Qualcomm Technologies, Inc.

Qualcomm Linux Display Guide

80-70018-18 AF

May 9, 2025

Contents

1 Display overview

The Qualcomm® Adreno™ Display Processing Unit (DPU) provides hardware-accelerated image processing. The DPU retrieves pixel data from the memory and sends it to the display peripherals through standard interfaces.

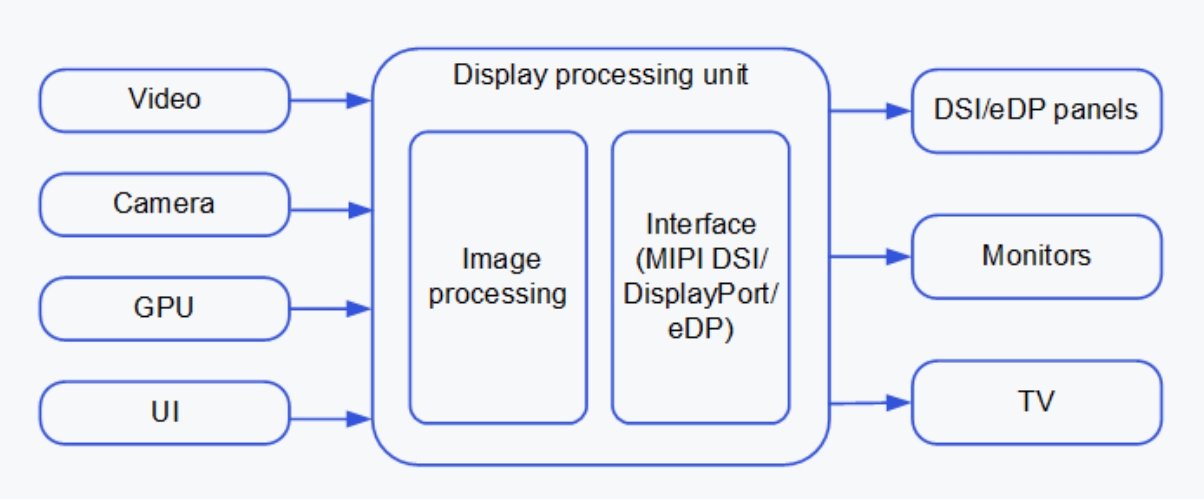

Figure : DPU architecture

The DPU provides the following image processing and interface capabilities:

- **Image processing** includes the following:
  - **Composition**: blends individual layers into a single frame and sends it to the display panel
  - **Scaling**: supports upscale and downscale of the source image to match the panel resolution
  - **Color format conversion**: converts colors from one color format to another (from YUV to RGB)
  - **Color space conversion**: converts colors from one color space to another (sRGB, P3, and BT2020)
  - **Color processing**: enhances the colors for better visual quality
- **Interfaces**: Connects with various panels using mobile industry processor interface display serial interface (MIPI DSI) and TV or Monitor with the DisplayPort interface

- **MIPI DSI**: defines the protocols between a host processor and peripheral devices that adhere to the MIPI alliance specifications for mobile device interfaces
- **DisplayPort**: serves as a digital display interface, primarily connecting video sources to display devices such as computer monitors, and can also transmit audio, USB, and other data forms

The Qualcomm® Intelligent Multimedia SDK (IM SDK) and the `libdrm` APIs expose the hardware capabilities of the Adreno DPU engines.

Note: See Hardware SoCs that are supported on Qualcomm® Linux®.

1.1 Display architecture

This information explains the primary components within the hardware and the software architecture of the Qualcomm Linux display subsystem.

Display hardware architecture

The hardware architecture includes components for source and destination processing, layer mixing, compression, and display interfaces.

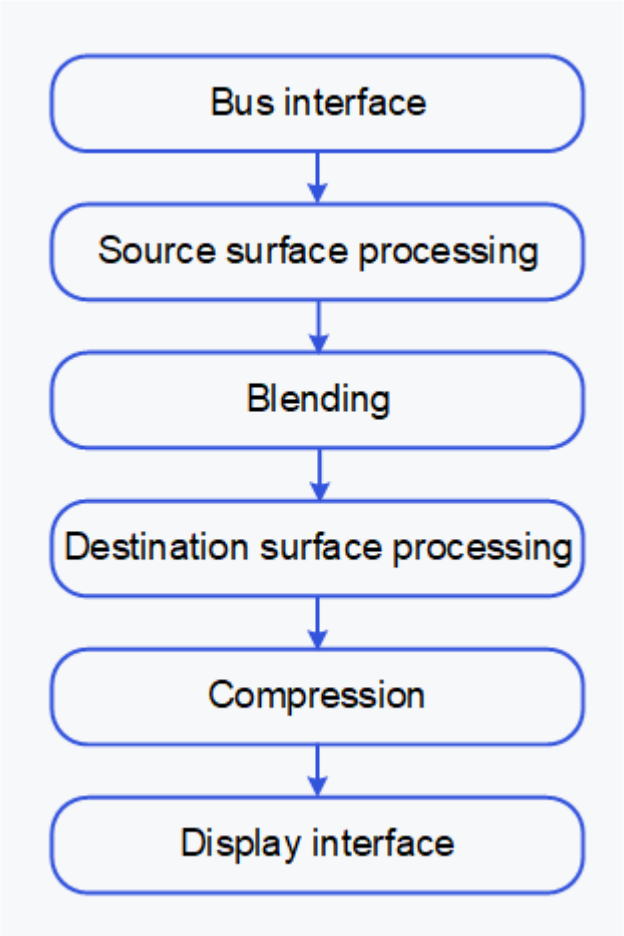

Figure : DPU hardware architecture block diagram

The following table lists the components of the display software architecture.

Table : Display hardware architecture components

Component	Description
Bus interface	Fetches data from memory for processing

Component	Description
Source surface processing	• Reads RGB and YUV surfaces from games and video applications • Performs format conversion and quality improvements for the source
Blending	Blends and mixes the source surface together
Destination surface processing	Converts, corrects, and adjusts the data based on panel characteristics
Compression	Reduces bandwidth and power consumption by sending compressed display buffers to the display
Display interface	Generates timings for the connected display peripherals

Detailed information on the display hardware architecture is available to licensed developers with authorized access. For more information, see Qualcomm Linux Display Guide - Addendum.

Display software architecture

The display subsystem consists of Wayland, a display server protocol, and its compositors, such as Weston. Wayland uses a client/server model where the client application interacts with the Wayland server.

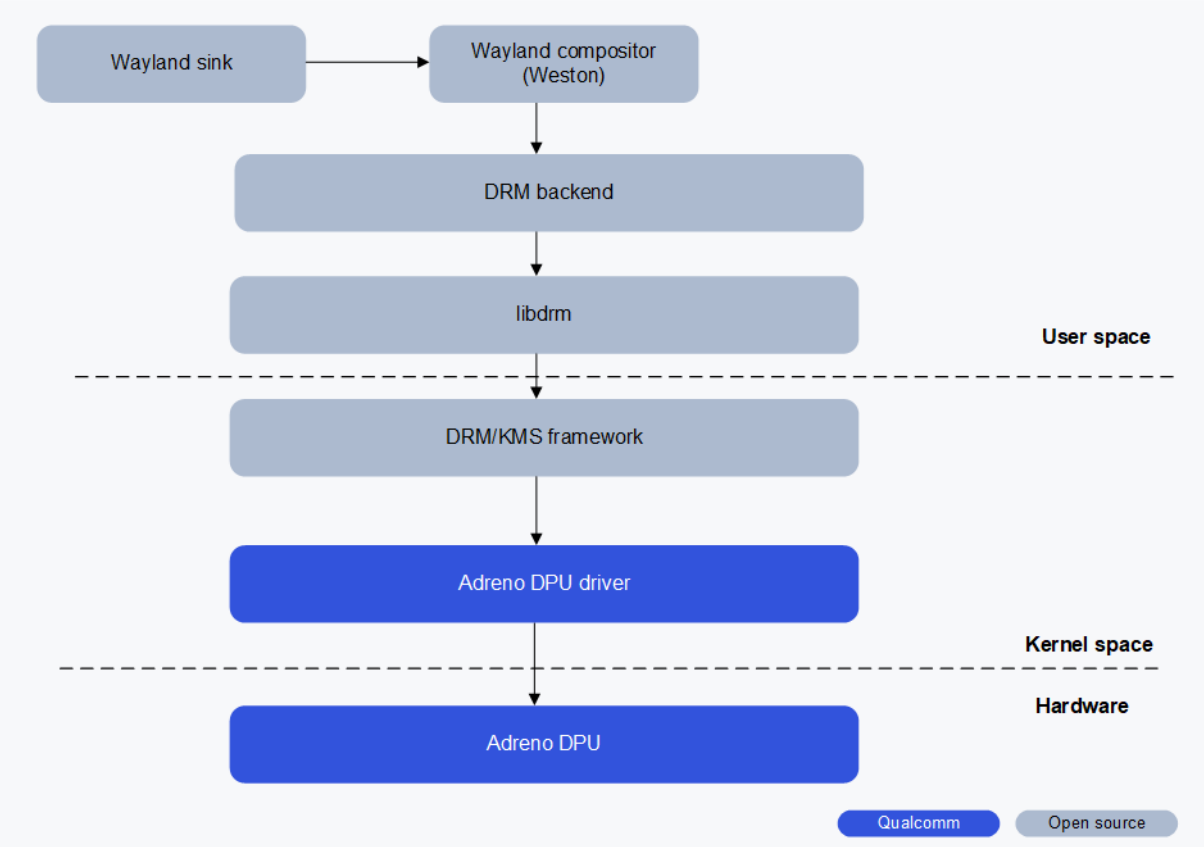

Figure : Display software architecture block diagram

The components of the display software architecture are as follows:

Component	Description
Wayland sink	The Wayland sink plug-in communicates with the Weston subsystem to render the display.
Wayland compositor (Weston)	The Wayland compositor uses the Wayland protocol as a display server. The compositor handles the composition and rendering requirements at the system level and runs as a separate process in the system. Weston serves as the reference implementation of a Wayland compositor.

Component	Description
Direct rendering manager (DRM) backend	The DRM backend software framework manages the Qualcomm Adreno DPU hardware resources. It selects the optimal composition strategy for each connected display and allocates display processor hardware resources for the layer stack received from the client.
`libdrm`	The `libdrm` library provides APIs for accessing the DRM I/O control (IOCTL).
DRM/KMS framework	The DRM/KMS framework offers kernel and userland level interface, with the help of `libdrm` libraries to access the related hardware features, configuration, and hardware acceleration.
Adreno DPU driver	The Adreno DPU driver manages all the pixel data paths to different panel interfaces.
Adreno DPU	The Adreno DPU is a hardware-accelerated engine that performs image processing and interfacing with minimal power consumption.

1.2 Software APIs

Qualcomm Linux provides the following display APIs:

- Weston API: facilitates interactions with the DRM compositor. For the Weston APIs, see Wayland client APIs and Wayland server APIs.
- `libdrm` API: facilitates interactions with the DRM/KMS framework. For the APIs, see libdrm.

Qualcomm provides you the GStreamer-based plugins. You can add the Waylandsink plugin to your media pipeline from IM SDK plugins.

For more information, see GStreamer application development and pipeline creation.

2 Enable display

Before you begin, set up your infrastructure as described in the Qualcomm Linux Build Guide.

2.1 Set up HDMI display

See the appropriate chip-product for device-specific information.

QCS6490

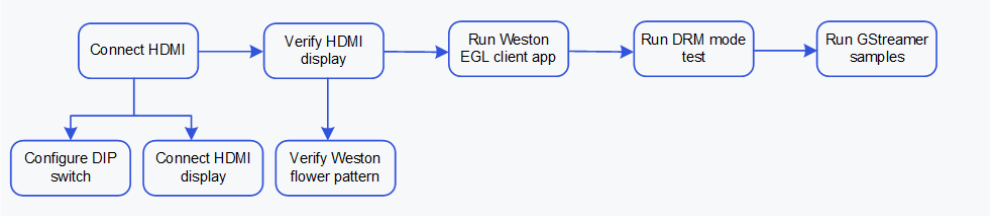

Figure : HDMI setup workflow

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

Prerequisites

Ensure that you configure the following settings on the device:

1. Set Switch 4 of DIP_SW_0 to OFF.
2. Connect the device to the HDMI display.

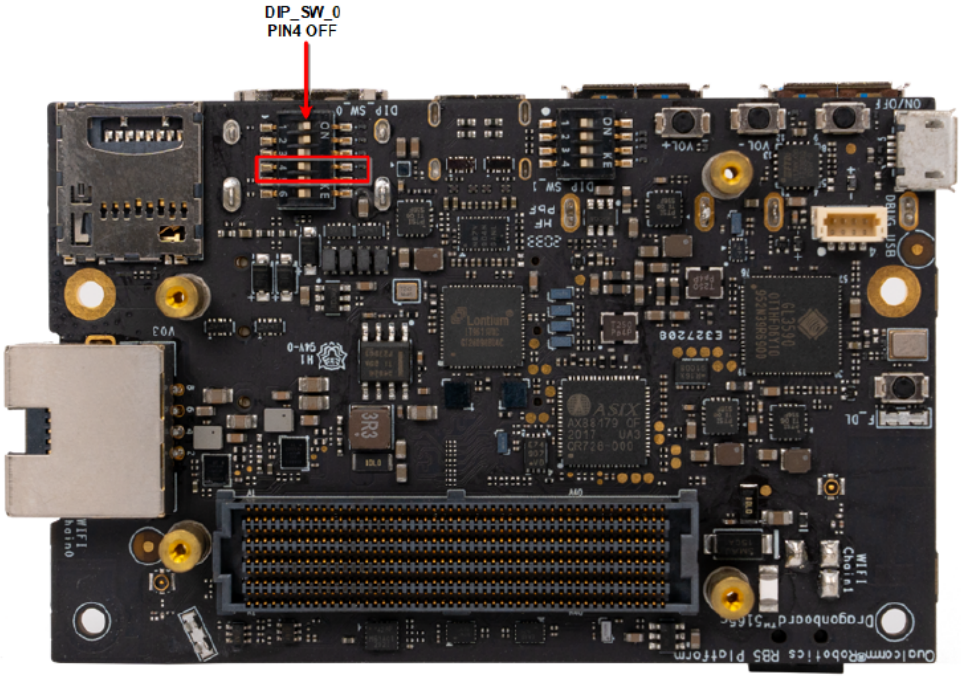

Figure : DIP switch selection for HDMI

Verify HDMI display setup

To verify the HDMI display setup, power on the device and connect to the HDMI cable. Weston starts automatically when the device boots up. You should see the following Weston flower pattern on the display monitor.

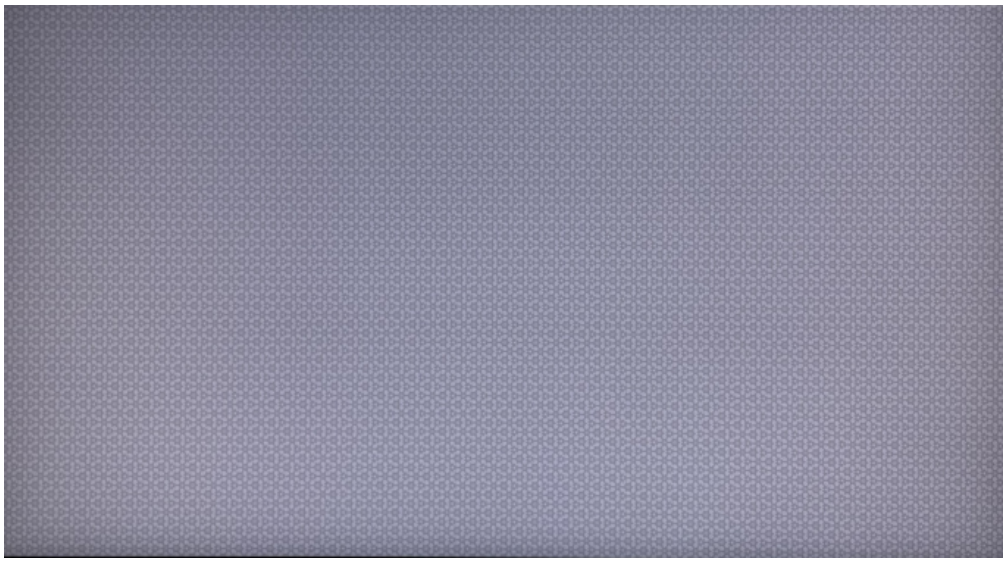

Figure : Weston flower pattern

Note:

- If you are experiencing a no-display screen issue on HDMI, see Debug HDMI display.
- If you connect the HDMI cable after powering ON the device, see Plug in the HDMI/eDP cable after the device is powered on.

Run the Weston simple EGL client application

To run the Weston simple EGL client application, run the following commands:

Note: You must enable SSH to access your host device. For instructions, see Sign in using SSH.

```
mount -o remount,rw /
```

```
su
```

```
. /etc/profile
```

```
export XDG_RUNTIME_DIR=/dev/socket/weston && 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.

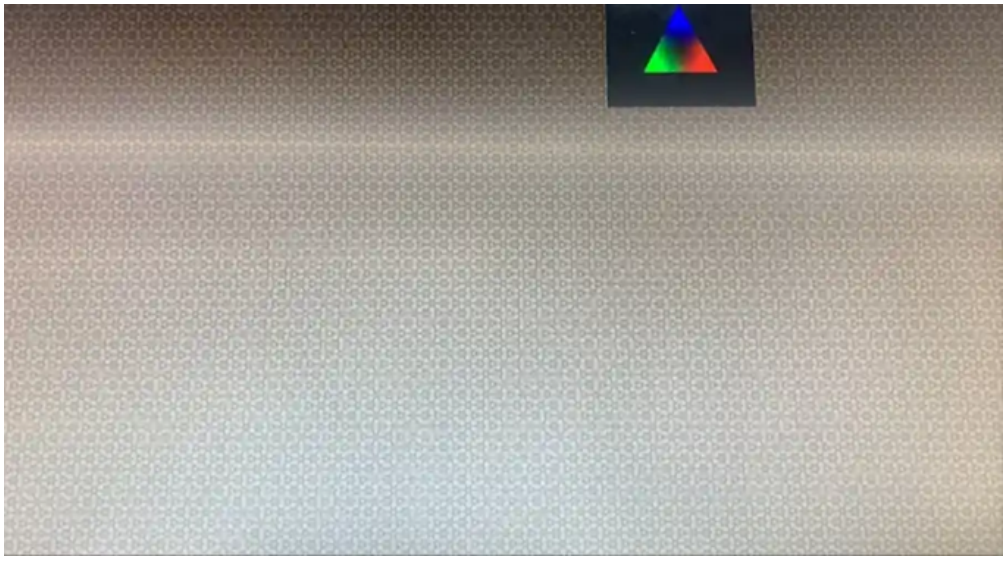

Figure : Weston simple EGL client application

Note: To kill the application, select **CTRL + C**.

Run DRM mode test

Note: To use the DRM mode test application, you must Generate the DRM mode test binary.

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

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:

   ```
   killall weston
   ```

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 -c msm_drm > /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.

```
sh-5.1# /usr/bin/modetest -M msm_drm
Encoders:
id      crtc    type    possible crtcs  possible clones
31      0       DSI     0x0000000f      0x00000001
55      0       TMDS    0x0000000f      0x00000002

Connectors:
id      encoder status          name            size (mm)       modes   encoders
32      0       connected       DSI-1           620x340         10      31
  modes:
        index name refresh (Hz) hdisp hss hse htot vdisp vss vse vtot
  #0 1920x1080 60.00 1920 2008 2052 2200 1080 1084 1089 1125 148500 flags: nhsync, nvsync; type: preferred, driver
```

Figure : Command prompt output

6. To fetch the connector ID and mode name, use the relevant values from the `/opt/drm_mode.txt` file:

   ```
   modetest -c msm_drm -s <Connector id>:<mode
   name>
   ```

7. To launch the modetest application, run the following sample command using the connector ID and mode name retrieved from the `/opt/drm_mode.txt` file. For example:

   ```
   /usr/bin/modetest -c msm_drm -s 32:1920x1080-60
   ```

   Note: On launching the DRM test application, ignore the warning log: `Failed to set gamma: Function not implemented`.

Figure : DRM modetest sample application output screen

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

Note: Ensure that you Relaunch Weston.

Run GStreamer video playback applications

The GStreamer sample applications for video playback are part of the Qualcomm IM SDK. For more information about prerequisites for GStreamer applications, see Sample applications.

To run the sample video test cases using `gst-launch-1-0`, see With video playback as an example.

Note: Before running the test cases, ensure that you meet the prerequisites.

QCS9075

Not applicable

QCS8275

Not applicable.

2.2 Set up DisplayPort

See the appropriate chip-product for information about your device.

QCS6490

Not applicable

QCS9075

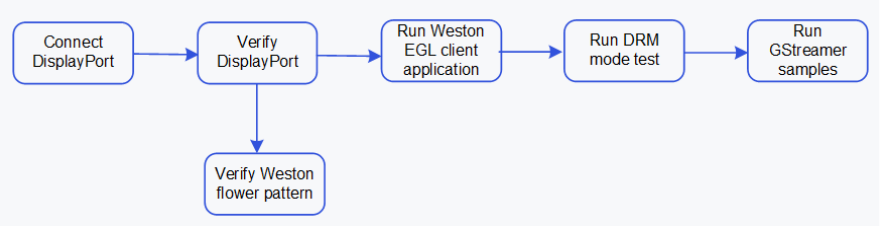

Figure : DisplayPort setup workflow

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

Prerequisites

Ensure that you configure the following settings on the device as shown in the figure.

- Connect the DisplayPort cable to the eDP0 connector.
- Ensure DIP switch 1 and 2 are ON.

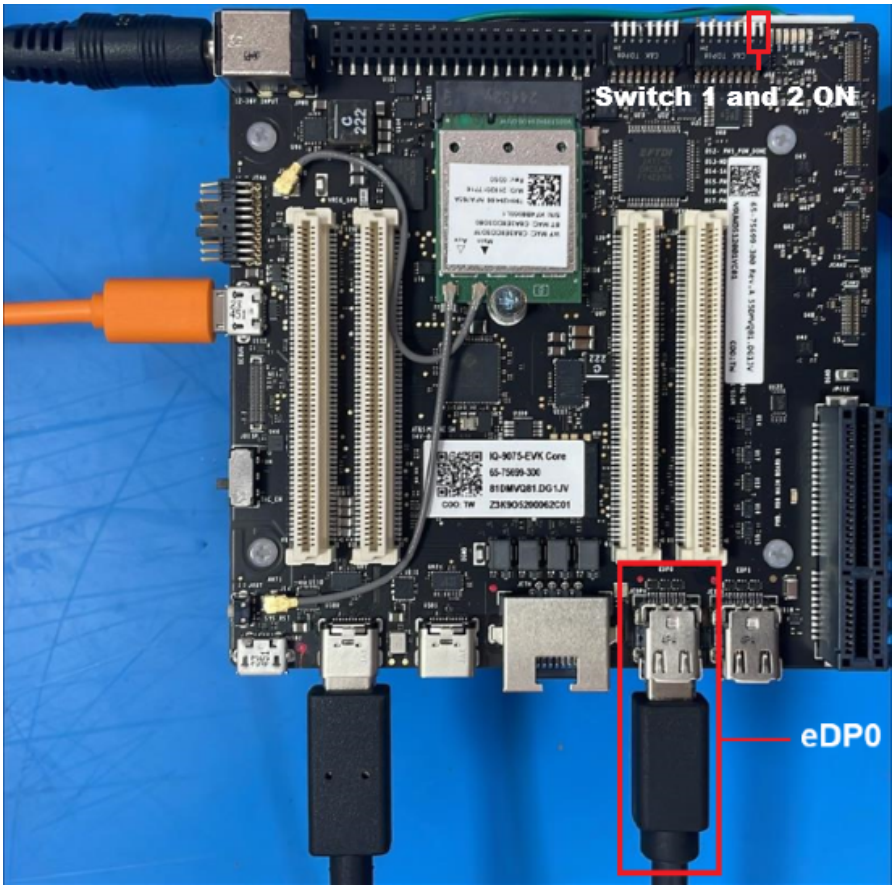

Figure : DisplayPort setup

Verify the DisplayPort setup

To verify the DisplayPort setup, power ON the device. Weston starts automatically when the device boots up. You should see the following Weston flower pattern on the display monitor.

Figure : Weston flower pattern

Note: If you connect the eDP cable after powering ON the device, see Plug in the HDMI/eDP cable after the device is powered on.

Run the Weston simple EGL client application

To run the Weston simple EGL client application, run the following commands:

Note: You must enable SSH to access your host device. For instructions, see Sign in using SSH.

```
mount -o remount,rw /
```

```
su
```

```
. /etc/profile
```

```
export XDG_RUNTIME_DIR=/dev/socket/weston && 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.

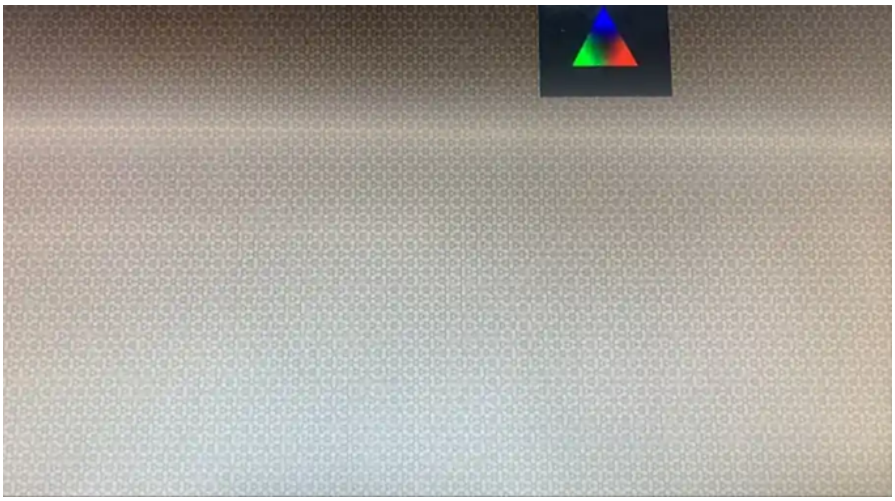

Figure : Weston simple EGL client application

Note: To kill the application, select **CTRL + C**.

Run DRM mode test

Note: To use the DRM mode test application, you must Generate the DRM mode test binary file.

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

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

   Note: You must enable SSH to access your host device. For instructions, see Sign in using SSH.

   ```
   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:

   ```
   killall weston
   ```

3. To change permissions of the modetest application, run the following command:

   ```
   chmod 0777 /usr/bin/modetest
   ```

4. To know the connector IDs and modes, run the following command:

```
/usr/bin/modetest -c msm_drm > /opt/drm_
mode.txt
```

5. To check for 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.

```
sh-5.1# /usr/bin/modetest -M msm_drm
Encoders:
id      crtc    type    possible crtcs  possible clones
31      0       DSI     0x0000000f      0x00000001
55      0       TMDS    0x0000000f      0x00000002

Connectors:
id      encoder status          name            size (mm)       modes   encoders
32      0       connected       DSI-1           620x340         10      31
  modes:
        index name refresh (Hz) hdisp hss hse htot vdisp vss vse vtot
  #0 1920x1080 60.00 1920 2008 2052 2200 1080 1084 1089 1125 148500 flags: nhsync, nvsync; type: preferred, driver
```

Figure : Command prompt output

6. To fetch the connector ID and mode name, use the relevant values from the `/opt/drm_mode.txt` file:

```
modetest -c msm_drm -s <Connector id>:<mode
name>
```

7. To launch the modetest application, run the following sample command using the connector ID and mode name retrieved from the `/opt/drm_mode.txt` file. For example:

```
/usr/bin/modetest -c msm_drm -s 32:
1920x1080-60
```

Note: On launching the DRM test application, ignore the warning log: `Failed to set gamma: Function not implemented`.

Figure : DRM modetest sample application output screen

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

Note: Ensure that you Relaunch Weston.

Run GStreamer video playback applications

The GStreamer sample applications for video playback are part of the Qualcomm IM SDK. For more information about prerequisites for GStreamer applications, see Sample applications.

To run the sample video test cases using `gst-launch-1-0`, see With video playback as an example.

Note: Before running the test cases, ensure that you meet the prerequisites.

QCS8275

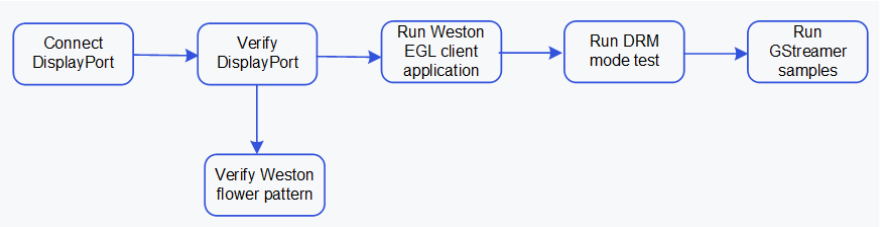

Figure : DisplayPort setup workflow

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

Connect the DisplayPort cable to the eDP1 connector as shown in the following figure.

Figure : QCS8275 DisplayPort setup

Verify DisplayPort setup

To verify the DisplayPort setup, power ON the device. Weston starts automatically when the device boots up. You should see the following Weston flower pattern on the display monitor.

Figure : Weston flower pattern

Note: If you connect the eDP cable after powering ON the device, see Plug in the HDMI/eDP cable after the device is powered on.

Run the Weston simple EGL client application

To run the Weston sample EGL client application, run the following commands:

Note: You must enable SSH to access your host device. For instructions, see Sign in using SSH.

```
mount -o remount,rw /
```

```
su
```

```
. /etc/profile
```

```
export XDG_RUNTIME_DIR=/dev/socket/weston && 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.

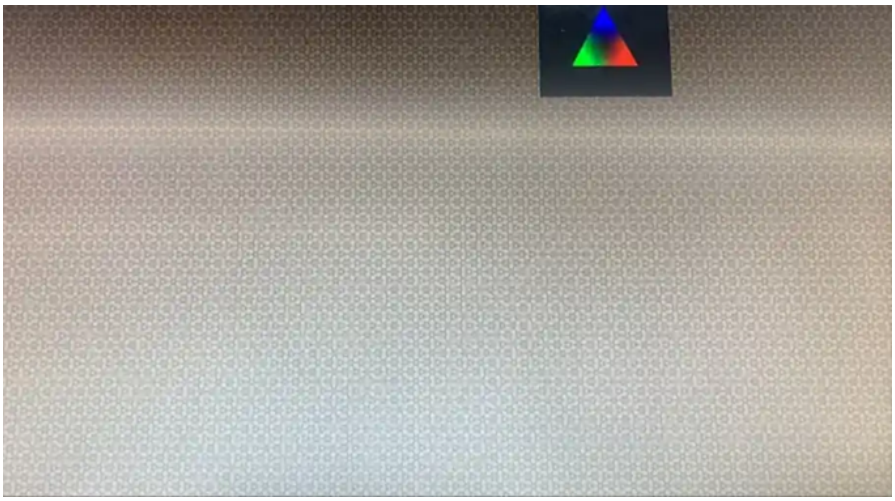

Figure : Weston simple EGL client application

Note: To kill the application, select **CTRL + C**.

Run DRM mode test

Note: To use the DRM mode test application, you must Generate DRM mode test binary file.

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

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

   Note: You must enable SSH to access your host device. For instructions, see Sign in using SSH.

   ```
   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:

   ```
   killall weston
   ```

3. To change permissions of the modetest application, run the following command:

   ```
   chmod 0777 /usr/bin/modetest
   ```

4. To know the connector IDs and modes, run the following command:

```
/usr/bin/modetest -c msm_drm > /opt/drm_
mode.txt
```

5. To check for 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.

```
sh-5.1# /usr/bin/modetest -M msm_drm
Encoders:
id      crtc    type    possible crtcs  possible clones
31      0       DSI     0x0000000f      0x00000001
55      0       TMDS    0x0000000f      0x00000002

Connectors:
id      encoder status          name            size (mm)       modes   encoders
32      0       connected       DSI-1           620x340         10      31
  modes:
        index name refresh (Hz) hdisp hss hse htot vdisp vss vse vtot
  #0 1920x1080 60.00 1920 2008 2052 2200 1080 1084 1089 1125 148500 flags: nhsync, nvsync; type: preferred, driver
```

Figure : Command prompt output

6. To fetch the connector ID and mode name, use the relevant values from the `/opt/drm_mode.txt` file:

```
modetest -c msm_drm -s <Connector id>:<mode
name>
```

7. To launch the modetest application, run the following sample command using the connector ID and mode name retrieved from the `/opt/drm_mode.txt` file. For example:

```
/usr/bin/modetest -c msm_drm -s 32:
1920x1080-60
```

Note: On launching the DRM test application, ignore the warning log: `Failed to set gamma: Function not implemented`.

Figure : DRM modetest sample application output screen

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

Note: Ensure that you Relaunch Weston.

Run GStreamer video playback applications

The GStreamer sample applications for video playback are part of the Qualcomm IM SDK. For more information on prerequisites for GStreamer applications, see Sample applications.

To run the sample video test cases using `gst-launch-1-0`, see With video playback as an example.

Note: Before running the test cases, ensure that you meet the prerequisites.

3 Display specifications

The display subsystem is a software component that allows high-quality graphics and video output on devices. The following sections explain the key features of the display subsystem along with the capabilities, and advanced features.

See the appropriate chip-product for device-specific information.

QCS6490

Key features

The display subsystem supports the following key features:

- Wayland and Weston with upstream aligned protocol
- Weston with direct rendering manager (DRM) backend module
- DSI 1.2
- MIPI DSI D-PHY
- Unified extensible firmware interface (UEFI)

Capabilities

The device supports two displays simultaneously and offers various configurations for maximum concurrency. The following table summarizes the capabilities.

Table : Capabilities and configurations

DSI 0	• FHD+ (1200 × 2520) at 120 fps (8-bit) • Supports Video Electronics Standards Association (VESA) display stream compression (DSC) 1.2
DSI 0 to HDMI	• 1920 × 1080p at 60 fps

DisplayPort	• 3840 × 2160 resolution at 30 fps, 24 bpp (requires two lanes at HBR3) • Single stream transport • Concurrent functionality of DisplayPort and USB 3.0 **Note:** To enable DisplayPort, see the build instructions and the configuration changes in Qualcomm Linux Display Guide - Addendum.
eDP	• A maximum resolution of 1920 × 1080p at 60 fps • Non-pluggable eDP panels • Concurrent functionality with the DisplayPort **Note:** To enable eDP, see the build instructions and configuration changes in Qualcomm Linux Display Guide - Addendum.
Maximum concurrency	• HD+ (1200 × 2520) (8-bit) at 60 fps with DSI primary + 4K at 30 fps DisplayPort Or • 1920 × 1080p at 60 fps with DSI to HDMI primary + 4K at 30 fps DisplayPort Or • 1920 × 1080p at 60 fps with eDP primary + 4K at 30 fps DisplayPort

Advanced features

The display subsystem enhances the visual performance and efficiency of the device. A key feature of the display subsystem is to provide improved picture quality with HDR10 support for internal display. This feature is available to licensed developers with authorized access.

QCS9075

Key features

The display subsystem supports the following key features:

- Wayland and Weston with upstream protocol
- Weston compositor with direct rendering manager (DRM) backend module
- 2xDisplayPort with single stream transport (SST) and multi-stream transport (MST)

Capabilities

The device supports two displays simultaneously and offers various configurations for maximum concurrency. The following table summarizes the capabilities.

DP0	• Supports 3840 × 2160 at 60 fps 30 bpp
DP1	• Supports 3840 × 2160 at 60 fps 30 bpp
Maximum concurrency	• Supports 3840 × 2160 at 60 fps 30 bpp with DP0 as Primary + 3840 × 2160 at 60 fps 30 bpp with DP1 as Secondary

QCS8275

Key features

The display subsystem supports the following key features:

- Wayland and Weston with upstream protocol
- Weston compositor with direct rendering manager (DRM) backend module
- 1xDisplayPort with single stream transport (SST) and multi-stream transport (MST)

Capabilities

DP1 supports 3840 × 2160 at 60 fps 30 bpp.

Note: Use EDP1 as the primary interface.

4 Troubleshoot display

This information provides the necessary details to troubleshoot and analyze various components, such as the DSI panel clock, DPU, and Weston.

See the appropriate chip-product for information pertaining to your device.

QCS6490

4.1 View DSI clock information

Note:

- You must enable SSH to access your host device. For instructions, see Sign in using SSH.
- To mount the `debugfs` file system to access the `debugfs` details of the display subsystem, run the following commands:

```
mount -o remount,rw /
```

```
mount -t debugfs none /sys/kernel/debug
```

To check the DSI clock information, run the following command:

```
cat /sys/kernel/debug/clk/disp_cc_mdss_pclk0_clk/clk_rate
```

4.2 Debug HDMI display

LT9611uxc is the DSI-to-HDMI bridge chip firmware. The recommended firmware version is `LT9611UXC_DSI_PortA_HDCP_Disable_V5.0.21.bin`. If you experience a no-display screen on the HDMI, upgrade the firmware.

Upgrade Lontium firmware

To upgrade the Lontium firmware, do the following:

1. Download the recommended LT9611uxc firmware version `LT9611UXC_DSI_PortA_HDCP_Disable_V5.0.21.bin` from http://www.lontiumsemi.com/download and go to the Qualcomm section.

   Note:

   - Enable SSH to access your host device. For more information, see Sign in using SSH.
   - When prompted for a password, enter oelinux123.

2. To upgrade the firmware, run the following Linux shell command:

   ```
   mount -o rw,remount /usr
   ```

   Note: Close SSH and copy the Lontium firmware using the following `scp` command:

   ```
   scp <LT firmware bin> root@<IP of the device>:/lib/firmware/lt9611uxc_fw.bin
   ```

   ```
   sync
   ```

   Note: Prior to running the following command, enable SSH again.

   ```
   echo 0 > /sys/bus/i2c/devices/0-002b/lt9611uxc_firmware
   ```

   Note: Wait for at least 3 minutes for the firmware upgrade to complete.

3. To retrieve the lt9611 logs, run the following command:

   ```
   dmesg | grep lt9611
   ```

The `Firmware updates successfully` log confirms that the firmware upgrade is successful. To ensure that the new firmware is in effect, hard reboot (unplug and plug in the power cable) the device.

4.3 Enable the XWayland Compositor

XWayland is a compatibility layer that allows X11 applications to run in a Wayland environment. It translates X11 protocol requests into Wayland-compatible operations.

Run the following commands to enable XWayland for the display subsystem:

Note: You must enable SSH to access your host device. For instructions, see Sign in using SSH.

```
killall weston
```

```
export XDG_RUNTIME_DIR=/dev/socket/weston && mkdir -p
$XDG_RUNTIME_DIR && weston --continue-without-input --
idle-time=0 --backend=drm-backend.so --xwayland
```

Upon running the commands, you should see the following Weston flower pattern on the display monitor.

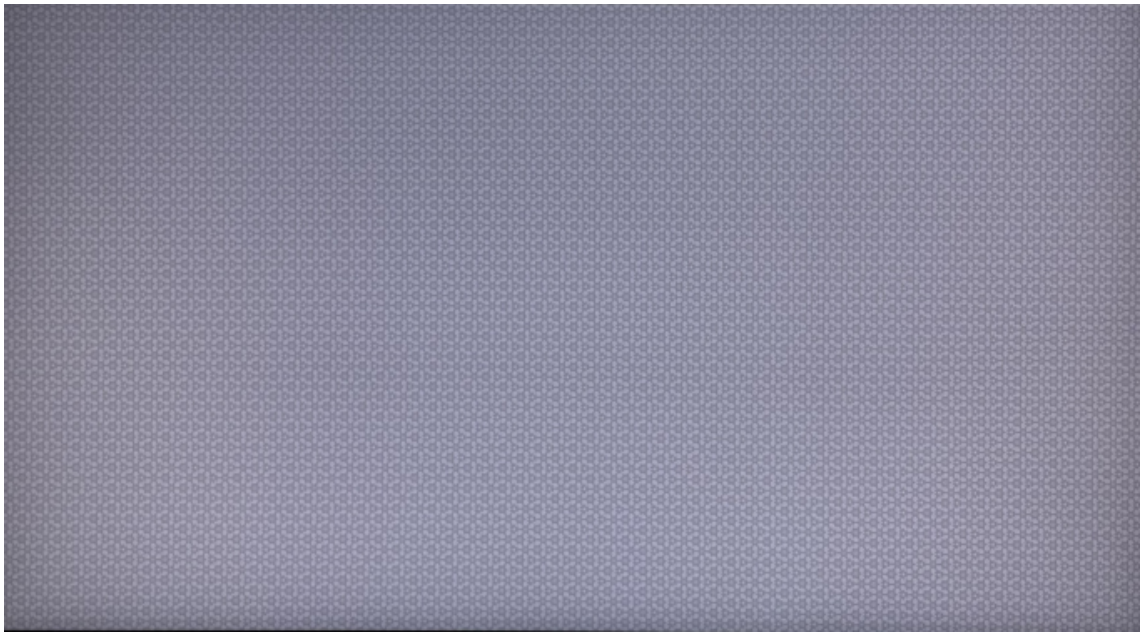

Figure : Weston flower pattern

QCS9075

4.4 Debug DisplayPort

Note:

- You must enable SSH to access your host device. For instructions, see Sign in using SSH.
- To mount the `debugfs` file system to access the `debugfs` details of the display subsystem, run the following commands:

```
mount -o remount,rw /
```

```
mount -t debugfs none /sys/kernel/debug
```

View DisplayPort information

To check the selected DisplayPort, run the following command:

```
cat /sys/kernel/debug/dri/0/DP-1/dp_debug
```

Sample output

```
name = msm_dp
drm_dp_link
        rate = 270000
        num_lanes = 4
        capabilities = 1
dp_panel_info:
        active = 3840x2160
        back_porch = 80x23
        front_porch = 48x3
        sync_width = 32x5
        active_low = 0x0
        h_skew = 0
        refresh rate = 30
        pixel clock khz = 262750
        bpp = 30
dp_link:
        test_requested = 512
        num_lanes = 4
        bw_code = 10
        lclk = 270000000
```

```
        v_level = 1
        p_level = 0
```

View status of DisplayPort cable

To verify the DisplayPort connection, run the following command:

```
cat  /sys/class/drm/card0-DP-1/status
```

Sample output

```
Connected
```

Check EDID mode

To check for the EDID mode of the DisplayPort connected, run the following command:

```
cat  /sys/class/drm/card0-DP-1/modes
```

Sample output

```
3840x2160
2048x1080
1920x1080
1280x720
```

QCS8275

4.5 Debug DisplayPort

Note:

- You must enable SSH to access your host device. For instructions, see Sign in using SSH.
- To mount the `debugfs` file system to access the `debugfs` details of the display subsystem, run the following commands:

```
mount -o remount,rw /
```

```
mount -t debugfs none /sys/kernel/debug
```

View DisplayPort information

To check the selected DisplayPort, run the following command:

```
cat /sys/kernel/debug/dri/0/DP-2/dp_debug
```

Sample output

```
name = msm_dp
drm_dp_link
        rate = 270000
        num_lanes = 4
        capabilities = 1
dp_panel_info:
        active = 3840x2160
        back_porch = 80x23
        front_porch = 48x3
        sync_width = 32x5
        active_low = 0x0
        h_skew = 0
        refresh rate = 30
        pixel clock khz = 262750
        bpp = 30
dp_link:
        test_requested = 512
        num_lanes = 4
        bw_code = 10
        lclk = 270000000
        v_level = 1
        p_level = 0
```

View status of DisplayPort cable

To verify whether the DisplayPort cable is connected, run the following command:

```
cat  /sys/class/drm/card0-DP-2/status
```

Sample output

```
Connected
```

Check EDID mode

To check for the EDID mode of the DisplayPort connected, run the following command:

```
cat  /sys/class/drm/card0-DP-2/modes
```

Sample output

```
3840x2160
2048x1080
1920x1080
1280x720
```

4.6 Debug User mode and Weston

Configure Weston

The `/etc/initscripts/init_qti_display` script file includes the Weston configurations and the corresponding command-line parameters to set up the Weston screen autolaunch.

Note: You must enable SSH to access your host device. For instructions, see Sign in using SSH.

```
mount -o remount,rw /
```

```
cat /etc/initscripts/init_qti_display
```

Relaunch Weston

To kill the Weston process and manually relaunch the Weston server, do the following:

Note: You must enable SSH in Permissive mode to securely access your host device. For instructions, see How to SSH.

1. To identify the ID of the Weston processes, run the following commands:

```
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-
```

2. To kill the Weston process, run the following command:

```
killall weston
```

3. To relaunch the Weston server, run the following commands:

```
mount -o remount,rw /
```

```
export XDG_RUNTIME_DIR=/dev/socket/weston && mkdir -p $XDG_
RUNTIME_DIR
```

```
weston --idle-time=0 --continue-without-input
```

4.7 Plug in the HDMI/eDP cable after the device is powered ON

Explains the procedure to troubleshoot a no-display screen issue upon connecting the HDMI/eDP cable after the device is powered ON.

As a prerequisite, create the `Weston.ini` file and add the following Weston configurations to the file:

```
# configuration file for Weston
[core]
require-outputs=none
```

To push the `Weston.ini` file, do the following:

1. To push the file to the `/etc/xdg/weston/weston.ini` path, run the following command:

```
scp <weston.ini> root@<IP of the device>:/etc/xdg/weston/
weston.ini
```

2. Reboot the device.

   After powering on the device, connect the HDMI/eDP cable to see the Weston flower pattern appear on the monitor.

4.8 Generate the DRM mode test binary

By default, the DRM mode test binary isn't included in the display subsystem. To generate the DRM mode test binary file, do the following:

1. Ensure that you have the latest QLI software. If you don't, see Release Note and Qualcomm Linux Build Guide.
2. To generate the binary, go to the workspace <workspace> and run the following commands:

```
export SHELL=/bin/bash
```

```
MACHINE=qcs6490-rb3gen2-core-kit DISTRO=qcom-
wayland source setup-environment
```

```
bitbake -fc install libdrm
```

   The system generates the binary at `<workspace>/build-qcom-wayland/tmp-glibc/work/armv8-2a-qcom-linux/libdrm/2.4.120/image/usr/bin/modetest` directory.

Note: You must enable SSH to access your host device. For instructions, see Sign in using SSH.

3. To push the modetest binary file to the device, run the following commands:

```
mount -o remount,rw /usr
```

```
scp <workspace>/build-qcom-wayland/tmp-glibc/work/armv8-
2a-qcom-linux/libdrm/2.4.120/image/usr/bin/modetest
root@<IP of the device>:/usr/bin
```

Note: If prompted for a password, enter `oelinux123`.

5 References

5.1 Related documents

Title	
Open-source	
DRM internals	https://www.kernel.org/doc/html/v5.4/gpu/drm-internals.html
Kernel mode setting (KMS)	https://www.kernel.org/doc/html/v5.4/gpu/drm-kms.html
Standards	
HDMI Specification	www.hdmi.org/
MIPI Alliance Specification for D-PHY	mipi.org
MIPI Alliance Specification for DSI	mipi.org
MIPI Alliance Specification for DCS	mipi.org
VESA Display Stream Compression	www.vesa.org
VESA DisplayPort	www.vesa.org

5.2 Acronyms and terms

Acronym or Term	Definition
DPU	Display processing unit
DSC	Display stream compression
DSI	Display serial interface
DRM	Direct rendering manager
EDID	Extended display identification data
FHD	Full HD
IOCTL	I/O control
KMS	Kernel mode setting
LM	Layer mixer
MIPI DSI	Mobile industry processor interface display serial interface
SST	Single stream transport

Acronym or Term	Definition
UEFI	Unified extensible firmware interface

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.