Qualcomm Technologies, Inc.

RB3 Gen 2 Quick Start Guide

80-70018-253 AD

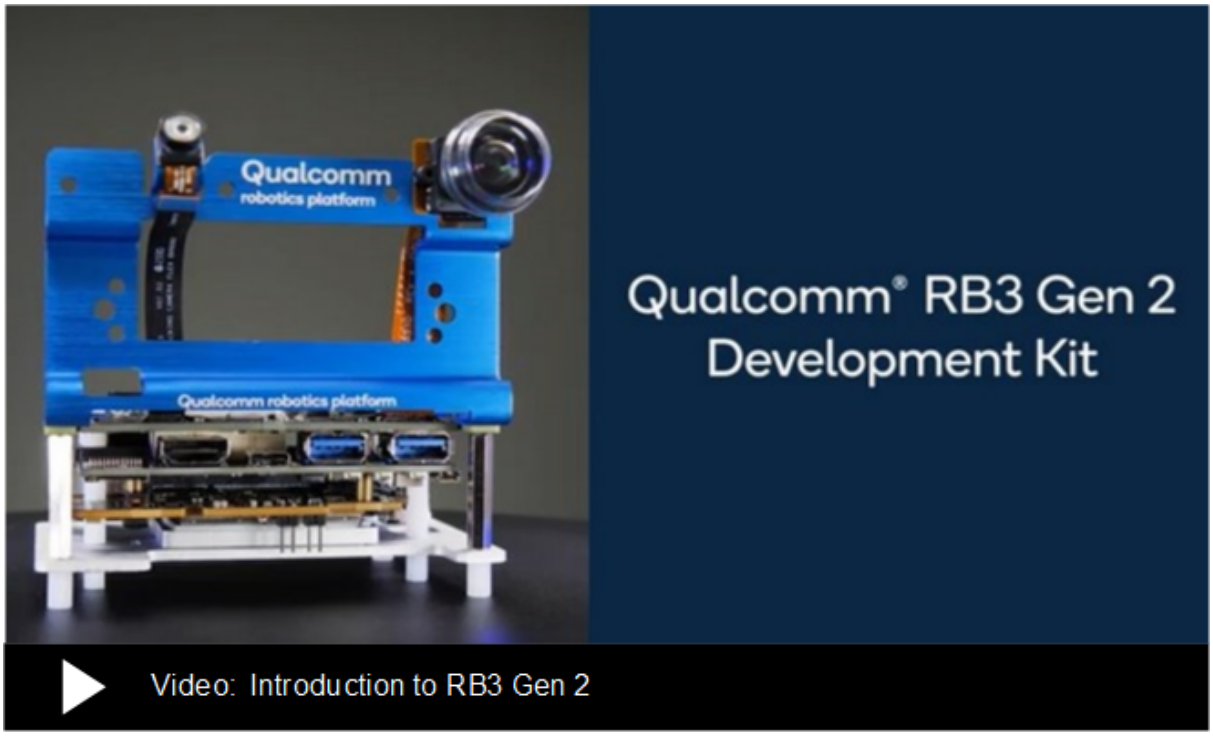
May 2, 2025

Contents

1 Get started

This Quick Start Guide helps you to set up the Qualcomm® RB3 Gen 2 Development Kit to explore and try out end-to-end multimedia and AI applications in a Linux-based environment. Additionally, this guide provides instructions on how to establish essential communication with host computers running Linux, Windows®, and macOS®.

Note: Qualcomm® Linux® allows you to develop applications for all SoCs listed at Hardware SoCs. Different development kits support specific SoCs. This Quick Start Guide applies to QCS6490 and QCS5430.



1.1 What's in the box

The Qualcomm RB3 Gen 2 Development Kit is available in two variants: the Core Kit and the Vision Kit.

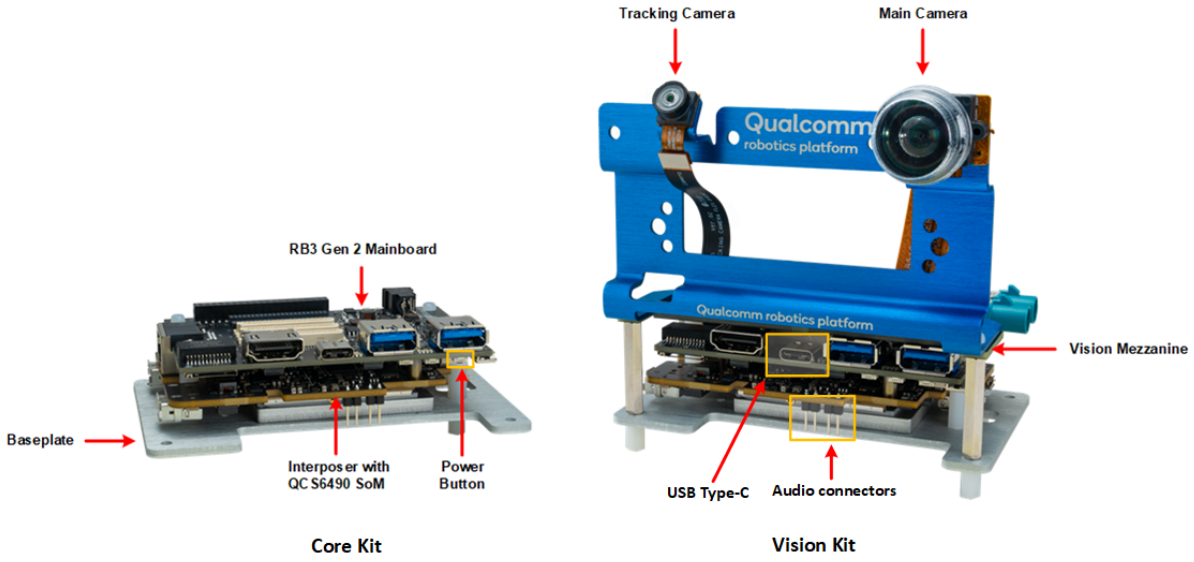


Figure: Core and Vision kits

Table: Core and Vision Kit components

Core Kit	Vision Kit
• Qualcomm RB3 Gen 2 mainboard • System on module (SoM) + interposer • Power supply (12 V) • USB-A to USB-C cable • 2 speakers • Pick tool to help access DIP switches	• All components of the Core Kit • Qualcomm Vision mezzanine board • IMX577 camera • OV9282 tracking camera

Note:

- The RB3 Gen 2 Vision Kit includes cameras with protective lens covers. Remove these covers before using the kit.
- Ensure that you have a computer connected to the internet.

1.2 Get familiar with the board

The following figure shows the locations of the different connectors on the mainboard and interposer with the baseplate removed. The image on the left shows the connectors from the top view of the mainboard. The image on the right shows the connectors from the bottom view of the interposer board (outlined in yellow) attached to the mainboard.

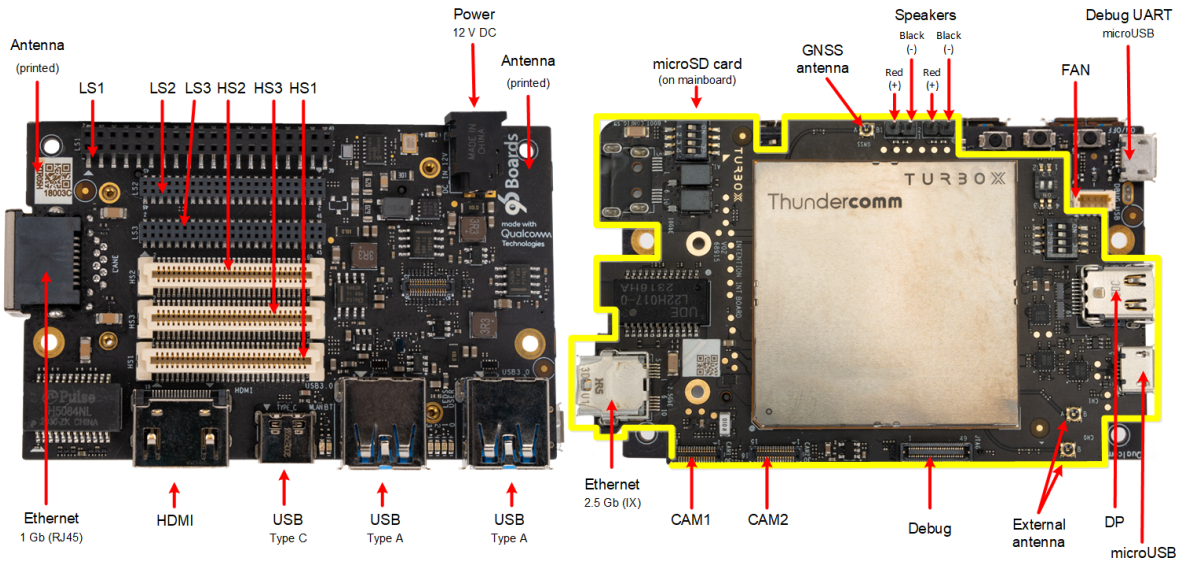
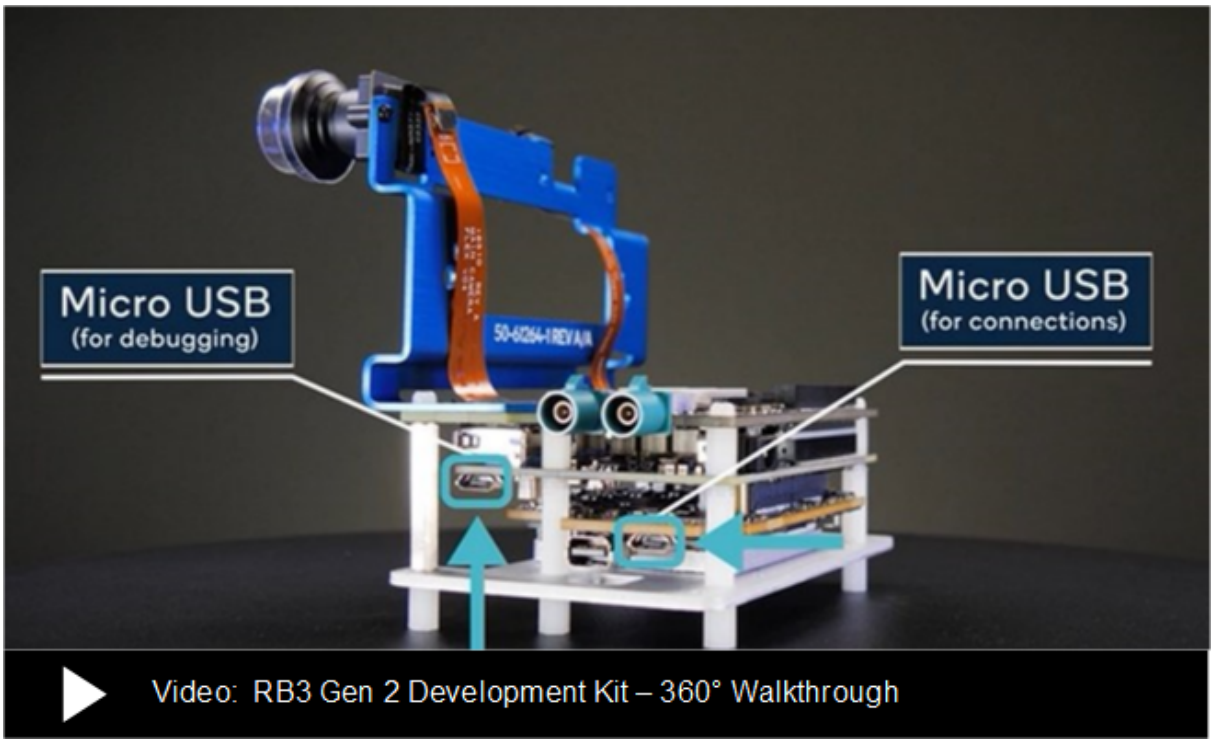


Figure: [L] Top view of mainboard connectors, [R] bottom view of interposer connectors



1.3 General workflow

Task	Actions
Set up the device	• Set up debug UART • Verify the Qualcomm Linux version • Connect to a network, Connect to SSH and HDMI display
Run sample applications	• Run multimedia and AI sample applications
Update the software	• Download the SDK • Move the device to emergency download (EDL) mode • Provision universal flash storage (UFS) and flash configuration data table (CDT)[1] • Flash using the Qualcomm Download (QDL) tool

[1] The Qualcomm® Linux® 1.3 release introduced a change in the partition table. Therefore, you must provision universal flash storage (UFS) and flash the configuration data table (CDT) on devices running Qualcomm Linux 1.0, 1.1, or 1.2 before upgrading to Qualcomm Linux 1.3 or later.

2 Set up the device

The Qualcomm® RB3 Gen 2 Development Kit comes preloaded with software, which allows quick device setup. You can seamlessly configure the RB3 Gen 2 device using Ubuntu, Windows®, and macOS®-based host computers.

2.1 Power up the device

Connect the +12-V power adapter, and do one of the following:

- Hold the **Power** button for 2-3 seconds until the power LED stays illuminated.
- Plug in the USB Type-C to USB Type-A or Type-C cable to establish an ADB connection and to flash the software.

2.2 Set up debug UART

The debug UART displays diagnostic messages and provides access to the device through the UART shell.

1. Connect one end of a Micro-USB cable to the Micro-USB port on the RB3 Gen 2 device.
2. Connect the other end of the Micro-USB cable to the host computer.

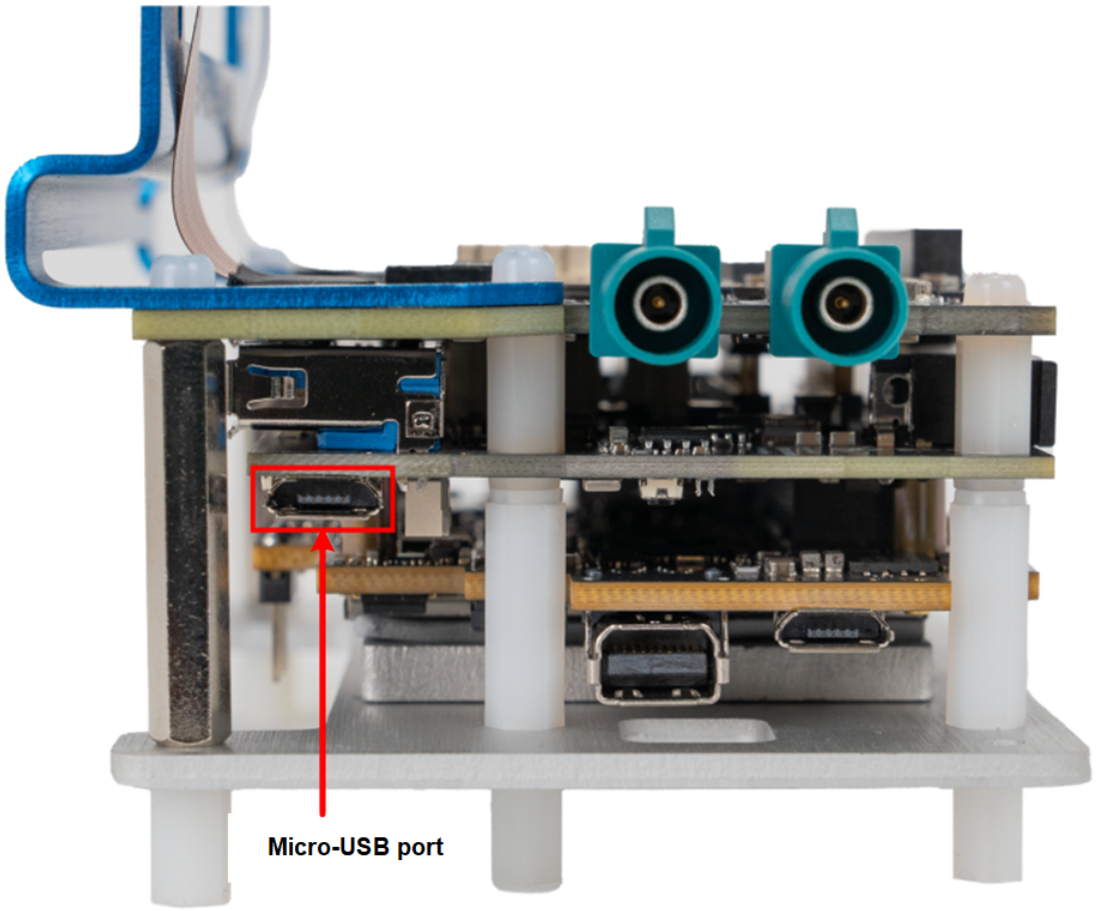


Figure: Location of the Micro-USB port

Select an option for detailed instructions depending on the host computer OS.

Ubuntu host

1. To install the `Screen` application on the Ubuntu host computer, run the following commands:

```
sudo apt update
```

```
sudo apt install screen
```

2. To verify the USB port, run the following command:

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

Sample output

```
/dev/ttyUSB0
```

3. To open the debug UART, run the following command:

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

Example command: `sudo screen /dev/ttyUSB0 115200`

4. Open the debug UART and select **ENTER** to bring up the authentication prompt.
5. To sign in to the UART console, specify the following:
   - **Sign in**: `root`
   - **Password**: `oelinux123`

Note: If you don't see the authentication console as expected, check the USB connection. If needed, disconnect and reconnect the Micro USB.

Windows host

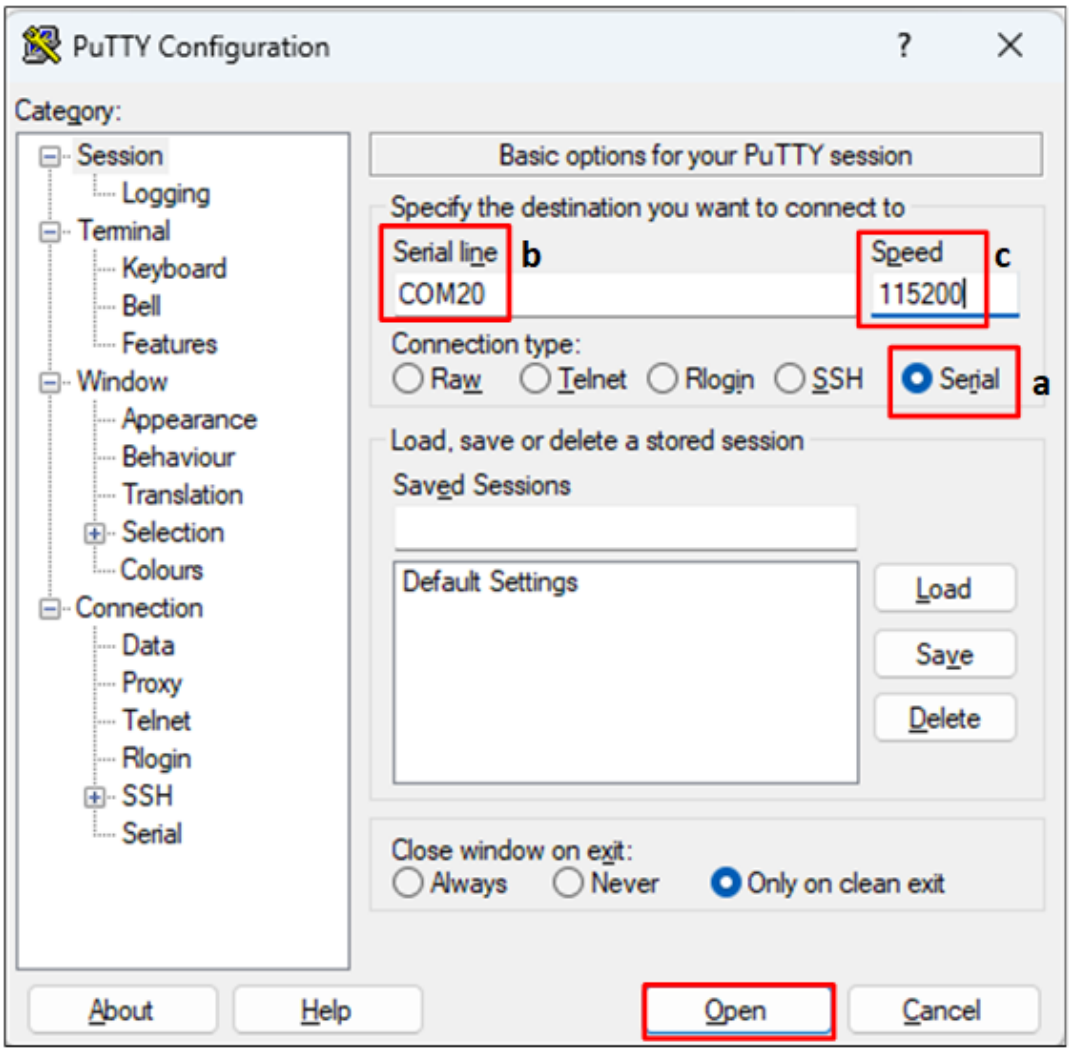
1. Download PuTTY for your Windows host computer. Ensure to choose the correct version for your Windows (32-bit or 64-bit).
2. Run the installation wizard and follow the prompts.
3. After installation, open the PuTTY application from the list of installed programs in the **Start** menu or search for it using the search bar in the taskbar.
4. In the **PuTTY Configuration** dialog, do the following:
   a. Select **Serial**.
   b. Specify a **Serial line** based on the UART port detected on the **Windows Device Manager**.

Note: If the UART port isn't detected, download the driver and use Windows Device Manager to update the driver.

- On x86: USB-to-UART serial driver
- On Arm®: Go to the https://oemdrivers.com/usb-ft232r-usb-uart-arm64 web page. In the **Drivers** section, click **FTDI CDM VCP Drivers**.

---

   c. Set the baud rate to **115200**.
   d. Select **Open** to start a PuTTY session.



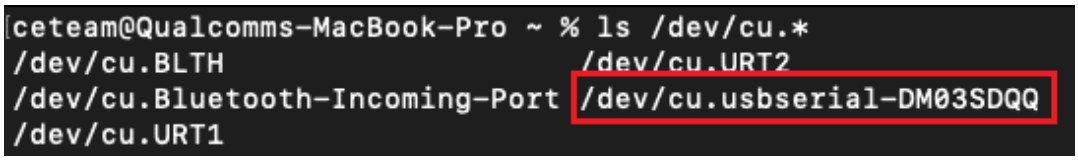
5. Open the serial device and select **ENTER** to bring up the authentication prompt.
6. To sign in to the UART console, provide the following:
    - **Sign in**: `root`
    - **Password**: `oelinux123`

macOS host

1. To check for the serial devices connected to the macOS host computer, run the following command:

```
ls /dev/cu.*
```

2. Locate the serial device from the listed serial devices. The name of your device should display in the list.



3. To open the serial device, run the following command:

```
screen <serial_device_node> <baud_rate>
```

Note:

- Replace `<serial_device_node>` with your device node name.
- Replace `<baud_rate>` with the appropriate baud rate.

Example command `ceteam@Qualcomms-MacBook-Pro ~ % screen /dev/cu.usbserial-DM03SDQQ 115200`

4. Open the serial device and select the **ENTER** key to bring up the authentication prompt.
5. To sign in to the UART console, provide the following:
   - **Sign in**: `root`
   - **Password**: `oelinux123`

2.3 Verify the Qualcomm Linux version

To verify the Qualcomm Linux version, run one of the following commands in your device shell:

- ```
  uname -a
  ```

  The output shows the Qualcomm Linux version.

  ```
  Linux qcs6490-rb3gen2-vision-kit 6.6.65-qli-1.4-ver.1.
  1-04093-ge21546bdd315-dirty #1 SMP PREEMPT Fri Mar 28 13:
  13:01 UTC 2025 aarch64 GNU/Linux
  ```

or

- ```
  cat /etc/os-release
  ```

  The output shows the Qualcomm Linux version.

  ```
  ID=qcom-wayland
  NAME="Qualcomm Linux"
  VERSION="1.4-ver.1.1"
  VERSION_ID=1.4-ver.1.1
  PRETTY_NAME="Qualcomm Linux 1.4-ver.1.1"
  CPE_NAME="cpe:/o:openembedded:qcom-wayland:1.4-ver.1.1"
  ```

Note: These commands are supported only on Qualcomm Linux 1.3 or later versions.

If you're using an older Qualcomm Linux version and want to upgrade, see Update the software.

2.4 Connect to a network

Establish an internet connection to retrieve the device IP address using one of the following methods.

Connect using Ethernet

1. Insert one end of the Ethernet cable into the Ethernet port (RJ45) of the RB3 Gen 2 device and connect the other end to your network router.

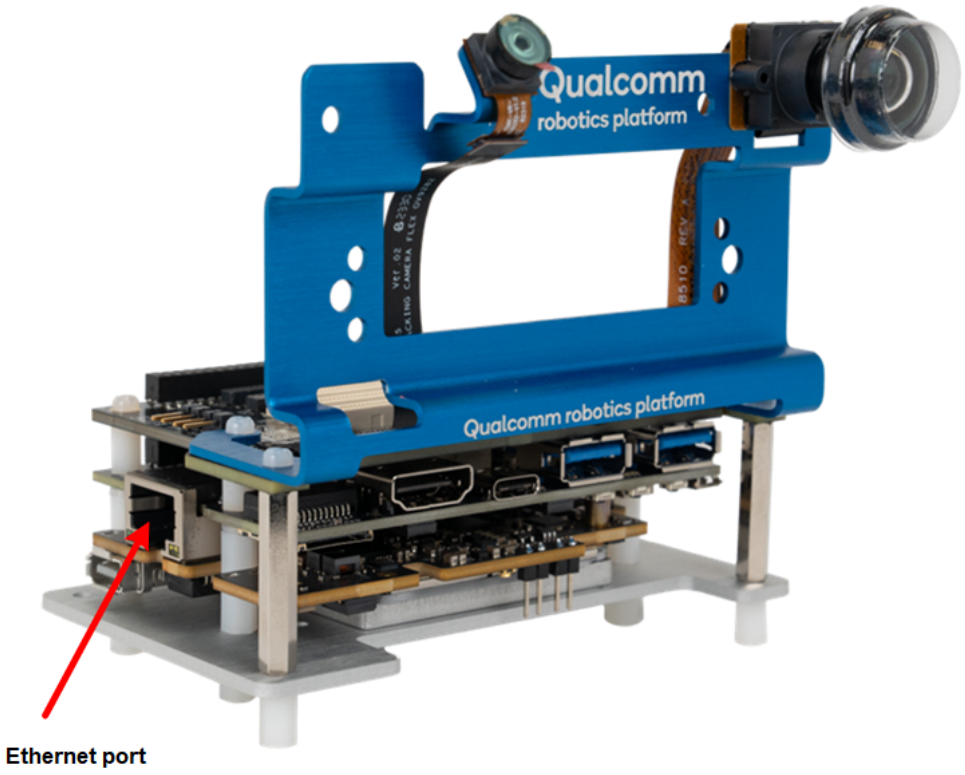
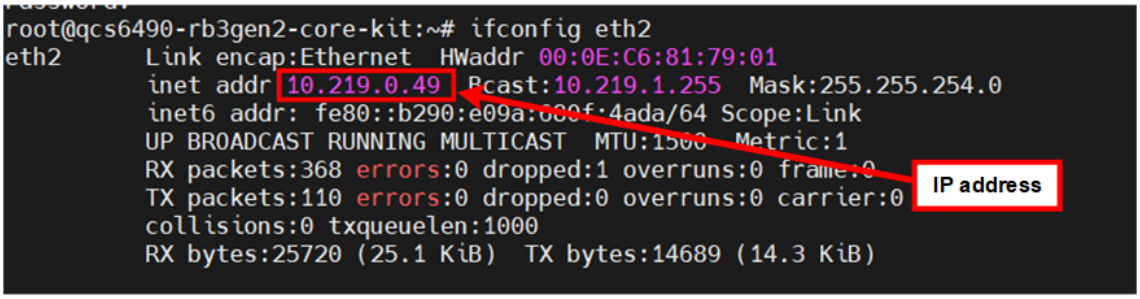


Figure: Location of the Ethernet port

2. After establishing the connection, run the following command on the UART serial console to obtain the IP address:

```
ifconfig eth2
```



Note: If you face issues in establishing the connection, see Update USB and Ethernet controller firmware.

Connect using Wi-Fi

The device operates Wi-Fi in Station mode. When the device boots up, it initializes the Wi-Fi host driver and authenticates for network management. You can establish a wireless connection using the **nmcli** command-line tool.

1. To connect to the wireless access point (Wi-Fi router), run the following command:

   ```
   nmcli dev wifi connect <WiFi-SSID> password <WiFi-password>
   ```

   Example

   ```
   root@qcs6490-rb3gen2-vision-kit:~# nmcli dev wifi connect QualcommWiFi password 1234567890
   ```

   ```
   Device 'wlan0' successfully activated with 'd7b990bd-3b77-4b13-b239-b706553abaf8'.
   ```

2. To verify connection and device status, run the following command:

   ```
   nmcli -p device
   ```

```
root@qcs6490-rb3gen2-vision-kit:~# nmcli -p device
=====================
  Status of devices
=====================
DEVICE  TYPE      STATE         CONNECTION
---------------------------------------------------------------------------
wlan0   wifi      connected     QualcommWiFi
eth0    ethernet  unavailable   --
eth1    ethernet  unavailable   --
can0    can       unmanaged     --
lo      loopback  unmanaged     --
root@qcs6490-rb3gen2-vision-kit:~#
```

3. To verify the WLAN connection status and IP address, run the following command:

```
ifconfig wlan0
```

```
root@qcs6490-rb3gen2-vision-kit:~# ifconfig wlan0
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
       inet 192.168.0.222  netmask 255.255.255.0  broadcast 192.168.0.255
       inet6 2401:4900:38f0:c563:dae0:89c5:389f:2cf0  prefixlen 64  scopeid 0x0<global>
       inet6 2401:4900:38f0:c563:36ca:e6d4:f00:d938  prefixlen 64  scopeid 0x0<global>
       inet6 fe80::54c6:3347:dce4:8f15  prefixlen 64  scopeid 0x20<link>
       inet6 fe80::603d:907a:f505:60f1  prefixlen 64  scopeid 0x20<link>
       ether 00:03:7f:12:a5:a5  txqueuelen 1000  (Ethernet)
       RX packets 131  bytes 12762 (12.4 KiB)
       RX errors 0  dropped 0  overruns 0  frame 0
       TX packets 181  bytes 23311 (22.7 KiB)
       TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

IP address

4. To ensure the connection is active, ping a website like Qualcomm:

```
ping qualcomm.com
```

2.5 Connect to SSH

SSH facilitates secure file transfers between the host computer and the RB3 Gen 2 device.

Ensure network connectivity before connecting to SSH.

1. To find the IP address of the RB3 Gen 2 device based on the type of network connection, run the appropriate command on the UART console of the host computer:

   For Ethernet:

```
ifconfig eth2
```

   For Wi-Fi:

```
ifconfig wlan0
```

2. From the host computer, run the following command to establish an SSH connection to the device. Use the IP address obtained from **step 1**.

```
ssh root@ip-address
```

Example

```
ssh root@192.168.0.222
```

3. To connect to the SSH, enter the following password:

```
oelinux123
```

Note: Ensure that the remote host connects to the same Wi-Fi access point as the device.

2.6 Connect to HDMI display

To facilitate HDMI display and view the output of sample applications, follow these steps:

1. Connect one end of the HDMI cable to the HDMI port on the RB3 Gen 2 device.
2. Connect the other end of the HDMI cable to your display monitor.

This allows you to view the output of sample applications on your connected display.

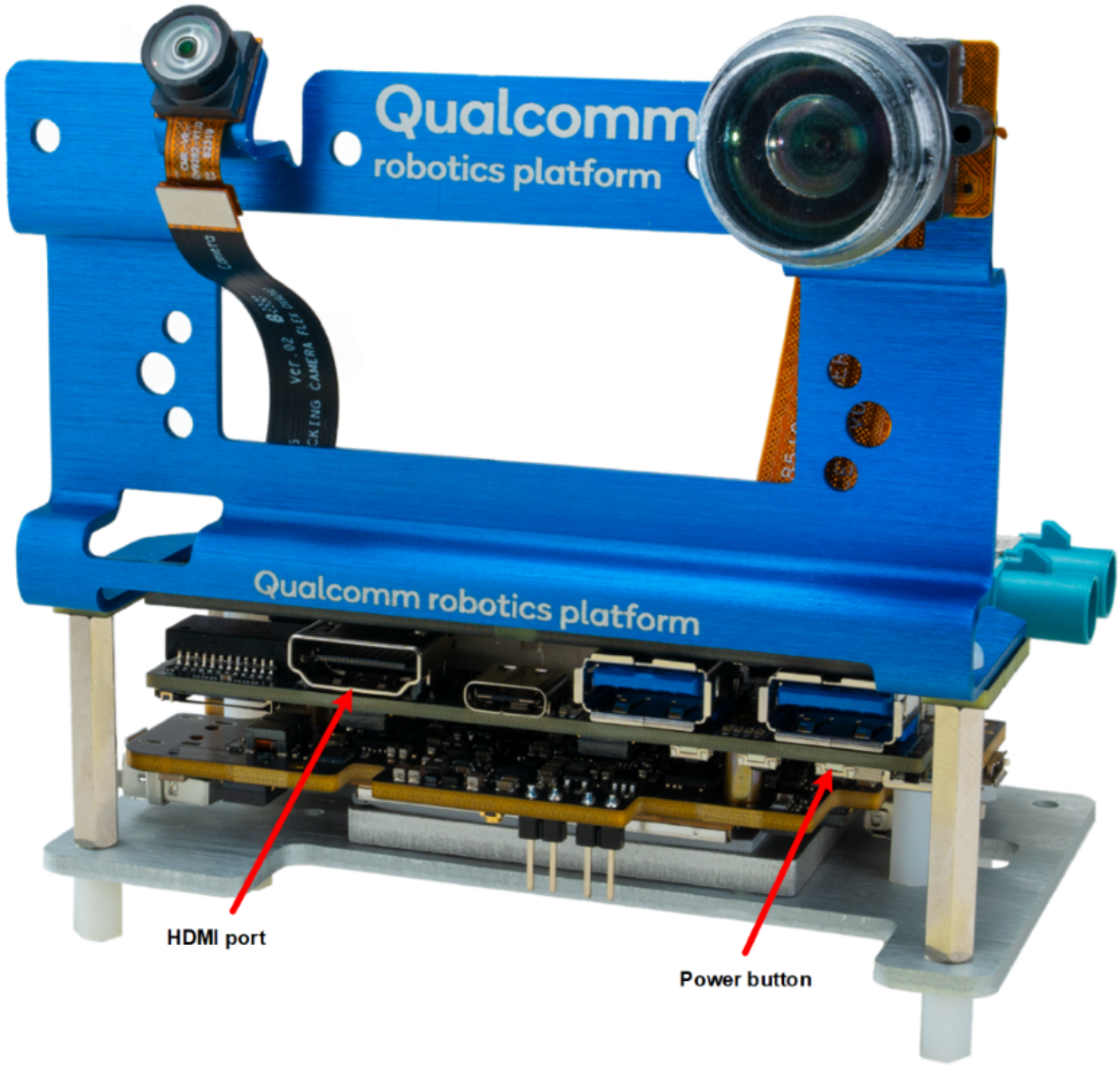


Figure: Location of the HDMI port and power button

Power on the device and check for the HDMI display. Look for a specific pattern on the display monitor.

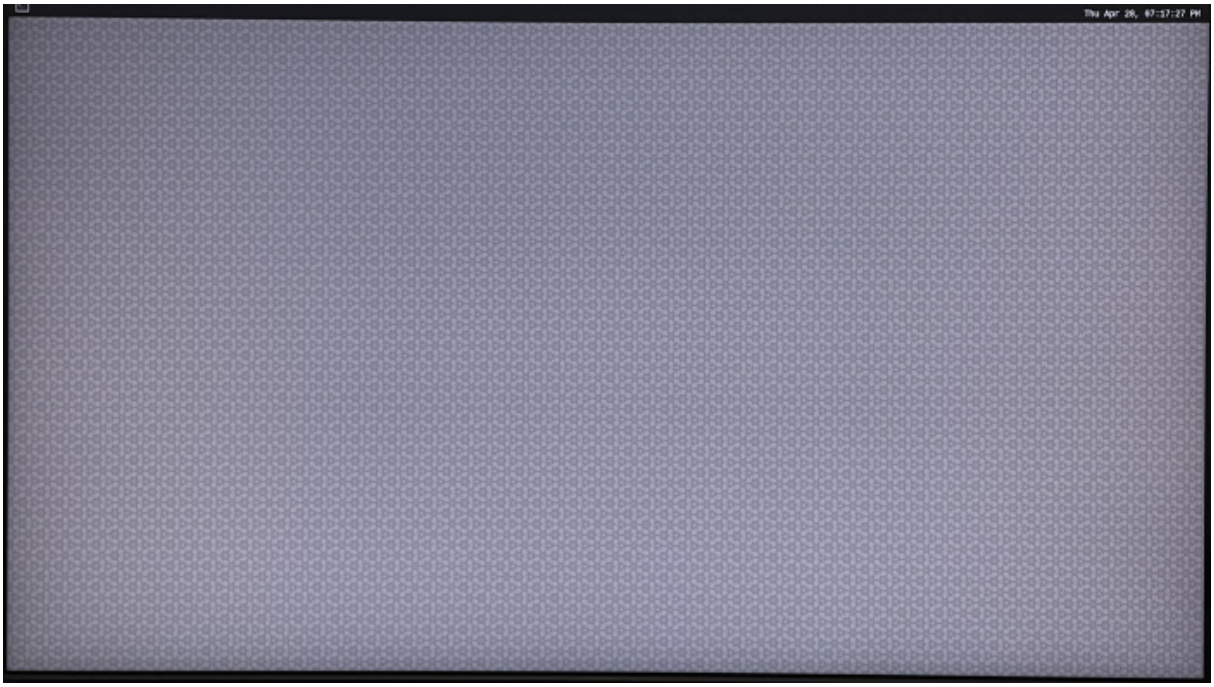


Figure: Weston flower pattern

If the display doesn't appear as expected, toggle the **Switch 4** of **DIP_SW_0** to OFF. For more details, see DIP switches.

If the issue persists, see Debug HDMI display to resolve the issue.

3 Run sample applications

Qualcomm® Linux® includes various sample applications. For more information, see Sample applications.

Two main groups of sample applications are available, each suited for different use cases such as retail, Qualcomm® Edge AI Box Solutions, and IP camera.

Run multimedia sample applications

These sample applications focus on camera, video, and audio functionalities.

Run AI sample applications

These sample applications focus on AI and machine learning (ML) capabilities.

You can run these applications using one of the following methods:

- Run sample applications using Qdemo, a GUI-based application
- Run sample applications manually
  - Run multimedia sample applications
  - Run AI sample applications

Note:

- To run the multimedia and AI applications, set up the Wi-Fi and establish SSH connectivity.
- To view the display output, connect the HDMI display to the HDMI port of the RB3 Gen 2 device (see Connect to HDMI display).
- To enable audio, see Qualcomm Linux Audio Guide.

3.1 Run sample applications using Qdemo

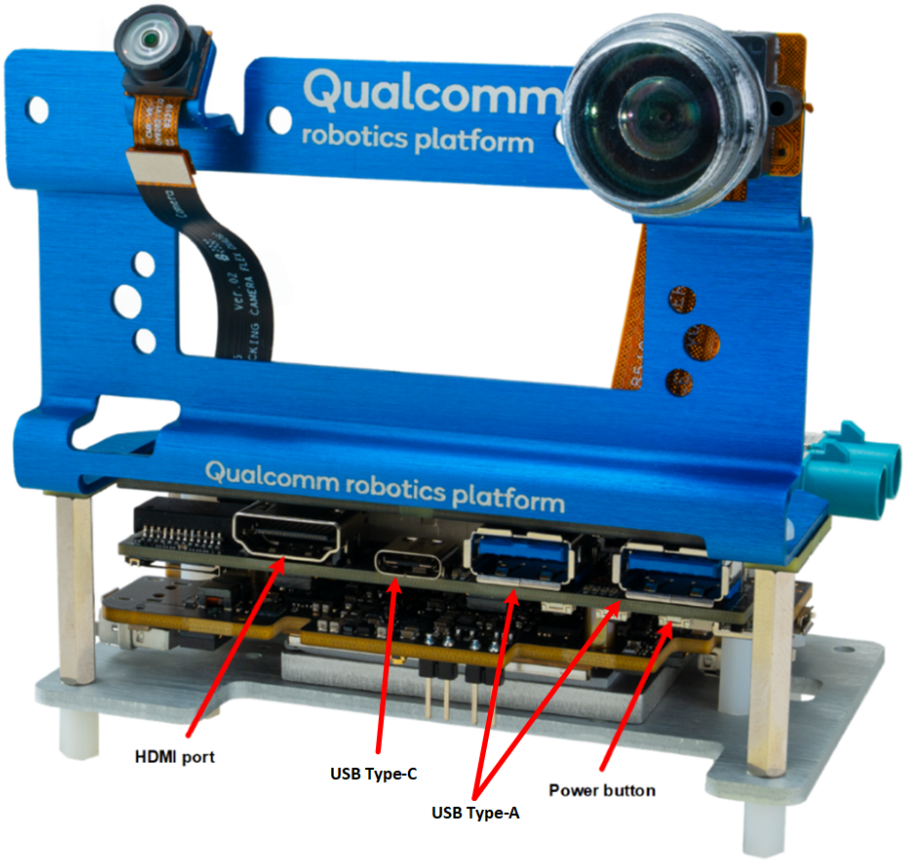
Qdemo is a GUI-based application that allows you to experience the AI and multimedia capabilities of the Qualcomm® RB3 Gen 2 Development Kit with a single click. You can get started with this application without the host setup and can connect to the Wi-Fi also.

Prerequisites

- HDMI monitor
- Mouse
- Keyboard
- Wi-Fi connection
- Recorded video to run the video wall application

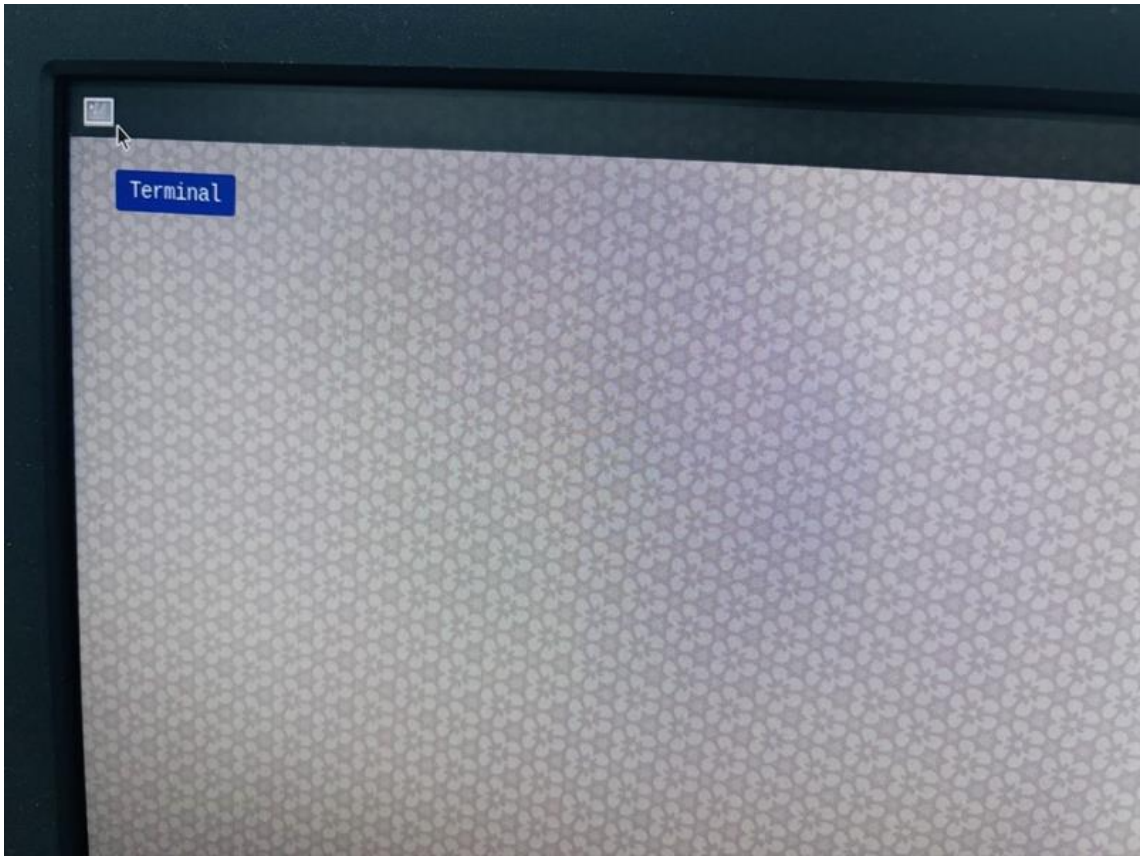
Procedure

1. Connect the mouse, keyboard, and HDMI monitor to the device, and then power on the device.



If you face any keyboard or mouse connectivity issue, update the USB firmware. For more information, see FAQs.

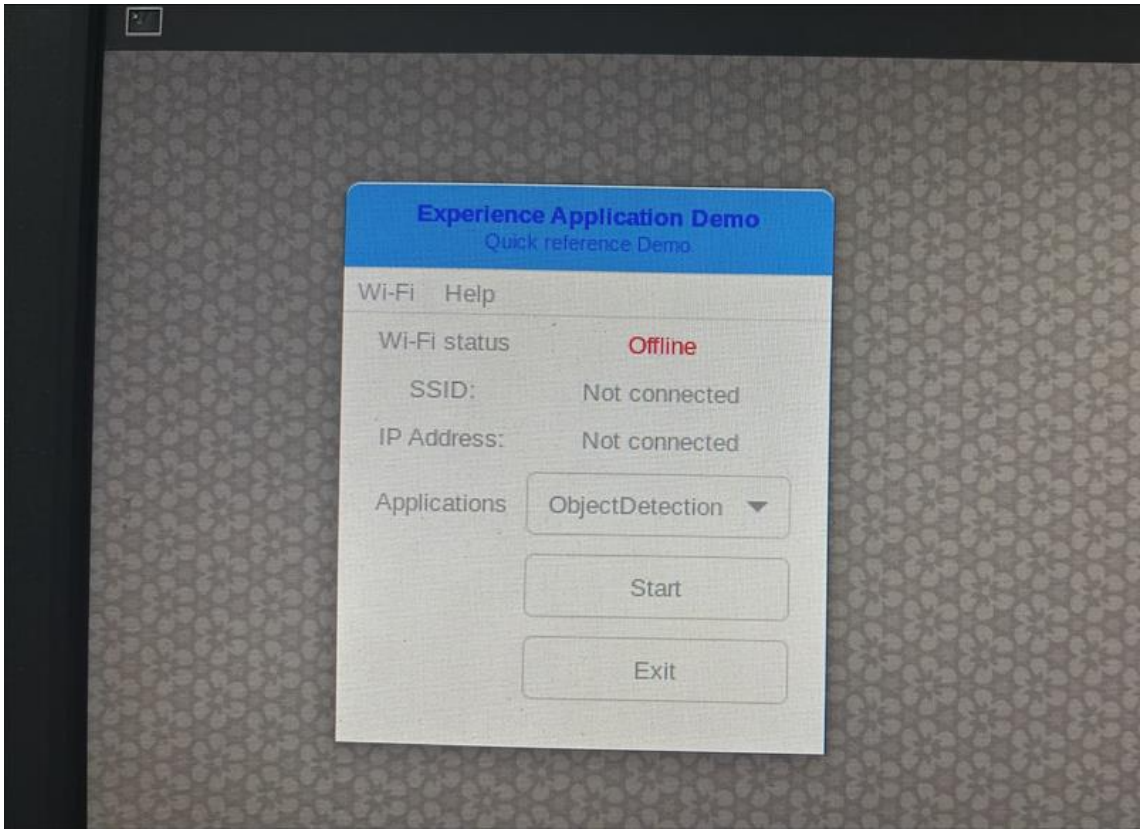
2. Click the terminal icon on the Wayland display.



3. Once the terminal opens, run the Qdemo application.

```
Qdemo
```

4. Connect to a Wi-Fi network.



5. In the *Applications* drop-down, select an application and click *Start*.

   When running the application for the first time on the device, this step may take more than a minute because the device downloads the AI models and labels from GitHub. When running the application subsequently, this step takes just a few seconds for execution.

6. Close the pop-up window to stop the use case.
7. Click *Exit* to close the application.

3.2 Run multimedia sample applications

The multimedia sample applications show use cases for camera, display, and video streams on the RB3 Gen 2 device.

Multicamera streaming or encoding (Dash cam)

The **gst-multi-camera-example** command-line application shows simultaneous streaming from two camera sensors on the RB3 Gen 2 device. The application composes the camera feeds side-by-side to display on a screen or encodes and stores the video streams to files.

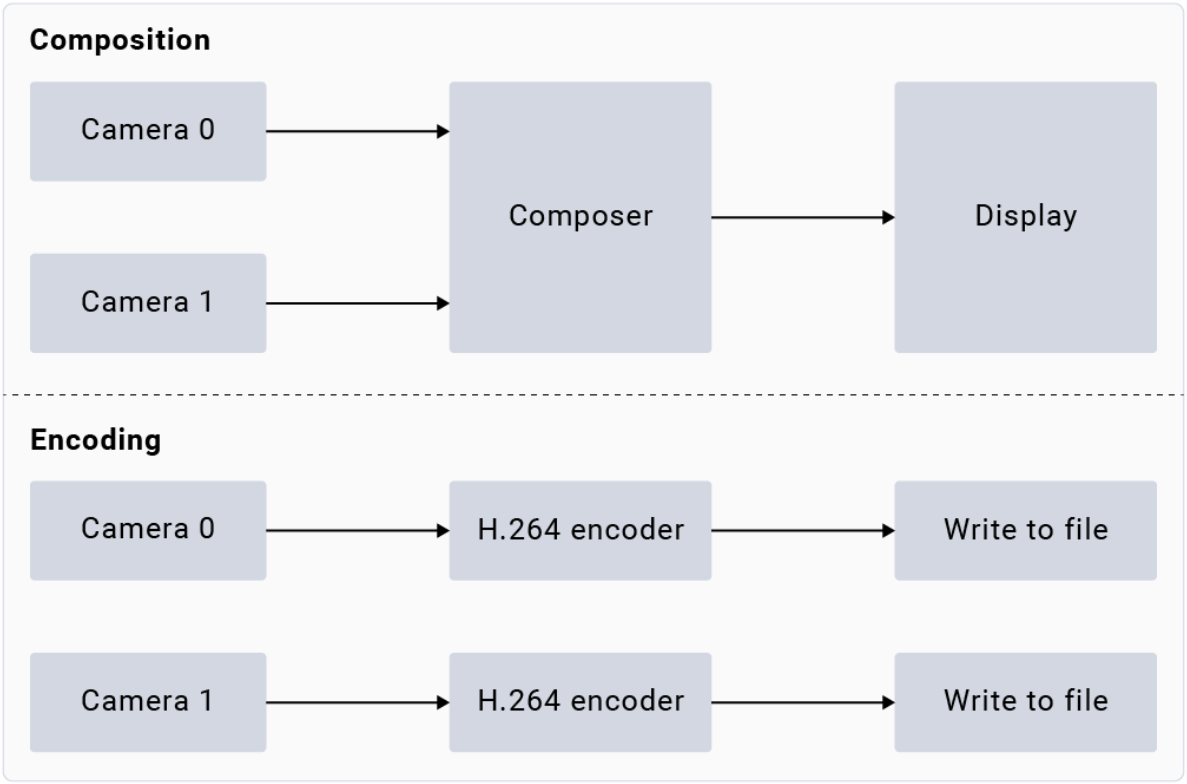
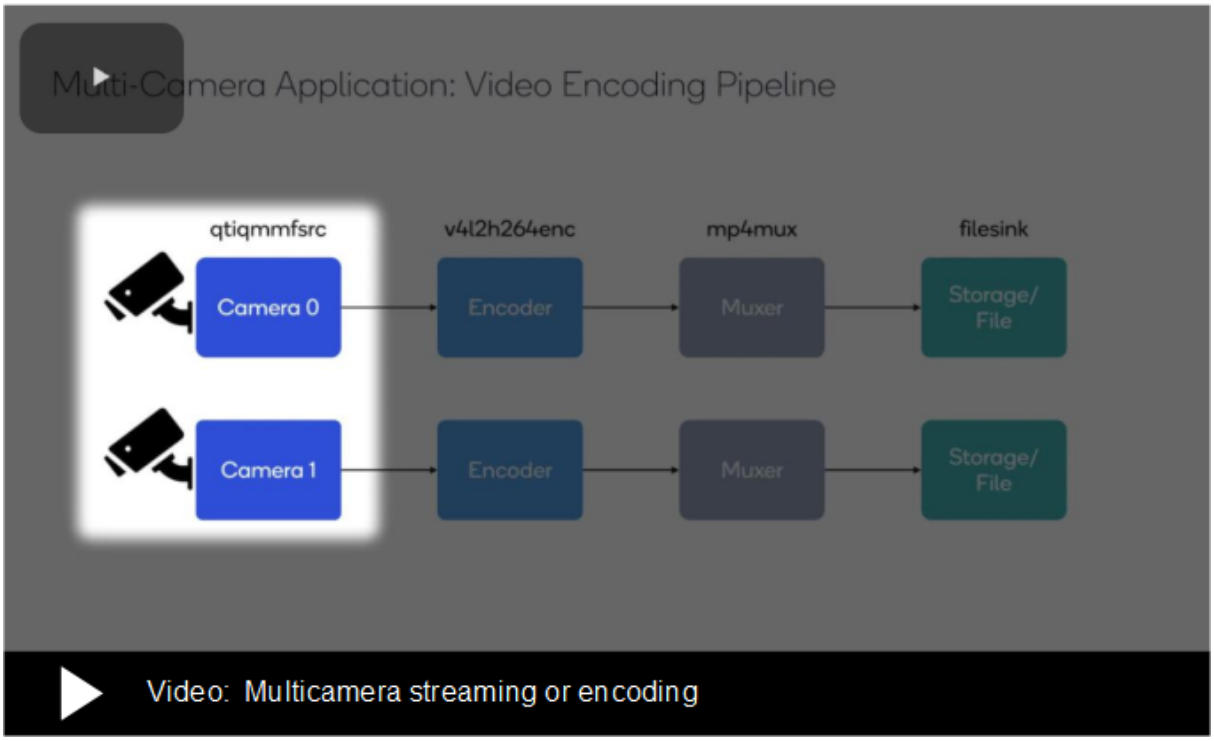


Figure: Dash cam application workflow



Example usage

To start the application, run the following use cases from the SSH terminal:

1. To view the sample application on the HDMI display, run the following export command:

```
export XDG_RUNTIME_DIR=/dev/socket/weston && export WAYLAND_
DISPLAY=wayland-1
```

Note: If Weston isn't enabled automatically, start two instances of secure shell: one to enable Weston and the other to run the application.

   a. To enable Weston, run the following command in the first shell:

```
export GBM_BACKEND=msm && export XDG_RUNTIME_DIR=/dev/socket/
weston && mkdir -p $XDG_RUNTIME_DIR && weston --continue-
without-input --idle-time=0
```

   b. To set up the Wayland Display environment, run the following command in the second shell:

```
export XDG_RUNTIME_DIR=/dev/socket/weston && export WAYLAND_
DISPLAY=wayland-1
```

2. To view the `waylandsink` output, run the following command:

```
gst-multi-camera-example -o 0
```

3. To store the encoder output, do the following:
   a. Run the following command:

```
gst-multi-camera-example -o 1
```

      The device stores the encoded files at `/opt/cam1_vid.mp4` and `/opt/cam2_vid.mp4` for camera 1 and camera 2, respectively.

   b. To pull the files from the host computer, run the following command:

```
scp root@<IP address of target device>:/opt/cam1_vid.mp4
<destination directory>
```

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

   c. To play the encoder output, you can use any media player that supports MP4 files.

- To stop the use case, select **CTRL + C**.
- To display the available help options, run the following command:

```
gst-multi-camera-example --help
```

- The `GST_DEBUG` environment variable controls the GStreamer debug output. Set the required level to allow logging. For example, to log all warnings, run the following command:

```
export GST_DEBUG=2
```

Multichannel video decode and compose (Video wall)

The **gst-concurrent-videoplay-composition** command-line application allows concurrent video decoding and playback for AVC-coded videos and composes them on a display for video wall applications. The application requires at least one input video file, which should be an MP4 file with the AVC codec.

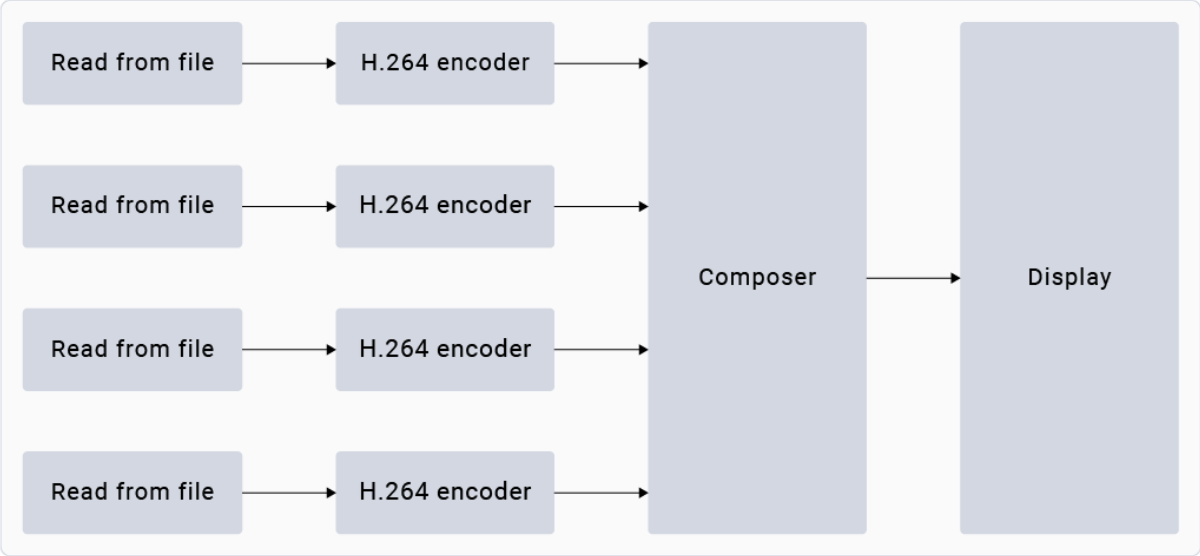
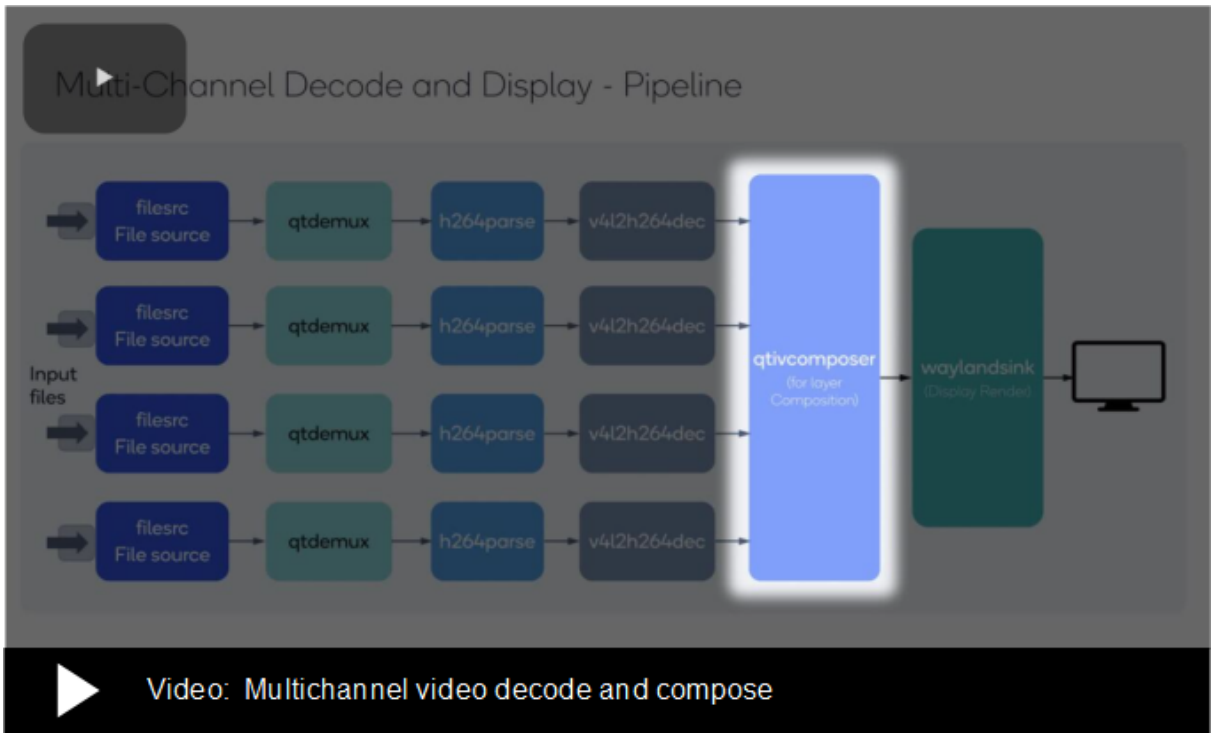


Figure: Video wall application workflow



Example usage

1. To transfer prerecorded or test videos that are in the AVC-encoded MP4 (H.264) format (with the filename as `<file_name>`) to your device, run the following command on the host computer:

```
scp <file_name> root@[DEVICE IP-ADDR]:/opt/
```

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

2. To view the sample application on the HDMI display, run the following export command from the SSH terminal:

```
export XDG_RUNTIME_DIR=/dev/socket/weston && export WAYLAND_
DISPLAY=wayland-1
```

Note: If Weston isn't enabled automatically, start two instances of secure shell: one to enable Weston and the other to run the application.

   a. To enable Weston, run the following command in the first shell:

```
export GBM_BACKEND=msm && export XDG_RUNTIME_DIR=/dev/socket/
weston && mkdir -p $XDG_RUNTIME_DIR && weston --continue-
without-input --idle-time=0
```

   b. To set up the Wayland Display environment, run the following command in the second shell:

```
export XDG_RUNTIME_DIR=/dev/socket/weston && export WAYLAND_
DISPLAY=wayland-1
```

3. To start concurrent playback for four channels, run the following command:

```
gst-concurrent-videoplay-composition -c 4 -i /opt/<file_name1>.
mp4 -i /opt/<file_name2>.mp4 -i /opt/<file_name3>.mp4 -i /opt/
<file_name4>.mp4
```

Note:

- `-c`: specifies the number of streams to be decoded for composition can be either 2, 4, or 8.
- `-i`: specifies the absolute path to the input video file.

- To stop the use case, select **CTRL + C**.
- To display the available help options, run the following command:

```
gst-concurrent-videoplay-composition --help
```

- The `GST_DEBUG` environment variable controls the GStreamer debug output. Set the required level to allow logging. For example, to log all warnings, run the following command:

```
export GST_DEBUG=2
```

3.3 Run AI sample applications

AI sample applications show use cases for object detection and parallel inferencing on input streams from a camera, video file, or Real-Time Streaming Protocol (RTSP) stream on the RB3 Gen 2 device. To run these sample applications, you must obtain AI models from Qualcomm® AI Hub and labels from GitHub. The procedure involves downloading the models and labels, transferring them to the RB3 Gen 2 device, and running the sample applications.

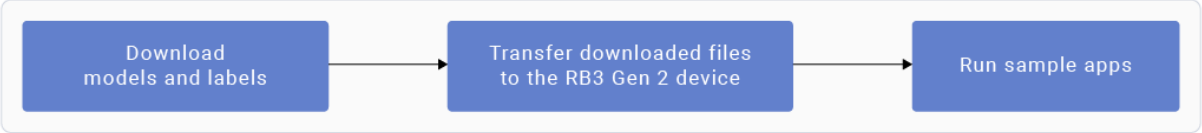


Figure: Workflow for running AI sample applications

Prerequisite

AI sample applications require model and label files on the device to run the application.

Procedure

1. You require the following models for the AI sample applications:

Sample application	Models required
AI object detection	Yolov8-Detection-Quantized
Parallel AI inference	Yolov8-Detection-Quantized
	Inception-v3-Quantized
	HRNetPoseQuantized
	DeepLabV3-Plus-MobileNet-Quantized

2. Download and run the automated script to get the model and label files on the device:

```
curl -L -O https://raw.githubusercontent.com/quic/sample-apps-
for-qualcomm-linux/refs/heads/main/download_artifacts.sh
```

```
chmod +x download_artifacts.sh
```

```
./download_artifacts.sh -v GA1.4-rel -c QCS6490
```

Note: The YOLOv8 models aren't part of the script. You need to export these models using the Qualcomm AI Hub APIs.

3. Export YOLOv8 from Qualcomm AI Hub.

   Follow these validated instructions to export models on your host computer using Ubuntu 22.04. You can also run these instructions on Windows through Windows Subsystem for Linux (WSL) or set up an Ubuntu 22.04 virtual machine on macOS. For more details, see Virtual Machine Setup Guide.

   a. Obtain the shell script for exporting the models:

   ```
   wget https://raw.githubusercontent.com/quic/sample-
   apps-for-qualcomm-linux/refs/heads/main/scripts/
   export_model.sh
   ```

   b. Update the script permissions to make it executable:

   ```
   chmod +x export_model.sh
   ```

   c. Run the export script with your Qualcomm AI Hub API token as the value for the `--api-token` argument:

   ```
   ./export_model.sh --api-token=<Your AI Hub API Token>
   ```

   Note: You can find your Qualcomm AI Hub API token in your account settings.

   d. The script downloads the models to the `build` directory. Copy these models to the `/etc/models/` directory of your device using the following commands:

   ```
   scp <working directory>/build/yolonas_quantized/
   yolonas_quantized.tflite root@<IP address of target
   device>:/etc/models/
   ```

   ```
   scp <working directory>/build/yolov8_det_quantized/
   yolov8_det_quantized.tflite root@<IP address of target
   ```

```
device>:/etc/models/
```

4. Update the `q_offset` and `q_scale` constants of the quantized LiteRT model in the JSON file. For instructions, see Get the model constants.
5. Use the following command to push the downloaded model files to the device:

```
scp <model filename> root@<IP addr of the target device>:/etc/
models
```

Example

```
scp inception_v3_quantized.tflite root@<IP addr of the target
device>:/etc/models/
```

6. Create a directory for test videos using the following commands:

```
ssh root@<ip-addr of the target device>
```

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

```
mkdir /etc/media/
```

7. From the host computer, push the test video files to the device:

```
scp <filename>.mp4 root@<IP address of target device>:/
etc/media/
```

AI object detection

The **gst-ai-object-detection** sample application shows the hardware capability to detect objects on input streams from a camera, video file, or RTSP stream. The pipeline receives the input stream, preprocesses it, runs inferences on AI hardware, and displays the results on the screen.

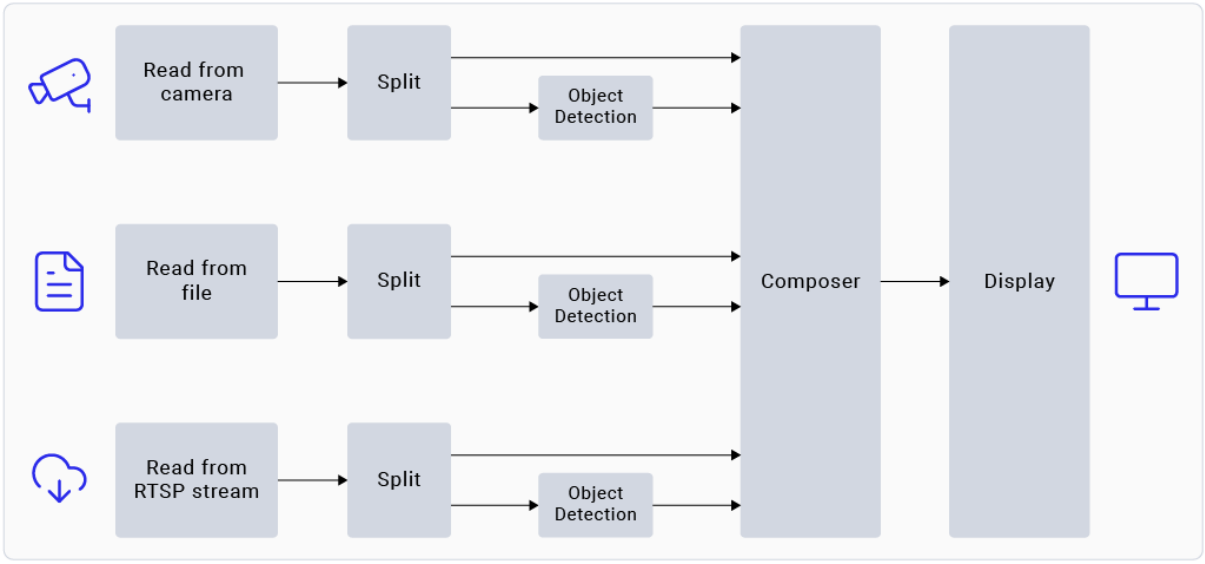
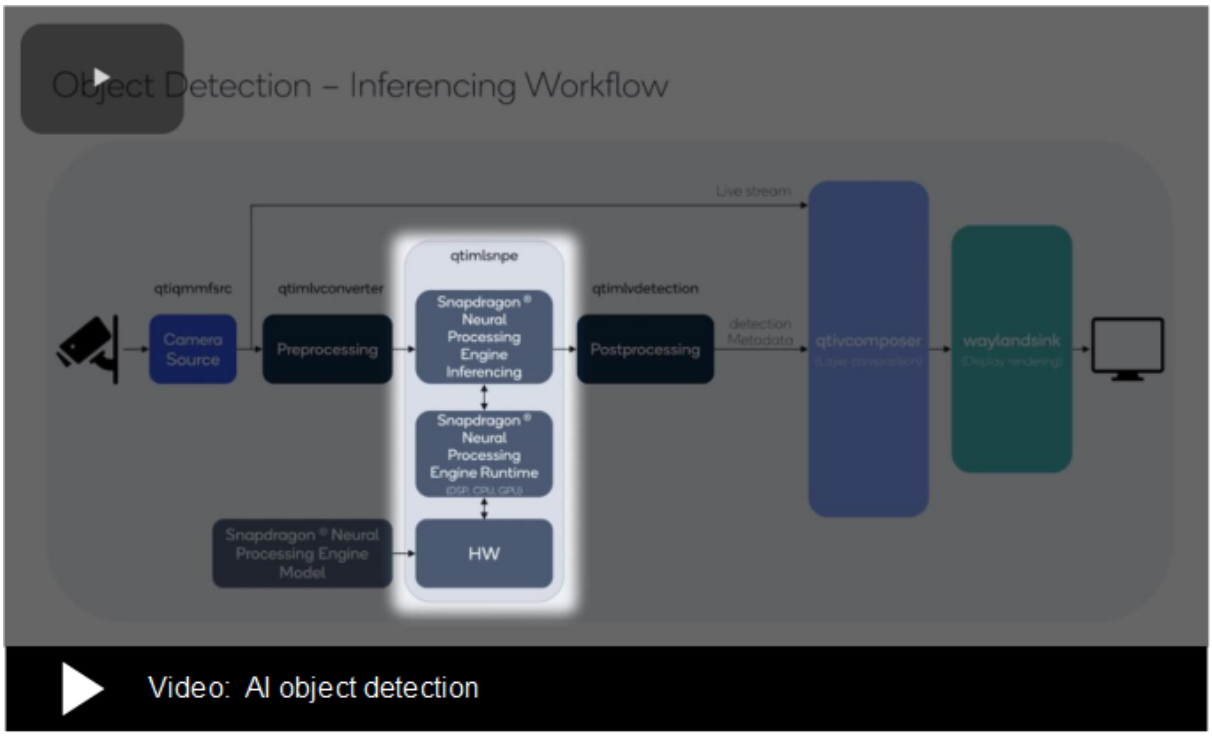


Figure: Object detection application workflow



Example usage

It's mandatory to push the model and label files to the device to run the sample application. For details, see Procedure.

1. Begin a new SSH session and start the HDMI display monitor if you haven't already:

```
ssh root@<ip-addr of the target device>
```

2. To view the sample application on the HDMI display, run the following export command from the SSH terminal:

```
export XDG_RUNTIME_DIR=/dev/socket/weston && export WAYLAND_
DISPLAY=wayland-1
```

Note: If Weston isn't enabled automatically, start two instances of secure shell: one to enable Weston and the other to run the application.

a. To enable Weston, run the following command in the first shell:

```
export GBM_BACKEND=msm && export XDG_RUNTIME_DIR=/dev/socket/
weston && mkdir -p $XDG_RUNTIME_DIR && weston --continue-
without-input --idle-time=0
```

b. To set up the Wayland Display environment, run the following command in the second shell:

```
export XDG_RUNTIME_DIR=/dev/socket/weston && export WAYLAND_
DISPLAY=wayland-1
```

---

3. Edit the `/etc/configs/config_detection.json` file on your device.

```
{
   "file-path": "/etc/media/video.mp4",
   "ml-framework": "tflite",
   "yolo-model-type": "yolov8",
   "model": "/etc/models/yolov8_det_quantized.tflite",
   "labels": "/etc/labels/yolonas.labels",
   "constants": "YOLOv8,q-offsets=<21.0, 0.0, 0.0>,q-scales=<3.
0546178817749023, 0.003793874057009816, 1.0>;",
   "threshold": 40,
   "runtime": "dsp"
}
```

Field	Values/description
ml-framework	
`snpe`	Uses the Qualcomm® Neural Processing SDK models
`tflite`	Uses the LiteRT models
`qnn`	Uses the Qualcomm® AI Engine direct models
yolo-model-type	
`yolov5` `yolov8` `yolonas`	Runs the YOLOv5, YOLOv8, and YOLO-NAS models, respectively. See Sample model and label files.
runtime	
`cpu`	Runs on the CPU
`gpu`	Runs on the GPU
`dsp`	Runs on the digital signal processor (DSP)
Input source	
`camera`	• 0 – Primary camera • 1 – Secondary camera
`file-path`	Directory path to the video file
`rtsp-ip-port`	Address of the RTSP stream in __ rtsp://<ip>:/<stream>__ format

4. To start the application, run the following command:

```
gst-ai-object-detection
```

- To stop the use case, select **CTRL + C**.
- To display the available help options, run the following command:

```
gst-ai-object-detection -h
```

- The `GST_DEBUG` environment variable controls the GStreamer debug output. Set the required level to allow logging. For example, to log all warnings, run the following command:

```
export GST_DEBUG=2
```

Parallel AI inference

The **gst-ai-parallel-inference** command-line application shows the hardware capability to perform four parallel AI inferences on input streams from a camera, video file, or RTSP stream. The pipeline detects objects, classifies objects, detects poses, and segments images on the input stream. The screen displays the results side-by-side.

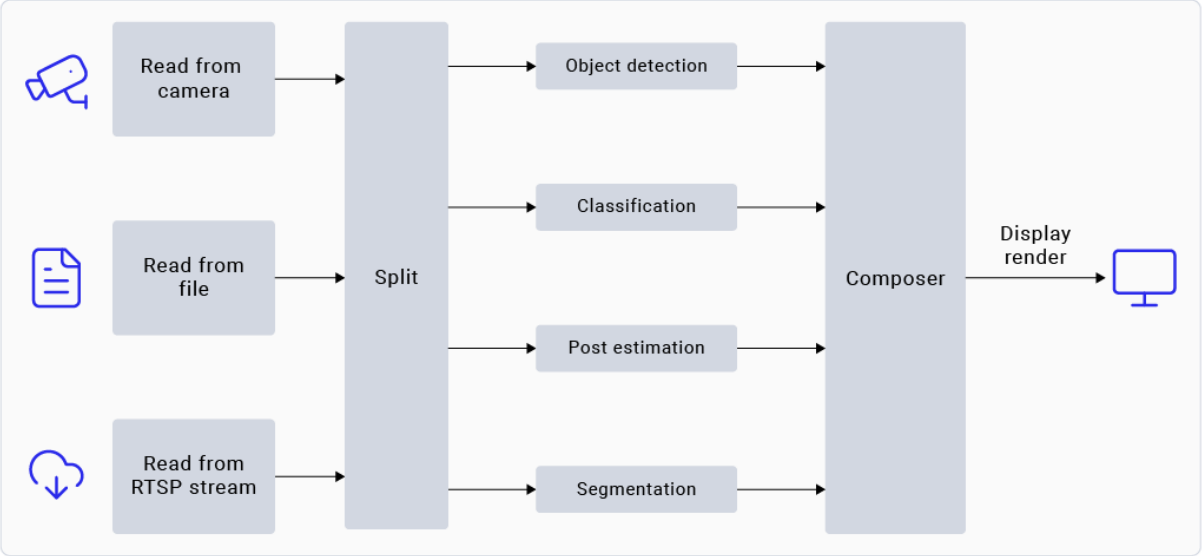
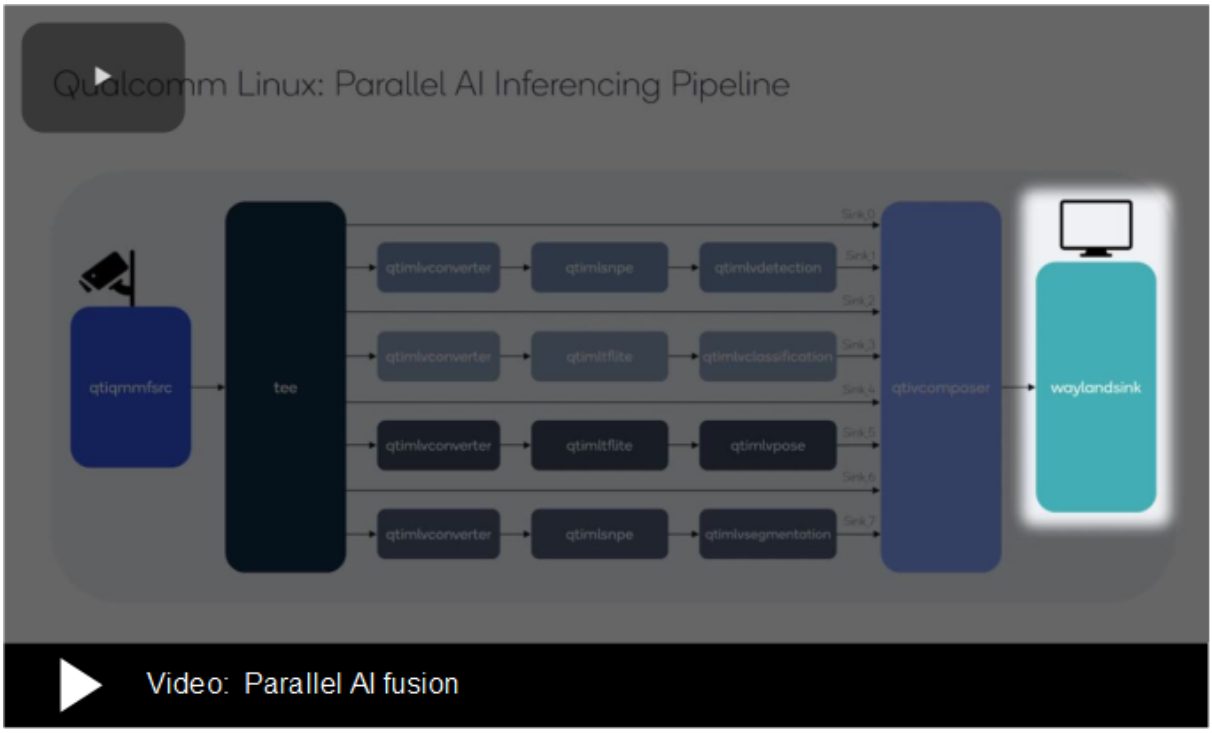


Figure: Parallel inference application workflow



Example usage

It's mandatory to push the model and label files to the device to run the sample application. For details, see Procedure.

1. Begin a new SSH session and start the HDMI display monitor if you haven't already:

```
ssh root@<ip-addr of the target device>
```

2. To view the sample application on the HDMI display, run the following export command from the SSH terminal:

```
export XDG_RUNTIME_DIR=/dev/socket/weston && export WAYLAND_
DISPLAY=wayland-1
```

Note: If Weston isn't enabled automatically, start two instances of secure shell: one to enable Weston and the other to run the application.

a. To enable Weston, run the following command in the first shell:

```
export GBM_BACKEND=msm && export XDG_RUNTIME_DIR=/dev/socket/
weston && mkdir -p $XDG_RUNTIME_DIR && weston --continue-
without-input --idle-time=0
```

b. To set up the Wayland Display environment, run the following command in the second shell:

```
export XDG_RUNTIME_DIR=/dev/socket/weston && export WAYLAND_
DISPLAY=wayland-1
```

---

3. To start the application, run the following command:

```
gst-ai-parallel-inference
```

- To stop the use case, select **CTRL + C**.
- To display the available help options, run the following command:

```
gst-ai-parallel-inference -h
```

Qualcomm AI Hub often updates models with the latest SDK versions. Using wrong model constants may lead to inaccurate results. If you face such issues, update the model constants. Provide the model constants for the sample application using the following command:

```
gst-ai-parallel-inference -s /etc/media/video.mp4 \
--object-detection-constants="YOLOv8,q-offsets=<21.0, 0.0, 0.0>,
q-scales=<3.0546178817749023, 0.003793874057009816, 1.0>;" \
--pose-detection-constants="Posenet,q-offsets=<8.0>,q-scales=<0.
0040499246679246426>;" \
--segmentation-constants="deeplab,q-offsets=<0.0>,q-scales=<1.0>
;" \
--classification-constants="Inceptionv3,q-offsets=<38.0>,q-
scales=<0.17039915919303894>;"
```

- The `GST_DEBUG` environment variable controls the GStreamer debug output. Set the required level to allow logging. For example, to log all warnings, run the following command:

```
export GST_DEBUG=2
```

Known issue

In pose detection, the model detects only one person, even if many people are present in the frame.

Note: Image classification using the Inception v3 model trains on the ImageNet data set. As a result, the model can't detect a person because this class isn't included in the data set.

3.4 More applications

The Qualcomm Linux release offers various sample applications. To explore more, see Sample applications.

4 Develop your own application

You can create applications using the Qualcomm® Visual Studio Code Extension.

For more information, see Qualcomm Visual Studio Code Extension Reference Guide.

5 Update the software

This section describes how to download and flash the prebuilt images onto the Qualcomm® RB3 Gen 2 device from different host computers running Ubuntu, Windows, or mac operating systems.

Caution: Updating the software with the prebuilt images erases all data on your device and installs a newer image. Ensure to back up important data before you start.

You can also perform an incremental OTA update. For detailed procedure on how to create OSTree and capsule update mechanisms, see OTA update for Qualcomm Linux.

Ubuntu

5.1 Download the software

The RB3 Gen 2 device supports the following SDKs:

- **Qualcomm Intelligent Multimedia Product (QIMP) SDK**: This is the recommended software for the RB3 Gen 2 device. The software comes with multimedia applications.
- **Qualcomm Intelligent Robotics Product (QIRP) SDK**: This version of the RB3 Gen 2 software includes robotic applications.

You can download these SDKs on either an x86 or Arm-based host computer using the following links:

Core Kit	• **x86**: https://artifacts.codelinaro.org/artifactory/qli-ci/flashable-binaries/qimpsdk/qcs6490-rb3gen2-core-kit/x86-qcom-6.6.65-QLI.1.4-Ver.1.1_qim-product-sdk-1.1.2.zip • **Arm**: https://artifacts.codelinaro.org/artifactory/qli-ci/flashable-binaries/qimpsdk/qcs6490-rb3gen2-core-kit/arm-qcom-6.6.65-QLI.1.4-Ver.1.1_qim-product-sdk-1.1.2.zip
Vision Kit	• **x86**: https://artifacts.codelinaro.org/artifactory/qli-ci/flashable-binaries/qimpsdk/qcs6490-rb3gen2-vision-kit/x86-qcom-6.6.65-QLI.1.4-Ver.1.1_qim-product-sdk-1.1.2.zip • **Arm**: https://artifacts.codelinaro.org/artifactory/qli-ci/flashable-binaries/qimpsdk/qcs6490-rb3gen2-vision-kit/arm-qcom-6.6.65-QLI.1.4-Ver.1.1_qim-product-sdk-1.1.2.zip

This document describes the download procedure for an x86-based host computer. If you are using an Arm-based host computer, update the commands with the appropriate download links.

Note: Ensure that you've enough free storage on the host computer before downloading the artifacts zip file, because unzipping it requires about 19 GB of space.

Download QIMP SDK

Core Kit

1. To download the QIMP SDK, run the following command:

```
wget https://artifacts.codelinaro.org/artifactory/qli-ci/
flashable-binaries/qimpsdk/qcs6490-rb3gen2-core-kit/x86-qcom-
6.6.65-QLI.1.4-Ver.1.1_qim-product-sdk-1.1.2.zip
```

2. To unzip the downloaded file on the development host, run the following command:

```
unzip x86-qcom-6.6.65-QLI.1.4-Ver.1.1_qim-product-sdk-1.1.2.
zip
```

Vision Kit

1. To download the QIMP SDK, run the following command:

```
wget https://artifacts.codelinaro.org/artifactory/qli-ci/
flashable-binaries/qimpsdk/qcs6490-rb3gen2-vision-kit/x86-
qcom-6.6.65-QLI.1.4-Ver.1.1_qim-product-sdk-1.1.2.zip
```

2. To unzip the downloaded file on the development host, run the following command:

```
unzip x86-qcom-6.6.65-QLI.1.4-Ver.1.1_qim-product-sdk-1.1.2.
zip
```

Download QIRP SDK

Vision Kit

1. To download the QIRP SDK, run the following command:

```
wget https://artifacts.codelinaro.org/artifactory/qli-ci/
flashable-binaries/qirpsdk/qcs6490-rb3gen2-vision-kit/x86-
qcom-6.6.65-QLI.1.4-Ver.1.1_robotics-product-sdk-1.1.zip
```

2. To unzip the downloaded file on the development host, run the following command:

```
unzip x86-qcom-6.6.65-QLI.1.4-Ver.1.1_robotics-product-sdk-1.
1.zip
```

5.2 Configure prerequisites

Ensure that the device is in emergency download (EDL) mode, and then proceed to update the `udev` rules.

Move to EDL mode

The device enters emergency download (EDL) mode under two conditions:

- If the device doesn't have installed software upon initial power-up
- If the currently installed software malfunctions

To facilitate software flashing, you must force the device into EDL mode using one of the following methods.

Using ADB shell

To force the device into EDL mode, run the following command:

```
adb shell reboot edl
```

For information about how to install ADB, see FAQs.

Using F_DL button

1. Connect the device to the host computer through the USB Type-C connector.
2. Hold down the **F_DL** button.

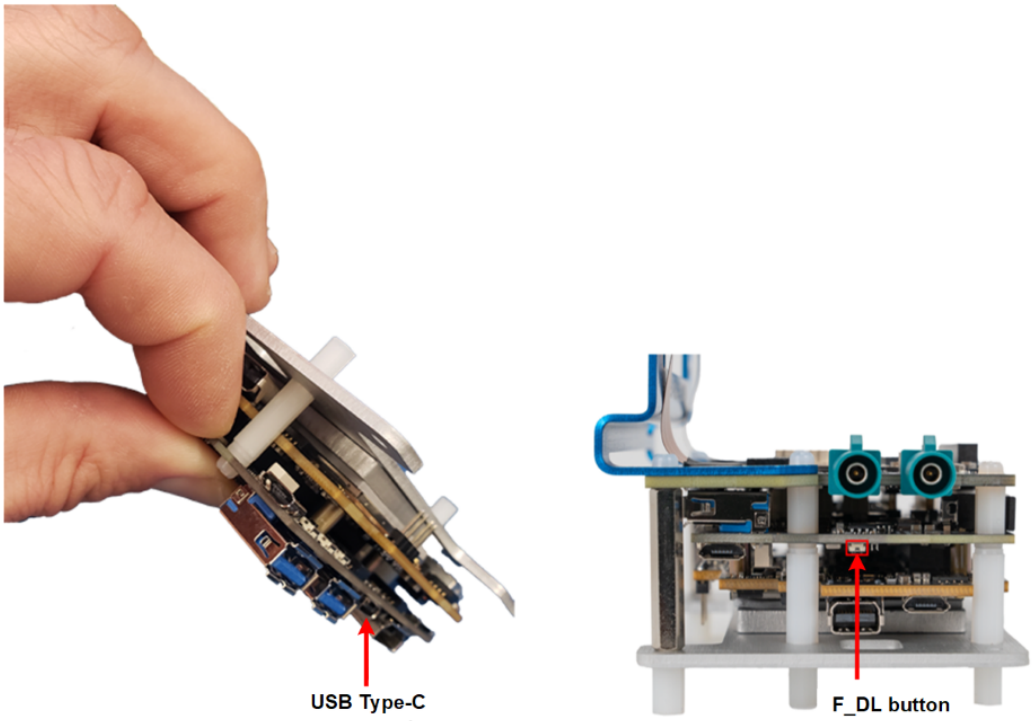


Figure: Location of the USB Type-C port and F_DL button

3. Connect the device to a 12-V wall power supply.
4. Release the **F_DL** button. The device should now be in EDL mode.
5. To verify whether the device has entered EDL mode, run the following command on your host computer:

```
lsusb
```

Sample output

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

Update udev rules

Udev rules ensure consistent device recognition and naming, facilitate seamless hardware changes, and allow dynamic device management.

Ensure that you configure the `udev` USB rules for the Qualcomm manufacturing vendor ID **05c6** on the Ubuntu host computer. Run the following commands on your Ubuntu host computer:

1. To go to the directory, run the following command:

```
cd /etc/udev/rules.d
```

2. To list the contents of the directory, run the following command:

```
ls
```

- If the `51-qcom-usb.rules` file isn't present, use `sudo vi 51-qcom-usb.rules` to create it, and add the following content to the file:

```
SUBSYSTEMS=="usb", ATTRS{idVendor}=="05c6", ATTRS{idProduct}
=="9008", MODE="0664", GROUP="plugdev"
```

- If the file exists, then check for the earlier content. Use the following command to view the content:

```
cat 51-qcom-usb.rules
```

3. To restart `udev`, run the following command:

```
sudo systemctl restart udev
```

If the USB cable is already connected to the host computer, unplug the cable and then reconnect it for the updated rules to take effect.

5.3 Flash using the QDL tool

The Qualcomm Download (QDL) tool is a flashing utility that communicates with USB devices identified by ID 05c6:9008. The tool uploads a flash loader to the device, which then flashes the images to the universal flash storage (UFS) built into the device.

1. Download the QDL tool and unzip the contents of the downloaded folder. Qualcomm Linux 1.4 requires QDL version 2.3.1 or higher.
2. To provide executable permission, run the following command:

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

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

Important: If your device uses Qualcomm Linux 1.0, 1.1, or 1.2, you must perform a one-time UFS provisioning and CDT flashing to upgrade to Qualcomm Linux 1.3 or later due to a change in the UFS partition table.

---

3. To flash the images, run the following commands:

   ```
   cd <extracted zip directory path>/target/qcs6490-rb3gen2-
   vision-kit/qcom-multimedia-image
   ```

   Note:

   - If flashing fails, see How to troubleshoot flashing-related issues for further help.
   - Replace the `<extracted zip directory path>` with the extracted `x86-qcom-6.6.65-QLI.1.4-Ver.1.1_qim-product-sdk-1.1.2.zip`

file path.

- For the Core Kit, replace `qcs6490-rb3gen2-vision-kit` with `qcs6490-rb3gen2-core-kit` in the command.

```
<qdl_tool_path>/qdl_2.3.1/QDL_Linux_x64/qdl prog_
firehose_ddr.elf rawprogram*.xml patch*.xml
```

Run the `lsusb` command after the device successfully reboots following a flashing operation to view the device information in the terminal window, as shown in line 4 of the following message:

```
# Sample output for QCS6490
Bus 002 Device 003: ID 05c6:9135 Qualcomm, Inc.
qcs6490-rb3gen2-vision-kit
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Note: If you've performed UFS provisioning, proceed to Update USB and Ethernet controller firmware.

Windows

5.4 Download the software

The RB3 Gen 2 device supports the following SDKs:

•

Qualcomm Intelligent Multimedia Product (QIMP) SDK: This is the recommended software for the RB3 Gen 2 device. The software comes with multimedia applications.

- **Qualcomm Intelligent Robotics Product (QIRP) SDK**: This version of the RB3 Gen 2 software includes robotic applications.

You can download these SDKs on either an x86 or Arm-based host computer using the following links:

Core Kit	• **x86**: https://artifacts.codelinaro.org/artifactory/qli-ci/flashable-binaries/qimpsdk/qcs6490-rb3gen2-core-kit/x86-qcom-6.6.65-QLI.1.4-Ver.1.1_qim-product-sdk-1.1.2.zip • **Arm**: https://artifacts.codelinaro.org/artifactory/qli-ci/flashable-binaries/qimpsdk/qcs6490-rb3gen2-core-kit/arm-qcom-6.6.65-QLI.1.4-Ver.1.1_qim-product-sdk-1.1.2.zip
Vision Kit	• **x86**: https://artifacts.codelinaro.org/artifactory/qli-ci/flashable-binaries/qimpsdk/qcs6490-rb3gen2-vision-kit/x86-qcom-6.6.65-QLI.1.4-Ver.1.1_qim-product-sdk-1.1.2.zip • **Arm**: https://artifacts.codelinaro.org/artifactory/qli-ci/flashable-binaries/qimpsdk/qcs6490-rb3gen2-vision-kit/arm-qcom-6.6.65-QLI.1.4-Ver.1.1_qim-product-sdk-1.1.2.zip

This document describes the download procedure for an x86-based host computer. If you are using an Arm-based host computer, update the commands with the appropriate download links.

Note: Ensure that you've enough free storage on the host computer before downloading the artifacts zip file, because unzipping it requires about 19 GB of space.

Download QIMP SDK

Core Kit

1. To download the QIMP SDK, paste the following link into a browser:

```
https://artifacts.codelinaro.org/artifactory/qli-ci/
flashable-binaries/qimpsdk/qcs6490-rb3gen2-core-kit/x86-qcom-
6.6.65-QLI.1.4-Ver.1.1_qim-product-sdk-1.1.2.zip
```

2. Unzip the downloaded file on the development host.

Vision Kit

1. To download the QIMP SDK, paste the following link into a browser:

```
https://artifacts.codelinaro.org/artifactory/qli-ci/
flashable-binaries/qimpsdk/qcs6490-rb3gen2-vision-kit/x86-
qcom-6.6.65-QLI.1.4-Ver.1.1_qim-product-sdk-1.1.2.zip
```

2. Unzip the downloaded file on the development host.

Download QIRP SDK

Vision Kit

1. To download the QIRP SDK, paste the following link into a browser:

   ```
   https://artifacts.codelinaro.org/artifactory/qli-ci/
   flashable-binaries/qirpsdk/qcs6490-rb3gen2-vision-kit/x86-
   qcom-6.6.65-QLI.1.4-Ver.1.1_robotics-product-sdk-1.1.zip
   ```

2. Unzip the downloaded file on the development host.

5.5 Configure prerequisite

Ensure that the device is in emergency download (EDL) mode.

Move to EDL mode

The device enters emergency download (EDL) mode under two conditions:

- If the device doesn't have installed software upon initial power-up
- If the currently installed software malfunctions

To facilitate software flashing, you must force the device into EDL mode using one of the following methods.

Using ADB shell

To force the device into EDL mode, run the following command:

```
adb shell reboot edl
```

For information about how to install ADB, see FAQs.

Using F_DL button

1. Connect the device to the host computer through the USB Type-C connector.
2. Hold down the **F_DL** button.

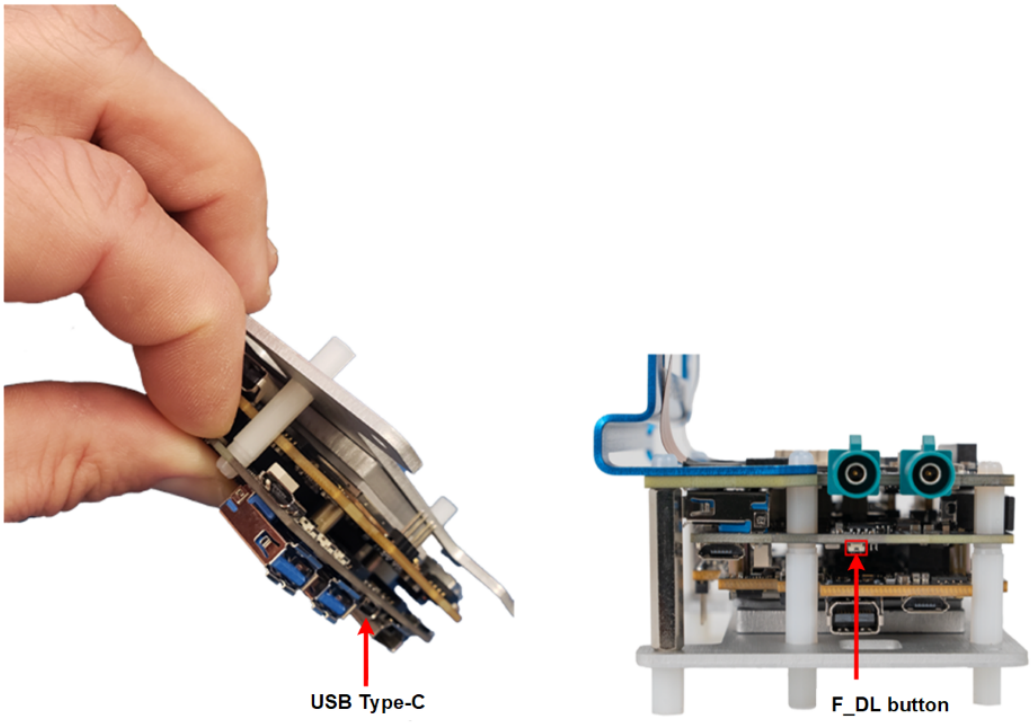


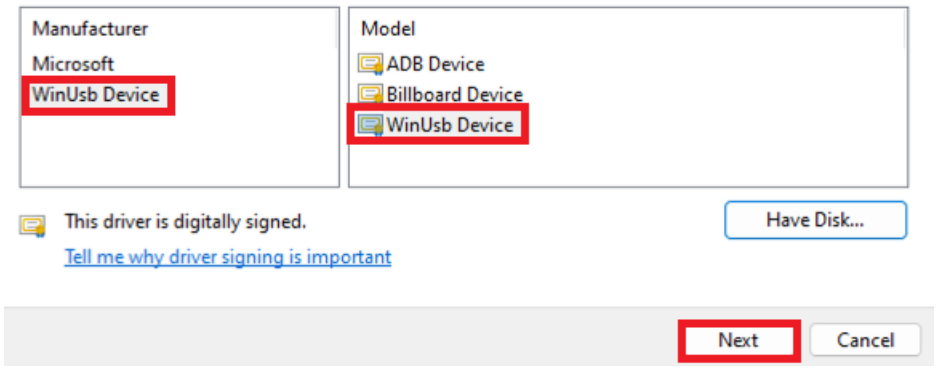
Figure: Location of the USB Type-C port and F_DL button

3. Connect the device to a 12-V wall power supply.
4. Release the **F_DL** button.

5.6 Flash using the QDL tool

The Qualcomm Download (QDL) tool is a flashing utility that communicates with USB devices identified by ID 05c6:9008. The tool uploads a flash loader to the device, which then flashes the images to the universal flash storage (UFS) built into the device.

1. Install Microsoft WinUSB.
    a. Uninstall any other drivers for the device. Ensure that drivers like Qualcomm USB driver aren't installed. The device shouldn't appear under **COM Ports** in the **Device Manager**.
    b. Plug your device into the host computer.
    c. Open **Device Manager** and locate the device.
    d. Right-click the device and select **Update driver software...** from the context menu.
    e. In the wizard, select **Browse my computer for drivers**.
    f. Select **Let me pick from a list of device drivers on my computer**.
    g. From the list of device classes, select **Universal Serial Bus devices**.
    h. The wizard displays **WinUsb Device**. Select it to load the driver.



2. Download the QDL tool and unzip the contents of the downloaded folder. Qualcomm Linux 1.4 requires QDL version 2.3.1 or higher.
3. Copy the `QDL.exe` and `libusb-1.0.dll` from `\qdl_2.3.1\QDL_Win_x64` to the `<extracted zip directory path>\target\qcs6490-rb3gen2-vision-kit\qcom-multimedia-image` directory.
4. To flash the images, run the following command:

Important: If your device uses Qualcomm Linux 1.0, 1.1, or 1.2, you must perform a one-time UFS provisioning and CDT flashing to upgrade to Qualcomm Linux 1.3 or later due to a change in the UFS partition table.

```
.\QDL.exe prog_firehose_ddr.elf rawprogram0.xml rawprogram1.
xml rawprogram2.xml rawprogram3.xml rawprogram4.xml
rawprogram5.xml patch0.xml patch1.xml patch2.xml patch3.xml
patch4.xml patch5.xml
```

Note:

- Wildcards aren't supported for program filenames. List each image file in the command.
- If flashing fails, see How to troubleshoot flashing-related issues for further help.
- Replace the `<extracted zip directory path>` with the extracted `x86-qcom-6.6.65-QLI.1.4-Ver.1.1_qim-product-sdk-1.1.2.zip` file path.
- For the Core Kit, replace `qcs6490-rb3gen2-vision-kit` with `qcs6490-rb3gen2-core-kit` in the command.

Note: If you've performed UFS provisioning, proceed to Update USB and Ethernet controller firmware.

---

macOS

5.7 Set up a virtual machine (VM)

To update the software on the RB3 Gen 2 device using a Mac host computer, you must set up a VM running the Ubuntu OS and perform the following procedures from the Ubuntu VM. To set up the VM environment, see Virtual Machine Setup Guide.

5.8 Download the software

The RB3 Gen 2 device supports the following SDKs:

- **Qualcomm Intelligent Multimedia Product (QIMP) SDK**: This is the recommended software for the RB3 Gen 2 device. The software comes with multimedia applications.
- **Qualcomm Intelligent Robotics Product (QIRP) SDK**: This version of the RB3 Gen 2 software includes robotic applications.

Note: Ensure that you've enough free storage on the host computer before downloading the artifacts zip file, because unzipping it requires about 19 GB of space.

Download QIMP SDK

Core Kit

1. To download the QIMP SDK, run the following command:

```
wget https://artifacts.codelinaro.org/artifactory/qli-ci/
flashable-binaries/qimpsdk/qcs6490-rb3gen2-core-kit/arm-qcom-
6.6.65-QLI.1.4-Ver.1.1_qim-product-sdk-1.1.2.zip
```

2. To unzip the downloaded file on the development host, run the following command:

```
unzip arm-qcom-6.6.65-QLI.1.4-Ver.1.1_qim-product-sdk-1.1.2.
zip
```

Vision Kit

1. To download the QIMP SDK, run the following command:

```
wget https://artifacts.codelinaro.org/artifactory/qli-ci/
flashable-binaries/qimpsdk/qcs6490-rb3gen2-vision-kit/arm-
qcom-6.6.65-QLI.1.4-Ver.1.1_qim-product-sdk-1.1.2.zip
```

2. To unzip the downloaded file on the development host, run the following command:

```
unzip arm-qcom-6.6.65-QLI.1.4-Ver.1.1_qim-product-sdk-1.1.2.
zip
```

Download QIRP SDK

Vision Kit

1. To download the QIRP SDK, run the following command:

```
wget https://artifacts.codelinaro.org/artifactory/qli-ci/
flashable-binaries/qirpsdk/qcs6490-rb3gen2-vision-kit/arm-
qcom-6.6.65-QLI.1.4-Ver.1.1_robotics-product-sdk-1.1.zip
```

2. To unzip the downloaded file on the development host, run the following command:

```
unzip arm-qcom-6.6.65-QLI.1.4-Ver.1.1_robotics-product-sdk-1.
1.zip
```

5.9 Configure prerequisites

After you set up the VM environment, ensure that the device is in emergency download (EDL) mode, and then proceed to update the `udev` rules.

Move to EDL mode

The device enters emergency download (EDL) mode under two conditions:

- If the device doesn't have installed software upon initial power-up
- If the currently installed software malfunctions

To facilitate software flashing, you must force the device into EDL mode using one of the following methods.

Using ADB shell

To force the device into EDL mode, run the following command:

```
adb shell reboot edl
```

For information about how to install ADB, see FAQs.

Using F_DL button

1. Connect the device to the host computer through the USB Type-C connector.

2. Hold down the **F_DL** button.

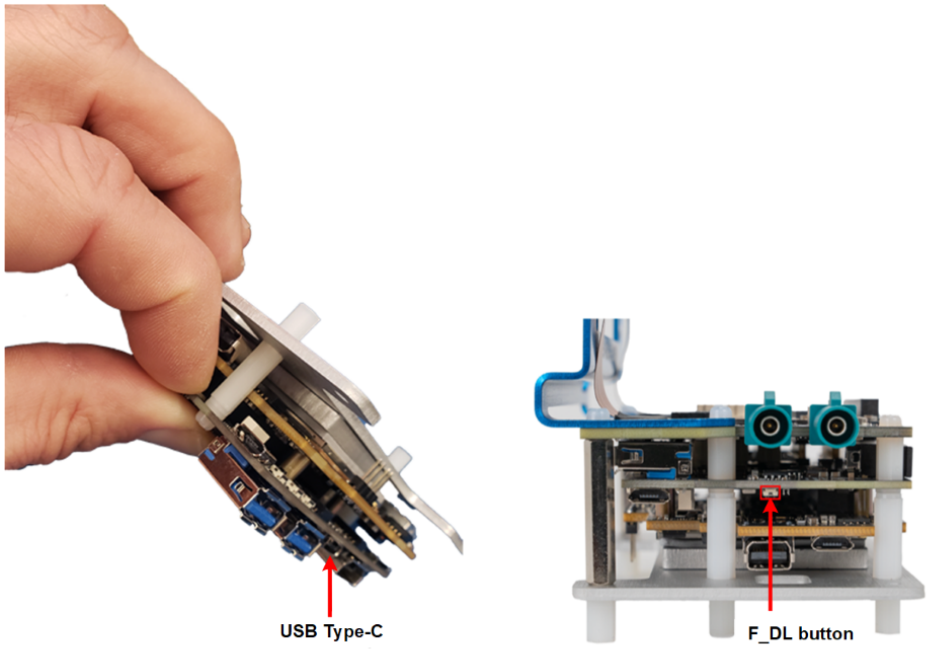


Figure: Location of the USB Type-C port and F_DL button

3. Connect the device to a 12-V wall power supply.
4. Release the **F_DL** button. The device should now be in EDL mode.
5. To verify whether the device has entered EDL mode, run the following command on your host computer:

```
lsusb
```

Sample output

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

Update udev rules

Udev rules ensure consistent device recognition and naming, facilitate seamless hardware changes, and allow dynamic device management.

Ensure that you configure the `udev` USB rules for the Qualcomm manufacturing vendor ID **05c6** on the Ubuntu VM. Run the following commands on the Ubuntu VM:

1. To go to the directory, run the following command:

```
cd /etc/udev/rules.d
```

2. To list the contents of the directory, run the following command:

```
ls
```

   - If the `51-qcom-usb.rules` file isn't present, use `sudo vi 51-qcom-usb.rules` to create it, and add the following content to the file:

```
SUBSYSTEMS=="usb", ATTRS{idVendor}=="05c6", ATTRS{idProduct}
=="9008", MODE="0664", GROUP="plugdev"
```

   - If the file exists, then check for the earlier content. Use the following command to view the content:

```
cat 51-qcom-usb.rules
```

3. To restart `udev`, run the following command:

```
sudo systemctl restart udev
```

If the USB cable is already connected to the host computer, unplug the cable and then reconnect it for the updated rules to take effect.

5.10 Flash using the QDL tool

The Qualcomm Download (QDL) tool is a flashing utility that communicates with USB devices identified by ID 05c6:9008. The tool uploads a flash loader to the device, which then flashes the images to the universal flash storage (UFS) built into the device.

1. Download the QDL tool and unzip the contents of the downloaded folder. Qualcomm

Linux 1.4 requires QDL version 2.3.1 or higher.

2. To provide executable permission, run the following command:

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

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

Important: If your device uses Qualcomm Linux 1.0, 1.1, or 1.2, you must perform a one-time UFS provisioning and CDT flashing to upgrade to Qualcomm Linux 1.3 or later due to a change in the UFS partition table.

---

3. To flash the images, run the following command:

```
cd <extracted zip directory path>/target/qcs6490-rb3gen2-
vision-kit/qcom-multimedia-image
```

Note:

- If flashing fails, see How to troubleshoot flashing-related issues for further help.
- Replace the `<extracted zip directory path>` with the extracted `x86-qcom-6.6.65-QLI.1.4-Ver.1.1_qim-product-sdk-1.1.2.zip` file path.
- For the Core Kit, replace `qcs6490-rb3gen2-vision-kit` with `qcs6490-rb3gen2-core-kit` in the command.

```
<qdl_tool_path>/qdl_2.3.1/QDL_Linux_x64/qdl prog_
firehose_ddr.elf rawprogram*.xml patch*.xml
```

Run the `lsusb` command after the device successfully reboots following a flashing operation to view the device information in the terminal window, as shown in line 4 of the following message:

```
# Sample output for QCS6490
Bus 002 Device 003: ID 05c6:9135 Qualcomm, Inc.
qcs6490-rb3gen2-vision-kit
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Note: If you've performed UFS provisioning, proceed to Update USB and Ethernet controller firmware.

5.11 Next steps

1. Set up the device
2. Run sample applications

6 FAQs

How do you recover your device?

1. Download the software.
2. Force the device to emergency download (EDL) mode.
3. Flash the downloaded software.

For detailed instructions, see Ubuntu, Windows, or macOS, based on your OS.

How do you flash the configuration data table (CDT)?

CDT provides platform or device-dependent data such as platform ID, subtype, and version. Various software (drivers/firmware) modules can use this information to dynamically detect and initialize the platform.

Note: Ensure that the device is in EDL mode. For more information, see Move to EDL mode.

---

1. Download the CDT binary.

Based on the required development kit, download the respective CDT from the **Flash CDT** table in the Release Notes.

```
mkdir <cdt_download_path>
```

```
cd <cdt_download_path>
```

```
wget https://artifacts.codelinaro.org/artifactory/codelinaro-le/
Qualcomm_Linux/QCS6490/cdt/rb3gen2-vision-kit.zip
```

Note: The download link provided is for the Vision Kit. For information on other development kit

CDTs, see Release Notes.

```
unzip rb3gen2-vision-kit.zip
```

2. To flash the CDT, run the following commands:

```
cd <cdt_download_path>
```

```
<qdl_tool_path>/qdl_2.3.1/QDL_Linux_x64/qdl prog_firehose_ddr.elf
rawprogram3.xml patch3.xml
```

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.

Note:

- To download the provision XML file and check the applicability of UFS provisioning for different SoCs, see the **UFS Provisioning** table in the Release Notes.
- Ensure that the device is in EDL mode. For more information, see Move to EDL mode.

1. To download the provision file, do the following:

Based on the required SoC, download the respective *provision* from the **UFS Provisioning** table in the Release Notes.

```
mkdir <provision_download_path>
```

```
cd <provision_download_path>
```

```
wget https://artifacts.codelinaro.org/artifactory/codelinaro-le/
Qualcomm_Linux/QCS6490/provision.zip
```

```
unzip provision.zip
```

2. To provision UFS, run the following commands:

```
cd <provision_download_path>
```

```
<qdl_tool_path>/qdl_2.3.1/QDL_Linux_x64/qdl --storage ufs prog_
firehose_ddr.elf <Provision file>
```

Example

<qdl_tool_path>/qdl_2.3.1/QDL_Linux_x64/qdl --storage ufs prog_firehose_ddr.elf provision_1_3.xml

How to put the device into the EDL mode without using the DIP switches?

Run the following command from the ADB shell:

```
adb shell reboot edl
```

How do you update the USB and Ethernet controller firmware?

A firmware update to the USB controller may solve USB or Ethernet connectivity issues on the RB3 Gen 2 device.

Prerequisites

- Ensure that you upgrade the software as described in Update the software before updating the Renesas firmware.
- Ensure that you install the ADB tool. Fastboot comes installed along with it.
- Connect the device to the Ubuntu host computer using a USB Type-C cable.

Procedure

Note: This procedure is applicable only to an Ubuntu 22.04 host computer. If you're using a Windows or macOS host computer, set up an Ubuntu virtual machine by following the instructions described in the Virtual Machine Setup Guide.

1. To download the firmware, register and sign in to the Renesas website. After signing in, you can access the firmware from the following link.
2. To create the `usb_fw.img` image and copy the USB firmware, run the following commands on the Linux host computer:

```
dd if=/dev/zero of=usb_fw.img bs=4k count=240
```

```
mkfs -t ext4 usb_fw.img
```

```
mkdir usb_fw
```

```
sudo mount -o loop usb_fw.img usb_fw/
```

```
sudo cp -rf renesas_usb_fw.mem usb_fw
```

```
sudo umount usb_fw
```

3. To start the device in Fastboot mode, run the following commands:

```
adb root
```

```
adb shell
```

```
reboot bootloader
```

4. When the device is in Fastboot mode, run the following command:

```
fastboot devices
```

Sample output

```
7dc85f5e         fastboot
```

5. To flash the `usb_fw.img` image to the device, run the following commands:

```
fastboot erase usb_fw
```

```
fastboot flash usb_fw usb_fw.img
```

```
fastboot reboot
```

6. To confirm that the firmware update was successful, run the following command:

```
adb shell dmesg
```

Sample output

```
[    6.589462] usbcore: registered new device driver onboard-usb-hub
[    6.653277] usb 2-1: new SuperSpeed USB device number 2 using
xhci_hcd
[    7.013061] usb 2-1.1: new SuperSpeed USB device number 3 using
xhci_hcd
[    7.120657] ax88179_178a 2-1.1:1.0 eth0: register 'ax88179_178a'
at usb-0001:04:00.0-1.1, ASIX AX88179 USB 3.0 Gigabit Ethernet, 3e:
9e:5e:ff:d3:fb
[    7.120767] usbcore: registered new interface driver ax88179_178a
```

How do you install the Qualcomm USB driver (QUD) on the host computer?

Install QUD on the host computer to detect the RB3 Gen 2 device.

Note: For Ubuntu 22.04, you may face an issue while installing QUD. To ensure a successful installation, you may need to enroll the public key on your Linux host computer. For more details, follow the steps provided in the README file available in the `/opt/QUIC/sign/signReadme.txt` directory.

1. To clone the GitHub repository, run the following commands:

```
sudo apt install git
```

```
sudo git clone https://github.com/quic/quic-usb-drivers.git
```

```
cd quic-usb-drivers
```

2. To provide executable permissions to the file, run the following command:

```
sudo chmod +x QcDevDriver.sh
```

3. To install the QUD driver, run the following command:

```
sudo ./QcDevDriver.sh install
```

4. To detect the device, run the following command:

```
lsusb
```

Sample output

```
Bus 002 Device 099: ID 05c6:901d Qualcomm, Inc. QCM6490_e05d8b80
```

How do you install and connect the android debug bridge (ADB) on the host computer?

Some Qualcomm proprietary tools require an ADB. ADB is a command-line tool that allows communication with the RB3 Gen 2 device. The `adb` command facilitates various device actions, such as debugging, capturing kernel logs, and pushing applications and files to the device.

Install ADB

1. Install QUD.
2. Run the following command on the Linux host computer:

```
sudo apt install git android-tools-adb android-tools-fastboot wget
```

3. To verify the ADB version on the Linux host computer, run the following command:

```
adb --version
```

Sample output

```
Android Debug Bridge version 1.0.39
Version 1:8.1.0+r23-5ubuntu2
```

Connect to ADB

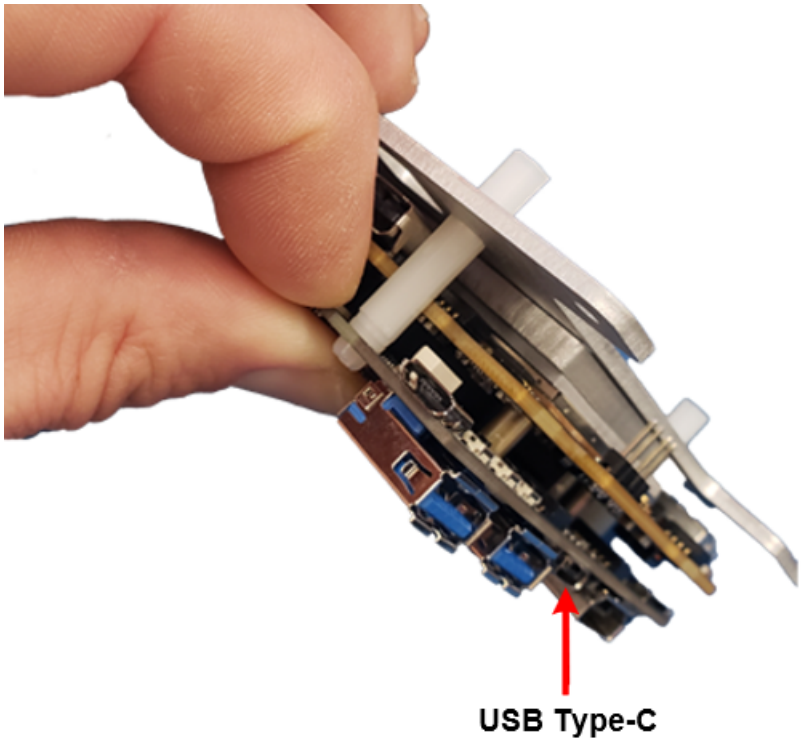


Figure: Location of the USB Type-C port

4. Using the USB Type-C cable, connect the RB3 Gen 2 device with the host computer. Ensure that the device displays as an ADB device.

```
adb devices
```

Sample output

```
List of devices attached
68f592a device
```

5. To enter the shell, run the following command:

```
adb shell
```

Sample output

```
#
```

How do you connect the Vision mezzanine to the Core Kit?

The Vision mezzanine supports interfacing cameras to the RB3 Gen 2 device through two, second-generation gigabit multimedia serial link (GMSL2) camera inputs and five CAM/CSI camera ports. This mezzanine also provides four DMICs, an IMU, compass/magnetometer, and a pressure sensor.

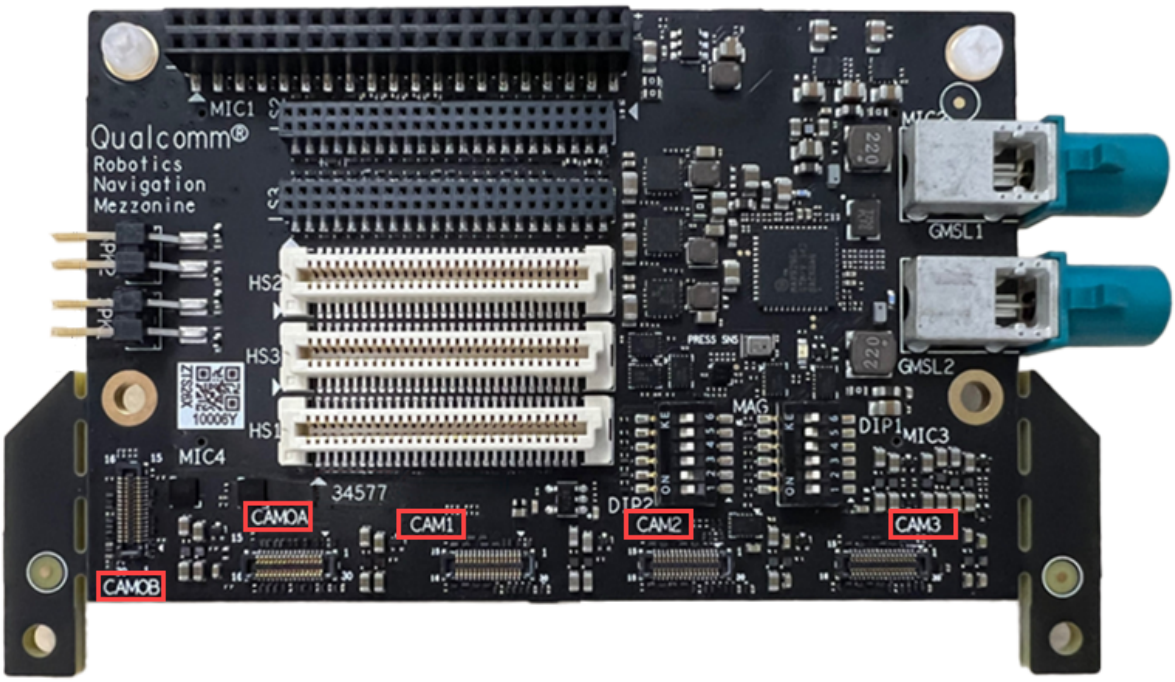


Figure: Vision mezzanine

For more information, see Vision mezzanine.

Follow these steps to connect the Vision mezzanine to the Core Kit. Flash the CDT to update the device configuration based on the new mezzanine. It's essential to follow these steps because the Core and Vision kits have different CDTs and software.

1. Flash the Vision Kit CDT, see Flash CDT.
2. Download and flash the Vision Kit software for your respective host computer, see Update the software.

How do you troubleshoot flashing-related issues?

Note: Ensure that the host computer isn't running the ModemManager tool, as it interferes with QDL flashing.

If you're using a Linux distribution with systemd, stop the ModemManager tool using the following command:

```
sudo systemctl stop ModemManager
```

- If you need ModemManager, restart it after the flashing is complete.

If flashing fails, do the following and retry the flashing procedure:

1. Power off the device.
2. Disconnect the device from the host computer.
3. Restart the host computer.

How do you reconnect to a different network when you're already connected to one?

1. To disconnect from the current network, run the following command:

```
nmcli c down QualcommWiFi
```

```
Connection 'QualcommWiFi' successfully deactivated (D-Bus
active path: /org/freedesktop/NetworkManager/
ActiveConnection/1)
```

2. To verify the disconnect status, run the following command:

```
nmcli -p device
```

```
root@qcs6490-rb3gen2-vision-kit:~# nmcli -p device
=====================
  Status of devices
=====================
DEVICE  TYPE      STATE         CONNECTION
--------------------------------------------------------------------
wlan0   wifi      disconnected  --
eth0    ethernet  unavailable   --
eth1    ethernet  unavailable   --
can0    can       unmanaged     --
lo      loopback  unmanaged     --
```

3. To connect to a different Wi-Fi network, run the following command:

```
nmcli dev wifi connect QualcommAP password XXXXXXXXX
```

```
Device 'wlan0' successfully activated with
'6159ac7c-58c2-44fa-938f-45dcb544fac3'.
```

How do you set up debug UART using Minicom?

Minicom is a serial communication program that allows an admin user to interact with the RB3 Gen 2 device through the serial port.

1. To install Minicom on the Linux host computer, run the following commands:

```
sudo apt update
```

```
sudo apt install minicom
```

2. To verify the USB port, run the following command:

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

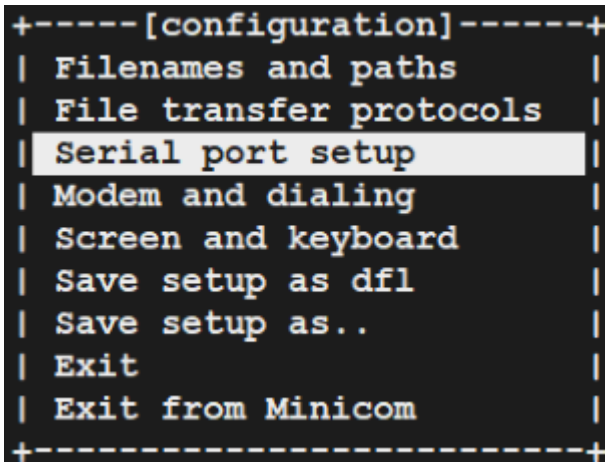
Sample output

```
/dev/ttyUSB0
```

3. To open Minicom, run the following command:

```
sudo minicom -s
```

4. Select the **Down** arrow key to select the **Serial port setup** option.

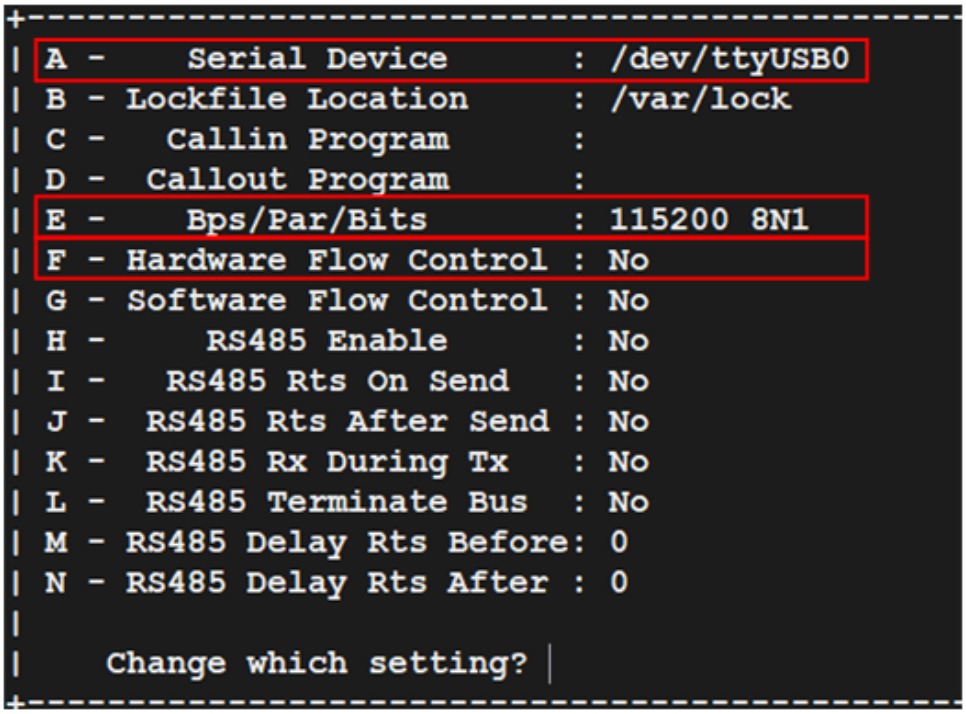


5. To set up the serial device configuration, do the following:
   a. Select **A** on your keyboard to set up the serial device name such as `/dev/ttyUSB0`.
   b. Select **Enter** to save the changes.
   c. Select **E** on your keyboard to set the baud rate. If the baud rate isn't set to **115200**, select the **E** key again. Select **Q** on your keyboard to set the configuration to **8N1**.

d. Select **Enter** to save the changes.

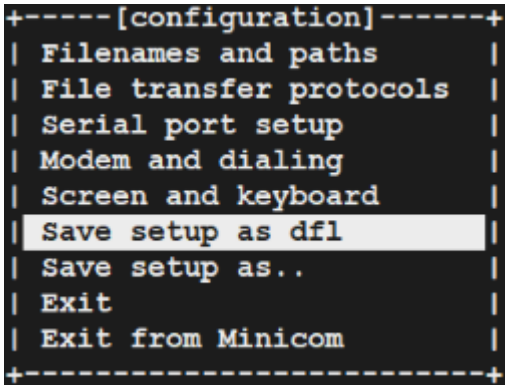
e. Select **F** on your keyboard to set the **Hardware Flow Control** to **No**.

Note: Ensure that the letters A, E, Q, and F are in uppercase.



f. Select **Enter** to save the changes.

6. Select the **Save setup as dfl** option and select **Enter**.



7. Select **Exit** to open the UART console.

8. Open the serial device and select **ENTER** to bring up the authentication prompt.

9. To sign in to the UART console, specify the following:
   - **Sign in**: `root`
   - **Password**: `oelinux123`

Note: If the authentication console doesn't display as expected, verify the USB connection. If necessary, disconnect and then reconnect the Micro USB.

How do you change the default root password?

By default, root users can sign in using the `oelinux123` password.

1. Sign in as a `root` user.
2. To change the `root` password, run the following command:

```
passwd
```

Note: This command prompts you to enter a new password.

3. Sign out using the `exit` command.
4. Sign in as `root` using your new password.

How do you create a non-root user account?

Note: By default, you can sign in as the `root` user.

1. To create a non-root user account, run the following command in the shell.

```
adduser guest
```

Note: This command prompts you to enter a new password.

2. Sign out using the `exit` command.
3. Sign in to the shell using your new user account credentials.

6.1 Further support

Ask your questions on the Qualcomm support forum.

7 Explore further

To delve deeper, explore the following documents:

Document	Description
Software Build Guide	Describes methods for synchronizing, building, and flashing Qualcomm Linux and the associated firmware on supported devices
Yocto Guide	Explains how the Yocto framework builds an embedded OS
Qualcomm Intelligent Multimedia Product (QIMP) SDK	Provides a unified environment for developing AI/ML and multimedia use cases
Qualcomm® Intelligent Multimedia SDK (IM SDK)	Allows seamless AI/ML and multimedia application deployment across Internet of Things (IoT) segments
Qualcomm Intelligent Robotics Product (QIRP) SDK	Provides an environment for developing applications for general robotics use cases
Qualcomm AI Engine direct SDK	Provides a unified API and modular per accelerator libraries for full-stack AI solutions
Qualcomm® Neural Processing SDK	Provides a software-accelerated runtime environment for running deep neural networks
Qualcomm Lite Runtime Reference	Provides the TensorFlow framework for on-device AI inferencing

For a comprehensive package of software, tools, and documentation, go to Qualcomm Linux.

8 Development regulatory notice

This development device contains RF/digital hardware and software intended for engineering development, engineering evaluation, or demonstration purposes only and is intended for use in a controlled environment. This device is not being placed on the market, leased or sold for use in a residential environment or for use by the general public as an end user device.

This development device is not intended to meet the requirements of a commercially available consumer device including those requirements specified in the European Union directives applicable for Radio devices being placed on the market, FCC equipment authorization rules or other regulations pertaining to consumer devices being placed on the market for use by the general public.

The device should be installed and operated with a minimum distance of 8 inches (20 cm) between radiating antennas and bystanders. Additional RF exposure information can be found on the FCC website at http://www.fcc.gov/encyclopedia/radio-frequency-safety.

FCC NOTICE: This kit is designed to allow:

(1) Product developers to evaluate electronic components, circuitry, or software associated with the kit to determine whether to incorporate such items in a finished product and

(2) Software developers to write software applications for use with the end product. This kit is not a finished product and when assembled may not be resold or otherwise marketed unless all required FCC equipment authorizations are first obtained. Operation is subject to the condition that this product not cause harmful interference to licensed radio stations and that this product accept harmful interference. Unless the assembled kit is designed to operate under Part 15, Part 18, or Part 95 of the U.S. Federal Communications Commission (FCC) rules in Title 47 of the Code of Federal Regulations (CFR), the operator of the kit must operate under the authority of an FCC license holder or must secure an experimental authorization under Part 5 of the FCC's rules.

If the kit is modified through the addition of external RF components such as high gain antennas then the owner must ensure operation with existing rules.

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.