30:7A:EE:AF:EF Modalias: bluetooth: v04E8p8080d0000
[CHG] Device A4:30:7A:EE:AF:EF UUIDs: 0000110a-0000-1000-8000-00805f9b34fb
[CHG] Device A4:30:7A:EE:AF:EF UUIDs: 0000110b-0000-1000-8000-00805f9b34fb
[CHG] Device A4:30:7A:EE:AF:EF UUIDs: 0000110c-0000-1000-8000-00805f9b34fb
[CHG] Device A4:30:7A:EE:AF:EF UUIDs: 0000110e-0000-1000-8000-00805f9b34fb
[CHG] Device A4:30:7A:EE:AF:EF UUIDs: 00001112-0000-1000-8000-00805f9b34fb
[CHG] Device A4:30:7A:EE:AF:EF UUIDs: 0000111f-0000-1000-8000-00805f9b34fb
[CHG] Device A4:30:7A:EE:AF:EF UUIDs: 00001200-0000-1000-8000-00805f9b34fb
[CHG] Device A4:30:7A:EE:AF:EF ManufacturerData Key: 0xff19
[CHG] Device A4:30:7A:EE:AF:EF ManufacturerData Value:
00 75 00 09 01 00 00 00 06 01 00 00 00 00 00 00 .u.....
00 00 00 00 00 00 00 00 .....
[NEW] Device F8:7D:76:9D:9B:6B MyDeviceB
[CHG] Device 7E:08:AE:BC:66:58 RSSI: -73

```

6.3 Stop Bluetooth inquiry scan

To stop an inquiry that's in progress, run the following command:

```
scan off
```

Sample output

```

[bluetooth]# scan off
Discovery stopped
[CHG] Device 00:95:01:ED:A3:1C RSSI is nil
[DEL] Device 00:95:01:ED:A3:1C 00-95-01-ED-A3-1C
[CHG] Device 8C:FD:F0:0F:41:DB RSSI is nil
[DEL] Device 8C:FD:F0:0F:41:DB 8C-FD-F0-0F-41-DB
[CHG] Device 3F:5C:33:B2:F4:13 RSSI is nil
[DEL] Device 3F:5C:33:B2:F4:13 3F-5C-33-B2-F4-13
[CHG] Device C8:12:0B:50:F2:51 RSSI is nil
[DEL] Device C8:12:0B:50:F2:51 C8-12-0B-50-F2-51
[CHG] Device 8C:FD:F0:0F:1B:B8 RSSI is nil
[DEL] Device 8C:FD:F0:0F:1B:B8 8C-FD-F0-0F-1B-B8
[CHG] Device 0E:19:D7:85:53:8A RSSI is nil
[DEL] Device 0E:19:D7:85:53:8A 0E-19-D7-85-53-8A
[CHG] Device 75:AD:91:DD:40:2A TxPower is nil
[CHG] Device 75:AD:91:DD:40:2A RSSI is nil
[CHG] Device E4:24:2C:94:28:BC RSSI is nil
[CHG] Device 94:7C:00:B0:38:28 RSSI is nil
[CHG] Controller 8C:FD:F0:21:84:23 Discovering: no

```

6.4 Pair with a remote Bluetooth device

Before you pair a remote device, run a [Bluetooth inquiry scan](#) to ensure that the remote device is available.

To pair with a remote Bluetooth device, run the following command:

```
pair <bt_address>
```

To accept the outgoing/incoming pairing, enter `yes`. To reject the outgoing/incoming pairing, enter `no`.

Parameters

<bt_address> is the Bluetooth address of the remote device.

Example

To pair with a remote device with <bt_address> `F8:7D:76:9D:9B:6B`, run the following command:

```
pair F8:7D:76:9D:9B:6B
```

Sample output

```
[bluetooth]# pair F8:7D:76:9D:9B:6B
Attempting to pair with F8:7D:76:9D:9B:6B
[CHG] Device F8:7D:76:9D:9B:6B Connected: yes
[CHG] Device F8:7D:76:9D:9B:6B Name: MyDeviceB
[CHG] Device F8:7D:76:9D:9B:6B Alias: MyDeviceB
Request /*-9
[agent] Confirm passkey 068560 (yes/no): yes
[CHG] Device F8:7D:76:9D:9B:6B Bonded: yes
[DEL] Device 28:DE:65:7B:59:54 28-DE-65-7B-59-54
[CHG] Device F8:7D:76:9D:9B:6B Modalias: bluetooth:v004Cp760Ad1160
[CHG] Device F8:7D:76:9D:9B:6B UUIDs: 00000000-deca-fade-deca-deafdecacafe
[CHG] Device F8:7D:76:9D:9B:6B UUIDs: 00001000-0000-1000-8000-00805f9b34fb
[CHG] Device F8:7D:76:9D:9B:6B UUIDs: 0000110a-0000-1000-8000-00805f9b34fb
[CHG] Device F8:7D:76:9D:9B:6B UUIDs: 0000110c-0000-1000-8000-00805f9b34fb
[CHG] Device F8:7D:76:9D:9B:6B UUIDs: 0000110e-0000-1000-8000-00805f9b34fb
[CHG] Device F8:7D:76:9D:9B:6B UUIDs: 00001116-0000-1000-8000-00805f9b34fb
[CHG] Device F8:7D:76:9D:9B:6B UUIDs: 0000111f-0000-1000-8000-00805f9b34fb
[CHG] Device F8:7D:76:9D:9B:6B UUIDs: 0000112f-0000-1000-8000-00805f9b34fb
[CHG] Device F8:7D:76:9D:9B:6B UUIDs: 00001132-0000-1000-8000-00805f9b34fb
[CHG] Device F8:7D:76:9D:9B:6B UUIDs: 00001200-0000-1000-8000-00805f9b34fb
[CHG] Device F8:7D:76:9D:9B:6B UUIDs: 00001801-0000-1000-8000-00805f9b34fb
[CHG] Device F8:7D:76:9D:9B:6B UUIDs: 02030302-1d19-415f-86f2-22a2106a0a77
[CHG] Device F8:7D:76:9D:9B:6B UUIDs: 1ff31936-572e-4b36-a2bf-b2409b1aa6f4
[CHG] Device F8:7D:76:9D:9B:6B ServicesResolved: yes
[CHG] Device F8:7D:76:9D:9B:6B Paired: yes
Pairing successful
[DEL] Device 28:DE:65:7B:5B:71 28-DE-65-7B-5B-71
[CHG] Device F8:7D:76:9D:9B:6B ServicesResolved: no
[CHG] Device F8:7D:76:9D:9B:6B Connected: no
```

6.5 Get the bonded/paired device list

To get a verified list of paired devices, run the following command:

```
devices
```

Sample output

```
[bluetooth]# devices
Device F8:7D:76:9D:9B:6B MyDeviceB
```

6.6 Unpair a device

To unpair a device, run the following command:

Example

To unpair a device with the address F8:7D:76:9D:9B:6B, run the following command:

```
remove F8:7D:76:9D:9B:6B
```

Sample output

```
[bluetooth]# remove F8:7D:76:9D:9B:6B
[DEL] Device F8:7D:76:9D:9B:6B MyDeviceB
Device has been removed
```

6.7 Enable device discovery

To enable discovery mode in the DUT, run the following command:

```
discoverable on
```

Sample output

```
[bluetooth]# discoverable on
Changing discoverable on succeeded
[CHG] Controller 8C:FD:F0:21:84:23 Discoverable: yes
```

6.8 Disable Bluetooth

To disable Bluetooth on the device, run the following command:

```
power off
```

Sample output

```
[bluetooth]# power off
[CHG] Controller 8C:FD:F0:21:84:23 Discoverable: no
Changing power off succeeded
[CHG] Controller 8C:FD:F0:21:84:23 Powered: no
[CHG] Controller 8C:FD:F0:21:84:23 Discovering: no
[CHG] Controller 8C:FD:F0:21:84:23 Class: 0x00000000
```

7 Bring up Ethernet

The following figure shows the workflow to bring up Ethernet on the reference kits.

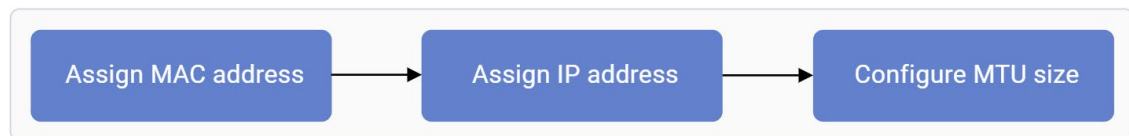


Figure 7-1 Ethernet bringup workflow

To bring up Ethernet functionality on the IQ-X5xxx Development Kit, do the following:

1. The IQ-X5xxx Development Kit comes with a preconfigured MAC address. Skip this step if you choose to use the same MAC address.
Alternatively, to change the preconfigured MAC address on the device, perform the following steps:
 - a. **Bring down the interface.**
`ip link set <Interface_name> down`
For example:
`ip link set enu1 down`
 - b. **Assign the MAC address.**
`ip link set dev <Interface_name> address <MAC>`
For example:
`ip link set dev enu1 address XX:XX:XX:YY:YY:YY`
 - c. **Bring up the interface.**
`ip addr add <ip>/<prefix> dev <Interface_name>`
`ip link set dev <Interface_name> up`
For example:
`ip addr add 169.254.227.235/16 dev enu1`
`ip link set dev enu1 up`

Sample output:

```
ip -s addr show dev enu1
```

```
enu1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP
group default qlen 1000
    link/ether 62:99:60:23:4d:52 brd ff:ff:ff:ff:ff:ff
    RX:  bytes  packets  errors  dropped  missed  mcast
         105971    313      0        2         0        0
    TX:  bytes  packets  errors  dropped  carrier  collsns
         145434    762      0         2         0         0
```

NOTE: This MAC address is valid only for the current boot cycle. On rebooting the device, the MAC address is updated with the address from persistent storage.

- When the device is connected to a public network, the DHCP server connected to the network assigns an IP address automatically. If a dynamic IP address isn't assigned to the device but Ethernet connectivity is required, configure a static IP address manually within the same subnet of the client.

```
ip addr add <ip>/<prefix> dev <Interface_name>
```

For example:

```
ip addr add 169.254.227.235/16 dev enu1
```

Sample output:

```
ip -s addr show dev enP1p5s0f1
enu1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
default qlen 1000
    link/ether 62:99:60:23:4d:52 brd ff:ff:ff:ff:ff:ff
    RX:  bytes  packets  errors  dropped  missed  mcast
         105971    313      0        2        0        0
    TX:  bytes  packets  errors  dropped  carrier  collsns
         145434    762      0        2        0        0
```

NOTE: This IP address is valid only for the current boot cycle. On rebooting the device, the IP address isn't retained.

- Configure the MTU size for the data packets over the Ethernet interface.

```
ip link set dev <Interface_name> down
ip link set dev <Interface_name> mtu <mtu_size>
ip link set dev <Interface_name> up
```

For example:

```
ip link set dev enu1 down
ip link set dev enu1 mtu 1500
ip link set dev enu1 up
```

Sample output:

```
ip -s addr show dev enu1
enu1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group
default qlen 1000
    link/ether 62:99:60:23:4d:52 brd ff:ff:ff:ff:ff:ff
    RX:  bytes  packets  errors  dropped  missed  mcast
         105971    313      0        2        0        0
    TX:  bytes  packets  errors  dropped  carrier  collsns
         145434    762      0        2        0        0
```

8 Set up the DisplayPort

To view the output of sample applications on your DisplayPort, set up the DisplayPort. See the appropriate chip product for information about your device.

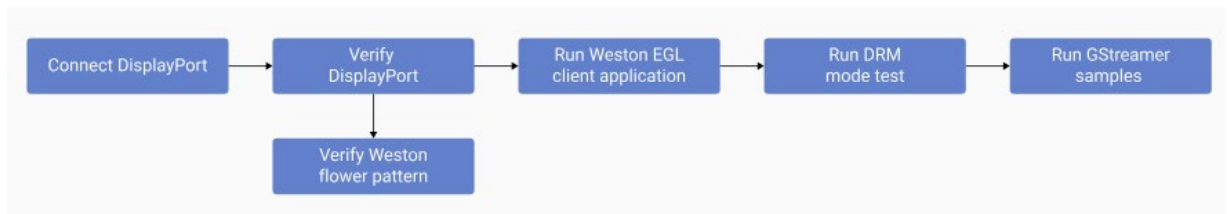


Figure 8-1 DisplayPort setup workflow

The display subsystem provides the Weston and Wayland compositors, along with `libdrm`, which interact with the DPU driver to render the Wayland client applications. Use the sample test applications to evaluate the basic rendering of the Wayland applications and to connect to the DisplayPort.

Verify the DisplayPort setup

1. Connect the DisplayPort cable to Embedded DisplayPort (eDP). See the image in [section 2](#) to locate the edp port on IQX
2. Power on the device. Weston launches automatically during boot.
3. Verify the display.

The following Weston flower pattern displays on both DisplayPort screens, indicating successful setup



NOTE: After you have set up the DisplayPort, run the following display test applications to verify the display functionality.

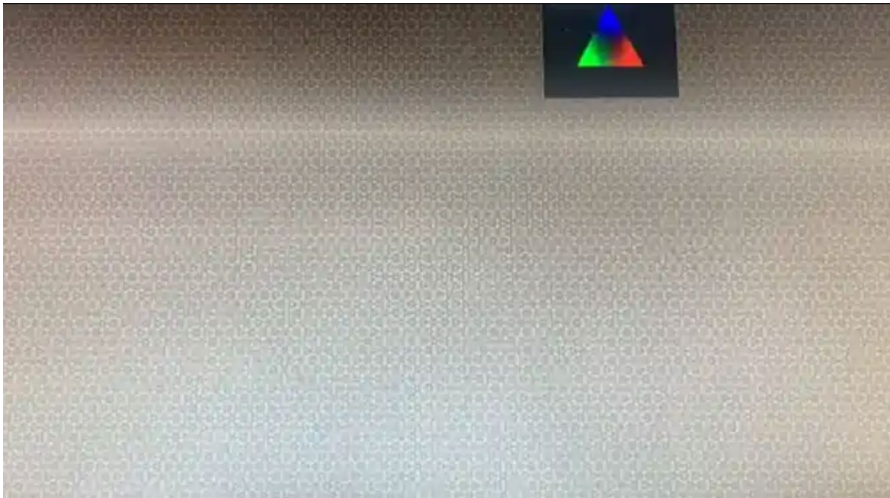
Run the Weston simple EGL client application

Weston simple EGL is a sample test application used to verify the composition and rendering of frames on your DisplayPort. To run the application, run the following commands:

NOTE: You must enable SSH to access your host device.

```
mount -o remount,rw /
su
. /etc/profile
export XDG_RUNTIME_DIR=/run/user/1000 && mkdir -p $XDG_RUNTIME_DIR
cd /usr/bin
export WAYLAND_DISPLAY=wayland-1 && weston-simple-egl
```

The following figure shows the Weston simple EGL client application running on the display monitor.



NOTE: To kill the application, select CTRL + C.

Run the DRM modetest application

DRM modetest ensures that the DSI and DPU paths are working effectively. To verify the display driver topology, run the DRM modetest.

NOTE: You must enable SSH to access your host device.

1. To kill the Weston client application, run the following commands in the device shell:

```
mount -o remount,rw /
ps -A | grep Weston
```

Sample output:

```
824 ?          00:00:12 weston
1090 ?         00:00:00 weston-keyboard
1092 ?         00:00:01 weston-desktop-
```

Use the Weston process ID from the sample Weston process list to kill the Weston processes.

2. To kill all Weston processes, run the following command:
`systemctl stop weston.service`
3. To change permissions of the modetest application, run the following command:
`chmod 0777 /usr/bin/modetest`
4. To view the connector IDs and modes, run the following command:
`/usr/bin/modetest -M msm > /opt/drm_mode.txt`
5. To verify DRM modes and connector information, pull the `/opt/drm_mode.txt` file locally on your host computer. Use the Secure Copy Protocol (SCP) to pull the `drm_mode.txt` file from the device to your host computer. For example,
`scp -r root@<IP of the device>:/opt/drm_mode.txt <directory path of the host machine>`

NOTE: When prompted for a password, enter `oelinux123`.

The output as shown in the following figure is from the `/opt/drm_mode.txt` file.

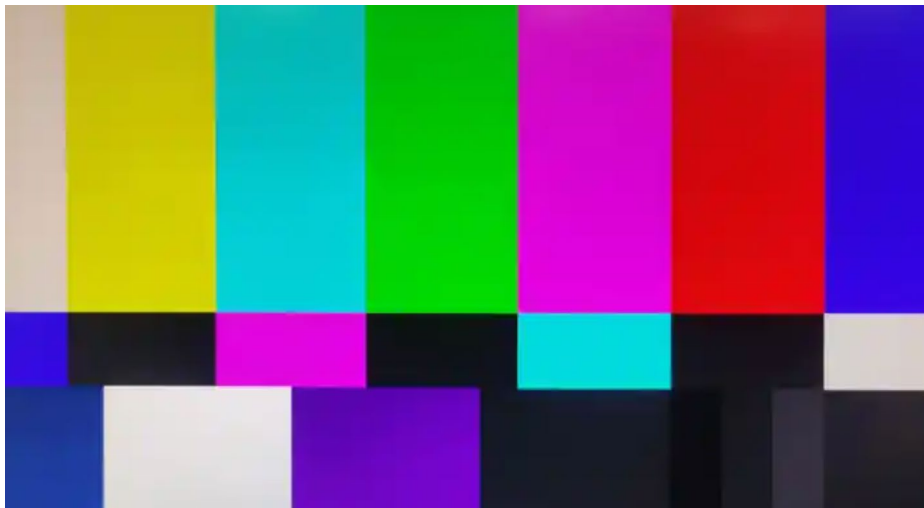
```

/usr/bin# ./modetest -M msm
Connectors:
id      encoder status      name      size (mm)      modes      encoders
39      38      connected      DP-2      600x340      13      38
modes:
index name refresh (Hz) hdisp hss hse htot vdisp vss vse vtot
#0 2560x1440 59.95 2560 2608 2640 2720 1440 1443 1448 1481 241500 flags: phsync, nvsyn

```

6. To fetch the connector ID and mode name, use the relevant values from the `/opt/drm_mode.txt` file:
`modetest -M msm -s <Connector id>:<mode name>`
7. To start the modetest application, run the following sample command using the connector ID and the mode name retrieved from the `/opt/drm_mode.txt` file. For example:
`/usr/bin/modetest -M msm -s 32:1920x1080-60`

NOTE: Start the DRM modetest application and ignore the warning log:
Failed to set gamma: Function not implemented.



8. To kill the modetest application, select CTRL + C.

NOTE: Ensure that you relaunch weston.

9 FAQs

9.1 How do you perform configuration data table (CDT) flashing?

CDT is a set of platform-specific configuration files used during flashing to configure hardware parameters on the Dragonwing IQ-X5xxx EVK.

To perform CDT flashing, do the following:

1. Download the IQ-X 1.4-EVK-CDT.tar.gz files from [JFrog](#).
2. Unzip the boot binaries.
3. Put the device in [EDL mode](#).
4. Verify whether the Dragonwing IQ-X5xxx EVK has entered the EDL mode by running the following command on the host computer:

```
$ lsusb
```

Sample output

```
Bus 002 Device 014: ID 05c6:9008 Qualcomm, Inc. Gobi Wireless Modem (QDL mode)
```

5. Navigate to CDTworkspace.

```
$ cd IQ-x.1.4-EVK-CDT
```

6. Wipeout CDT:

```
$ <qdl_tool_path>/qdl --storage spinor xbl_s_devprg_ns.melf  
rawprogram0_WIPE_PARTITIONS.xml  
$ <qdl_tool_path>/qdl --storage spinor xbl_s_devprg_ns.melf  
rawprogram0_BLANK_GPT.xml
```

7. Flash CDT

```
$ <qdl_tool_path>/qdl --storage spinor xbl_s_devprg_ns.melf  
rawprogram0.xml patch0.xml
```

9.2 How do you provision universal flash storage (UFS)?

UFS provisioning helps divide the storage into several logical unit numbers (LUNs), allowing you to store different types of data separately. This improves access efficiency and system organization.

To provision the UFS, do the following:

1. Download the provisioning.zip file from [JFrog](#) and unzip it.
2. Put the device in [EDL mode](#).
3. Navigate to the provisioning directory.

```
$ cd provisioning
```

4. Provision UFS:

```
$ <qdl_tool_path>/qdl --storage ufs xbl_s_devprg_ns.melf  
provision_default.xml
```

9.3 Further support

Post your questions on the [Qualcomm support forum](#).

A References

A.1 Related documents

Title	Number
Qualcomm Technologies, Inc.	
<i>IQ-X Series Module Data Sheet</i>	80-77181-1
<i>IQ-X Series Module Pin Assignment and GPIO Configuration Specification Spreadsheet</i>	80-77181-1A
<i>IQ-X Series Module Technical Reference Manual</i>	80-77181-5

A.2 Acronyms and terms

Acronym or term	Definition
CDT	Configuration data table
DIP	Dual in-line package
EDL	Emergency Download
EVK	Evaluation kit
NVMe	Non-Volatile Memory Express
QDL	Qualcomm Downloader
UART	Universal asynchronous receiver/transmitter
UEFI	Unified extensible firmware interface
UFS	Universal flash storage

LEGAL INFORMATION

Your access to and use of this material, along with any documents, software, specifications, reference board files, drawings, diagnostics and other information contained herein (collectively this "Material"), is subject to your (including the corporation or other legal entity you represent, collectively "You" or "Your") acceptance of the terms and conditions ("Terms of Use") set forth below. If You do not agree to these Terms of Use, you may not use this Material and shall immediately destroy any copy thereof.

1) Legal Notice.

This Material is being made available to You solely for Your internal use with those products and service offerings of Qualcomm Technologies, Inc. ("Qualcomm Technologies"), its affiliates and/or licensors described in this Material, and shall not be used for any other purposes. If this Material is marked as "Qualcomm Internal Use Only", no license is granted to You herein, and You must immediately (a) destroy or return this Material to Qualcomm Technologies, and (b) report Your receipt of this Material to qualcomm.support@qti.qualcomm.com. This Material may not be altered, edited, or modified in any way without Qualcomm Technologies' prior written approval, nor may it be used for any machine learning or artificial intelligence development purpose which results, whether directly or indirectly, in the creation or development of an automated device, program, tool, algorithm, process, methodology, product and/or other output. Unauthorized use or disclosure of this Material or the information contained herein is strictly prohibited, and You agree to indemnify Qualcomm Technologies, its affiliates and licensors for any damages or losses suffered by Qualcomm Technologies, its affiliates and/or licensors for any such unauthorized uses or disclosures of this Material, in whole or part.

Qualcomm Technologies, its affiliates and/or licensors retain all rights and ownership in and to this Material. No license to any trademark, patent, copyright, mask work protection right or any other intellectual property right is either granted or implied by this Material or any information disclosed herein, including, but not limited to, any license to make, use, import or sell any product, service or technology offering embodying any of the information in this Material.

THIS MATERIAL IS BEING PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, WHETHER EXPRESSED, IMPLIED, STATUTORY OR OTHERWISE. TO THE MAXIMUM EXTENT PERMITTED BY LAW, QUALCOMM TECHNOLOGIES, ITS AFFILIATES AND/OR LICENSORS SPECIFICALLY DISCLAIM ALL WARRANTIES OF TITLE, MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR A PARTICULAR PURPOSE, SATISFACTORY QUALITY, COMPLETENESS OR ACCURACY, AND ALL WARRANTIES ARISING OUT OF TRADE USAGE OR OUT OF A COURSE OF DEALING OR COURSE OF PERFORMANCE. MOREOVER, NEITHER QUALCOMM TECHNOLOGIES, NOR ANY OF ITS AFFILIATES AND/OR LICENSORS, SHALL BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY EXPENSES, LOSSES, USE, OR ACTIONS HOWSOEVER INCURRED OR UNDERTAKEN BY YOU IN RELIANCE ON THIS MATERIAL.

Certain product kits, tools and other items referenced in this Material may require You to accept additional terms and conditions before accessing or using those items.

Technical data specified in this Material may be subject to U.S. and other applicable export control laws. Transmission contrary to U.S. and any other applicable law is strictly prohibited.

Nothing in this Material is an offer to sell any of the components or devices referenced herein.

This Material is subject to change without further notification.

In the event of a conflict between these Terms of Use and the Website Terms of Use on www.qualcomm.com, the *Qualcomm Privacy Policy* referenced on www.qualcomm.com, or other legal statements or notices found on prior pages of the Material, these Terms of Use will control. In the event of a conflict between these Terms of Use and any other agreement (written or click-through, including, without limitation any non-disclosure agreement) executed by You and Qualcomm Technologies or a Qualcomm Technologies affiliate and/or licensor with respect to Your access to and use of this Material, the other agreement will control.

These Terms of Use shall be governed by and construed and enforced in accordance with the laws of the State of California, excluding the U.N. Convention on International Sale of Goods, without regard to conflict of laws principles. Any dispute, claim or controversy arising out of or relating to these Terms of Use, or the breach or validity hereof, shall be adjudicated only by a court of competent jurisdiction in the county of San Diego, State of California, and You hereby consent to the personal jurisdiction of such courts for that purpose.

2) Trademark and Product Attribution Statements.

Qualcomm is a trademark or registered trademark of Qualcomm Incorporated. Arm is a registered trademark of Arm Limited (or its subsidiaries) in the U.S. and/or elsewhere. The Bluetooth® word mark is a registered trademark owned by Bluetooth SIG, Inc. Other product and brand names referenced in this Material may be trademarks or registered trademarks of their respective owners.

Snapdragon and Qualcomm branded products referenced in this Material are products of Qualcomm Technologies, Inc. and/or its subsidiaries. Qualcomm patented technologies are licensed by Qualcomm Incorporated.

THE DOCUMENTATION ACCOMPANYING THE MATERIALS AND/OR RELEVANT PRODUCTS AND SERVICE OFFERINGS MAY INCLUDE IMPORTANT USE LIMITATIONS. ANY DEVIATIONS FROM APPLICABLE USE LIMITATIONS MAY ADVERSELY IMPACT PERFORMANCE, DURABILITY, QUALITY OR SAFETY. YOU ASSUME ALL RISKS AND LIABILITIES ASSOCIATED WITH ANY DEVIATIONS.