8-07-2022

VVDN QCS 610 EVK User Guide

Revision History:

DATE	REVISION NO.	DESCRIPTION	BY
29-10-2021	**A0-01**	**Initial release**	**VVDN**
08-11-2021	**A0-02**	**Added Mini PCIe & Expansion HDR details**	**VVDN**
16-03-2022	**A0-03**	**Added performance validation of interfaces**	**VVDN**
23-03-2022	**A0-04**	**Internal review comments updated**	**VVDN**
28-06-2022	**A0-05**	**Camera section added**	**VVDN**
8-07-2022	**A0-06**	**Internal review comments updated**	**VVDN**

Table of Contents

List of Figures

1. Introduction

This document contains information about the powering-up steps required for the VVDN QCS610 development board.

2. QCS610 + Carrier Board

A 12V DC adaptor is required to power the QCS610 development board (minimum 1A current is required)

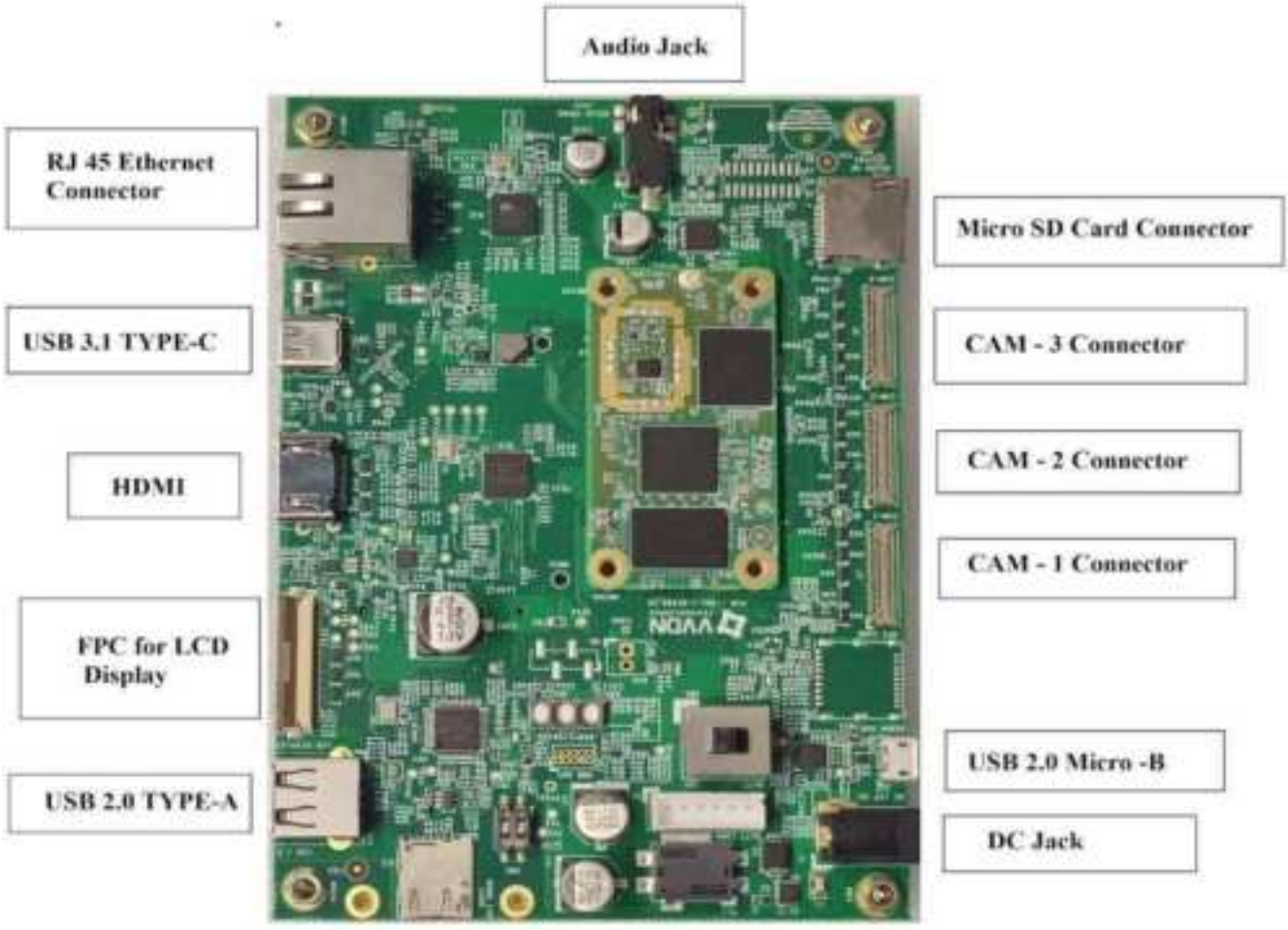


Figure 1: QCS610 SOM + Carrier Board

2.1 Power Source Selection

Below figures shows the switch position options for DC Power and Battery Power selection.

Figure 2: Position of Switch

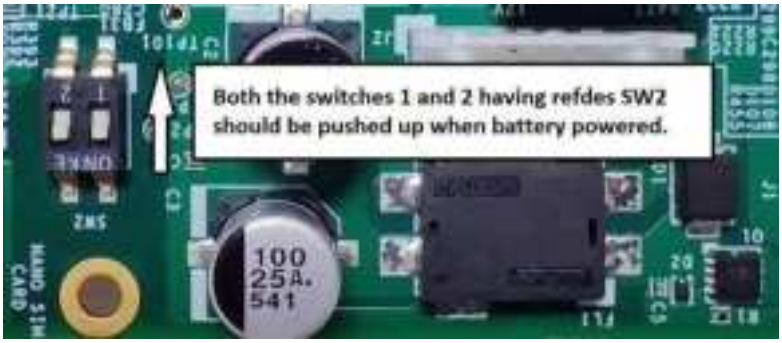


Figure 3: Positioning of Switch (SW2)

3. Debug Console (Micro B)

Connect USB Micro -B cable to the J13 port of the board and PC to get the UART console.

Figure 4: Micro-B (J13)

4. USB 3.1 Type -C

Connect Type -C cable to the USB 3.1 Type C port of the board and PC to get adb console.

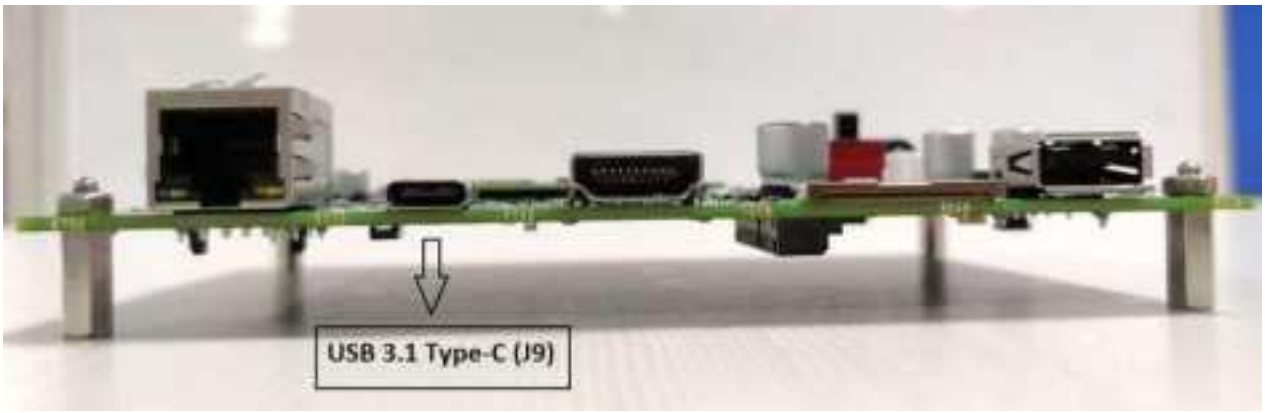


Figure 5: USB 3.1 Type-C (J9)

5. UART Console setup

UART Console can be obtained either using minicom or picocom.

Using minicom-

- Run **"minicom -s"** in the PC and make the minicom settings from **"Serial port setup"** option. Refer the image attached below:

Figure 6: Console settings in minicom

- After making the proper settings run the command ***"minicom -D /dev/tty<device_node>"*** in the laptop to get the board console.

For example : minicom -D /dev/ttyUSB0

Check the print by running the command ***"dmesg | grep tty | tail -1"*** as shown below

Using picocom-

- Use the command ***"picocom -b 115200 /dev/tty<device_node>"*** in the laptop to get the board console.

For example : picocom -b 115200 /dev/ttyUSB0.

Figure 7: Console settings in picocom

6. Powering up the system

Below are the steps needed for setting up the system

1. USB 3.1 type-C cable, connect the board and PC for ADB shell

2. Micro-B cable, connect the board and PC for Debug Console

3. 12V DC adapter to the DC jack for Power

4. Once the Red LED (D67) is ON ,Press the Phone ON button (SW4) (QCS610 has multiple power On events like USB In , CBL_PWR# etc… depends on that the board may auto power On)

Figure 8: Connections

- Press the phone ON switch shown below to power on QCS610 SOM board or by simply connecting the type-C cable itself the SOM board will power ON.

Figure 9: Phone ON Switch

- After powering the SOM board , booting starts and the booting logs can be seen in picocom/minicom terminal.

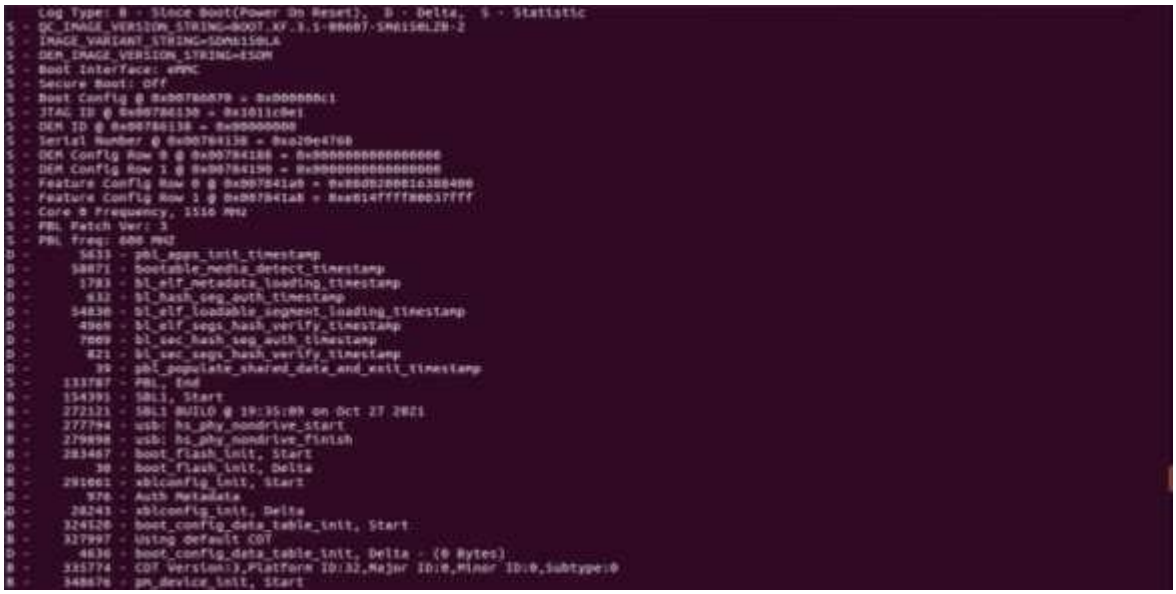


Figure 10: Logs - Booting

After booting at the login screen, enter "**root**" as the username and "**oelinux123**" as password .The image below shows the obtained console and login screen.

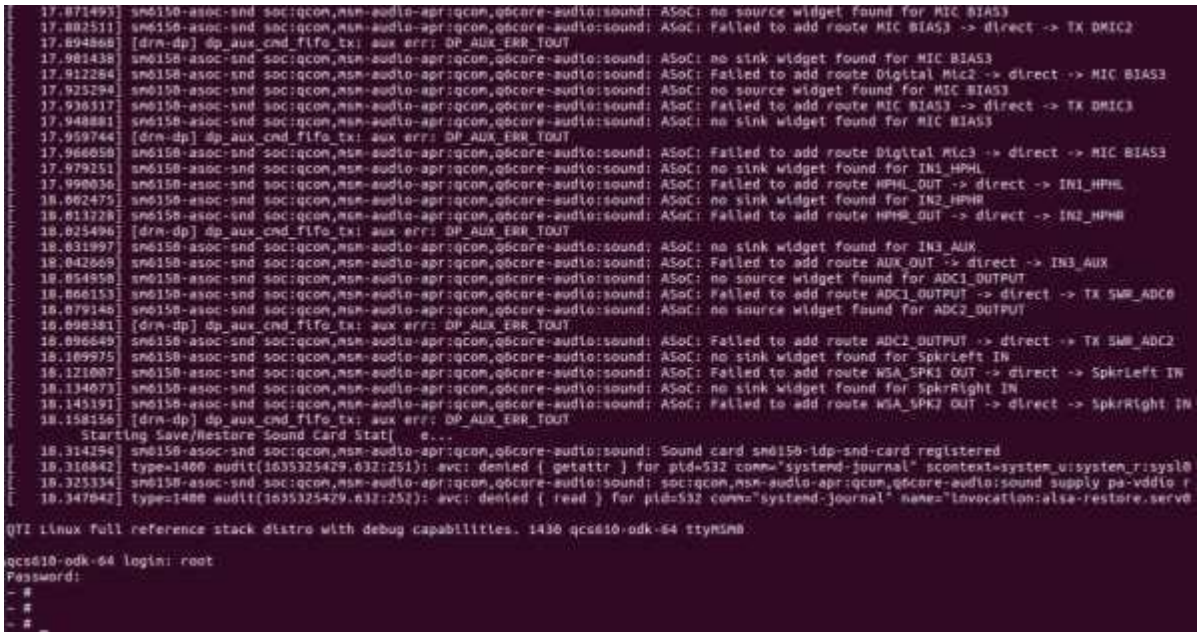


Figure 11: Console and Login Screen

7. ADB setup:-

Connect the USB 3.1 Type C cable, **wait for few seconds after the boot up** and then open the terminal and enter the below commands for getting into adb shell.

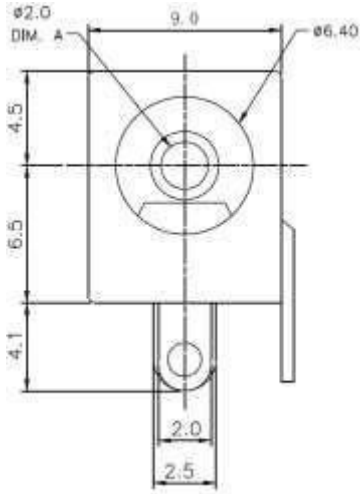
1. adb root
2. adb shell

The figure below show the obtained terminal and login

Figure 12: Terminal showing ADB

8. DC Power Jack Details

Manufacturer: - CHUFON TECHNOLOGY CO., LTD (C-TEK)
Part Number: - C020083-02



Dimensions are in mm

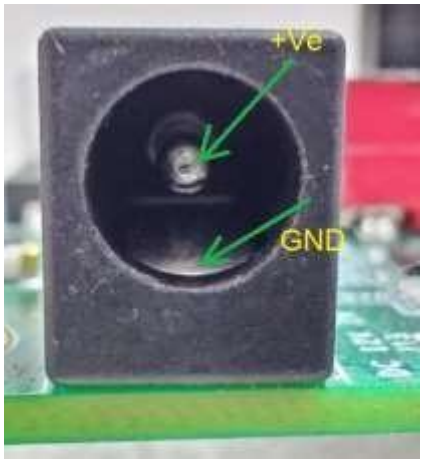


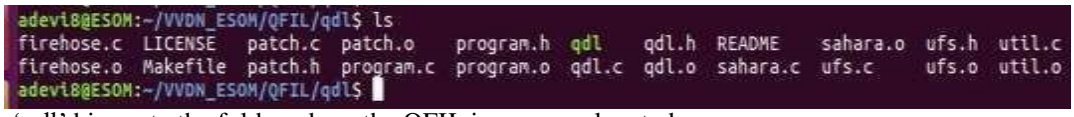
Figure 13: DC Power Jack Details

9. QDL & FASTBOOT Flashing Guide (Linux)

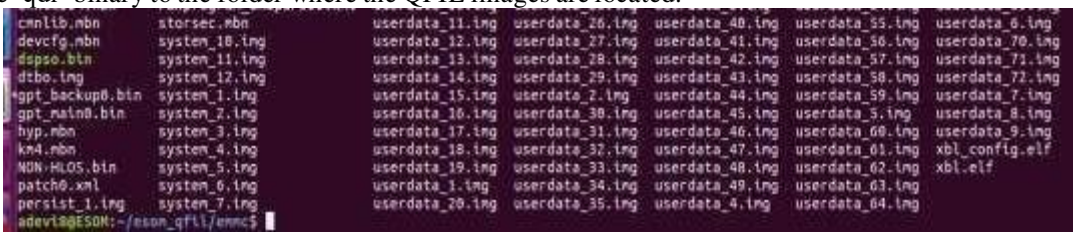
The QCS610 Device can be flashed using QFIL (initial full flashing) or using Fastboot.

9.1 QDL Flashing Steps (Linux):-

1. Install below packages
   - **sudo apt-get install libxml2-dev libudev-dev**
2. Clone qdl source code and compile
   - **git clone** https://git.linaro.org/landing-teams/working/qualcomm/qdl.git
   - **cd qdl**
   - **make** (A binary named '**qdl**' will be compiled in the cloned folder)

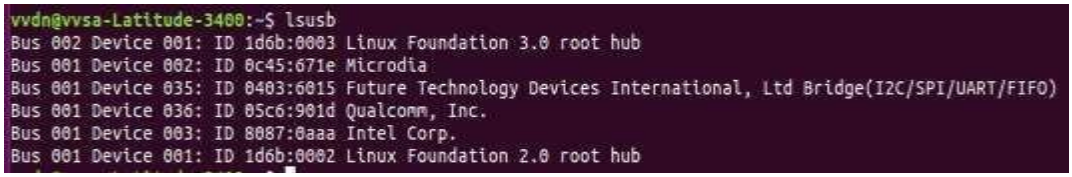


3. Copy the 'qdl' binary to the folder where the QFIL images are located.



4. Switch on the device and connect the C type USB cable to it.
5. Check the device availability in the terminal using the command:
   - **adb devices**

   (or)
   - **lsusb**



6. Now open terminal and type the following commands:
   - **adb root**
   - **adb reboot edl**

   (Now the device will be in EDL mode (9008) and the flashing process will begin)

```
vvdn@vvsa-Latitude-3400:~$ lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 002: ID 0c45:671e Microdia
Bus 001 Device 035: ID 0403:6015 Future Technology Devices International, Ltd Bridge(I2C/SPI/UART/FIFO)
Bus 001 Device 036: ID 05c6:901d Qualcomm, Inc.
Bus 001 Device 003: ID 8087:0aaa Intel Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
vvdn@vvsa-Latitude-3400:~$ adb root
restarting adbd as root
vvdn@vvsa-Latitude-3400:~$ adb reboot edl
vvdn@vvsa-Latitude-3400:~$ lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 002: ID 0c45:671e Microdia
Bus 001 Device 035: ID 0403:6015 Future Technology Devices International, Ltd Bridge(I2C/SPI/UART/FIFO)
Bus 001 Device 038: ID 05c6:9008 Qualcomm, Inc. Gobi Wireless Modem (QDL mode)
Bus 001 Device 003: ID 8087:0aaa Intel Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

7. Run the below command from the flat build folder (the folder in which the qdl and qfil binaries are located):
   - For QCS610:

     sudo ./qdl --debug --storage emmc prog_firehose_ddr.elf rawprogram_unsparse0.xml patch0.xml

   (Now the linux system will be in waiting mode for the EDL device)

```
root@vvsa-Latitude-3400:/home/vvdn/Documents/VVDN_ESOM/Images/emmc# sudo ./qdl --debug --storage emmc prog_firehose_ddr.elf rawprogram_unsparse
0.xml patch0.xml
Waiting for EDL device
```

   Now, the qdl waiting in previous terminal will continue flashing the device

```
root@vvsa-Latitude-3400:/home/vvdn/Documents/VVDN_ESOM/Images/emmc# sudo ./qdl --debug --storage emmc prog_firehose_ddr.elf rawprogram_unsparse
0.xml patch0.xml
Waiting for EDL device
HELLO version: 0x2 compatible: 0x1 max_len: 1024 mode: 0
READ64 image: 13 offset: 0x0 length: 0x40
READ64 image: 13 offset: 0x40 length: 0x3b0
READ64 image: 13 offset: 0x1000 length: 0x1000
READ64 image: 13 offset: 0x2000 length: 0xcd0
READ64 image: 13 offset: 0x57ce0 length: 0x1000
READ64 image: 13 offset: 0x58ce0 length: 0x1000
READ64 image: 13 offset: 0x59ce0 length: 0xc
```

8. Now the binaries are successfully flashed and we can reboot the device.

```
LOG: INFO: Using scheme of value= 0
FIREHOSE READ: <?xml version="1.0" encoding="UTF-8" ?>
<data>
<response value="ACK" rawmode="false" /></data>
partition 0 is now bootable
FIREHOSE WRITE: <?xml version="1.0"?>
<data><power value="reset"/></data>

FIREHOSE READ: <?xml version="1.0" encoding="UTF-8" ?>
<data>
<log value="INFO: Calling handler for power" /></data>
LOG: INFO: Calling handler for power
FIREHOSE READ: <?xml version="1.0" encoding="UTF-8" ?>
<data>
<response value="ACK" rawmode="false" /></data>
```

9.2 FASTBOOT STEPS :-

Power on the device and after the device booting is completed enter the below commands from terminal.

1. extract the tar file

$ tar -xf ESOM_fastboot_img.tar.gz

$ cd Esom_fastboot

```
vkchlt0459@7870--VISN--KOCHI1:~$ cd Esom_fastboot/
vkchlt0459@7870--VISN--KOCHI1:~/Esom_fastboot$
```

$ adb root

```
vkchlt0459@7870--VISN--KOCHI1:~/Esom_fastboot$ adb root
restarting adbd as root
vkchlt0459@7870--VISN--KOCHI1:~/Esom_fastboot$ adb devices
List of devices attached
5a8bd67d        device
```

$ adb reboot bootloader

Check device is coming to fastboot mode or not using :

$ fastboot devices

```
vkchlt0459@7870--VISN--KOCHI1:~/Esom_fastboot$ adb reboot bootloader
vkchlt0459@7870--VISN--KOCHI1:~/Esom_fastboot$ fastboot devices
5a8bd67d        fastboot
vkchlt0459@7870--VISN--KOCHI1:~/Esom_fastboot$
```

$ fastboot flash abl ./abl.elf

$ fastboot flash boot ./boot.img

$ fastboot flash dtbo ./dtbo.img

$ fastboot flash system ./system.img

$ fastboot flash persist ./persist.img

$ fastboot flash userdata ./userdata.img

$ fastboot reboot

```
adb: unable to connect for root: no devices/emulators found
error: no devices/emulators found
error: no devices/emulators found
target reported max download size of 805306368 bytes
sending 'abl' (148 KB)...
OKAY [  0.009s]
writing 'abl'...
OKAY [  0.007s]
finished. total time: 0.016s
target reported max download size of 805306368 bytes
sending 'boot_a' (12656 KB)...
OKAY [  0.326s]
writing 'boot_a'...
OKAY [  0.109s]
finished. total time: 0.435s
target reported max download size of 805306368 bytes
sending 'dtbo' (2969 KB)...
OKAY [  0.086s]
writing 'dtbo'...
OKAY [  0.037s]
finished. total time: 0.123s
target reported max download size of 805306368 bytes
erasing 'system_a'...
OKAY [  0.088s]
sending 'system_a' (616638 KB)...
OKAY [ 15.881s]
writing 'system_a'...
OKAY [  0.000s]
finished. total time: 15.969s
target reported max download size of 805306368 bytes
```

After flashing is successfully completed you will get this log as finished.

```
target reported max download size of 805306368 bytes
sending 'persist' (4252 KB)...
OKAY [  0.118s]
writing 'persist'...
OKAY [  0.038s]
finished. total time: 0.156s
target reported max download size of 805306368 bytes
erasing 'userdata'...
OKAY [  0.070s]
sending 'userdata' (135785 KB)...
OKAY [  3.168s]
writing 'userdata'...
OKAY [  0.000s]
finished. total time: 3.238s
rebooting...

finished. total time: 0.050s
vkchlt0455@7870--VISN--KOCHI1:~/Esom_fastboot$
```

After flashing, restart the device.

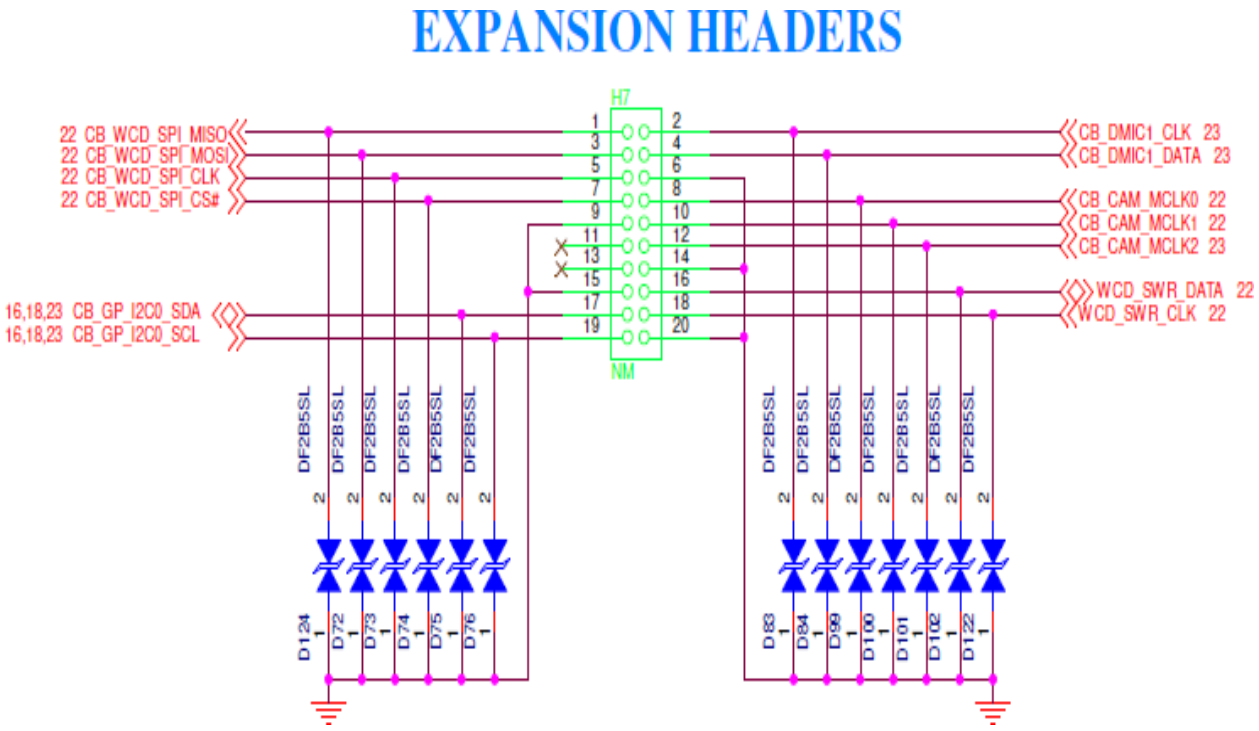
9.3 QFIL Flashing (using Windows PC)

QFIL is a Windows-based image flashing tool and must be used when the developer uses Windows for flashing the device.

For QFIL please refer to the Document "QCS610/QCS410 Linux Platform Development Kit Software guide."

Ref ID: **80-PL631-200 (section 2.4.2.2)**

10. Expansion Header details (H7)



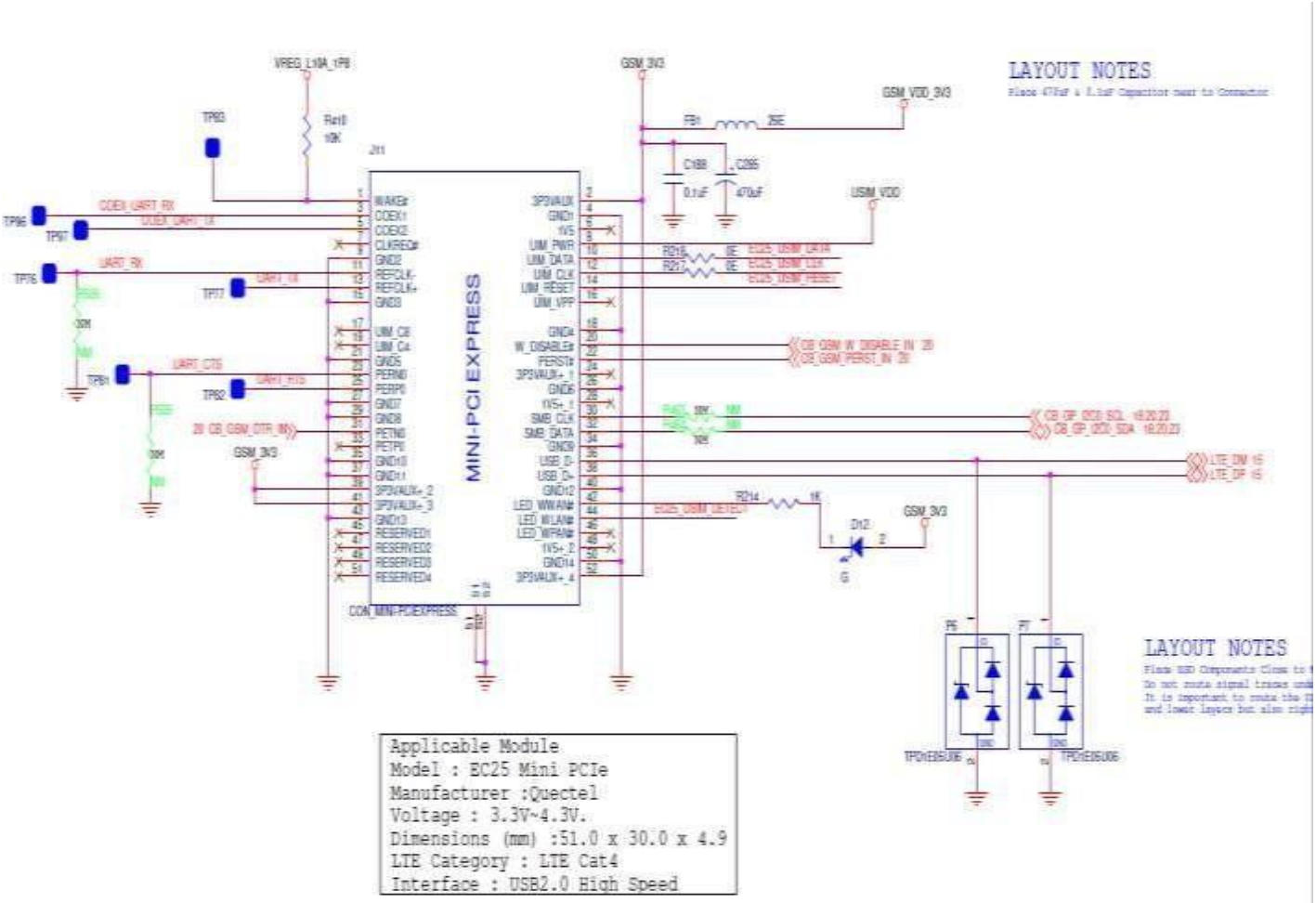
All these signals are coming under 1.8V IO level.

Pull ups are present in Carrier board (1.8V) for (CB_GP_I2C0_SCL, CB_GP_I2C0_SDA) and this I2C is already connected to Audio Codec in carrier board.

11. Mini PCIe Connector Details (J11)

Manufacturer = TE Connectivity

P/N = 1775862-2



Mini PCIe connector is an expansion option to add LTE feature to the carrier board. This can directly support EC25 LTE module from Quectel.

The Power “GSM_3V3” can source up to 3A current.

QCS610 can communicate to the Mini PCIe module through the USB2.0 interface.

The Signals EC25_USIM_DATA , EC25_USIM_CLK & EC25_USIM_RESET signals are for SIM card interface (J12).

12. Performance validation of interfaces

The section covers the performance validation procedure and test results of major interfaces

12.1 Ethernet

12.1.1 IPERF Test

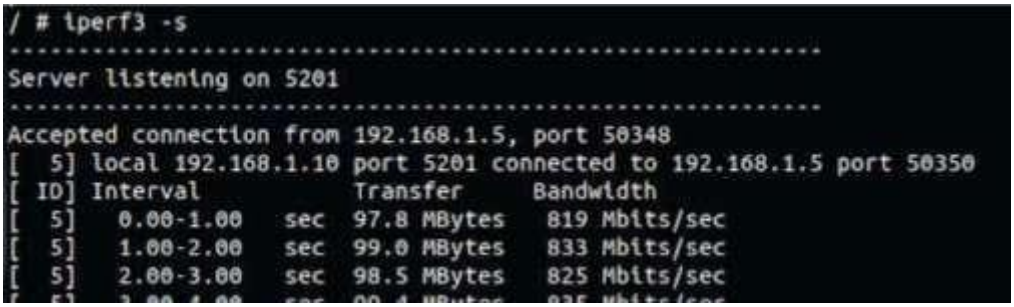
The LAN data rates are performed using the IPERF tool.

Iperf is a tool for network performance measurement. The device and PC can be used as either client or server and perform thetest.

The test procedures are as follows,

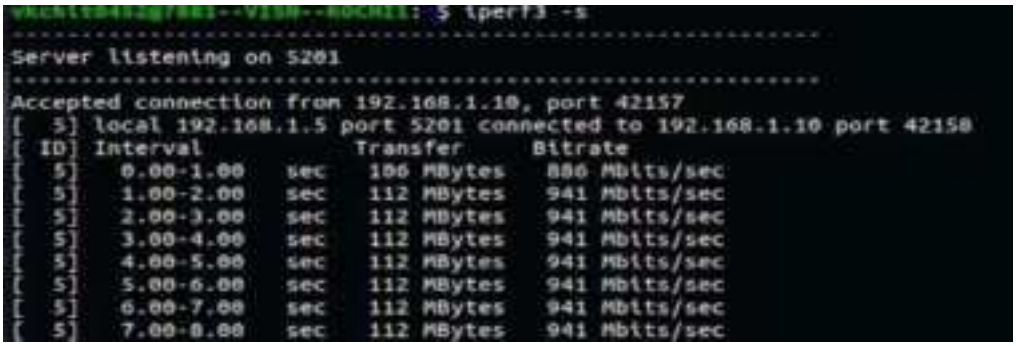
1 2.1.1.1 Device as Server

- On Device enter the command **iperf3 -s**
- On PC enter the command **iperf3 -c <board ip> -t 60**
- Then the data rate as shown below



12.1.1.2 Device as Client

- On PC enter the command **iperf3 -s**
- On Device enter the command**iperf3 -c <laptop ip> -t 60**
- Then the data rate as shown below



12.1.2 Streaming over Ethernet

- Run the following command in the device debug console or adb shell

```
gst-launch-1.0 -e qtiqmmfsrc name=qmmf ! video/x-raw\(memory:GBM\), width=1920, height=1080, framerate=30/1,format=NV12 ! omxh264enc target-bitrate=600000 control-rate=3 min-quant-i-frames=25 min-quant-p-frames=25 max-quant-i-frames=36 max-quant-p-frames=36 interval-intraframes=149 num-ltr-frames=1 ! video/x-h264, profile=high ! h264parse config-interval=1 ! mpegtsmux name=muxer ! queue ! tcpserversink port=8900 host=x.x.x.x
```

"host=x.x.x.x", the IP address must be the same as configured for the device.

- Start VLC-Media-Player on PC

Open VLC media play and please turn off the Windows Firewall!
Go to "Media" —> "Open Network Stream".
Enter "tcp://127.0.0.1:8900" for network URL

Note: Make sure the device(board) and host(PC) are in the same network

12.2 Wi-Fi

12.2.1 Wi-Fi Client mode

To verify the connectivity while device is in station mode, Execute the following commands.

- Scan Nearby Acess Point

```
# iw wlan0 scan
```

- Connect with AP and get IP address by using **below command**
  Write SSID and password in /etc/misc/wifi/wpa_supplicant.conf file.

```
# killall wpa_supplicant #
killall dhcpcd
# wpa_supplicant -Dnl80211 -iwlan0 -c /etc/misc/wifi/wpa_supplicant.conf -dddd #
dhcpcd -i wlan0
```

- Check the IP address

```
# ifconfig wlan0
```

- Check the Internet connection by pinging to the google

```
# ping 8.8.8.8
```

12.2.2 Wi-Fi AP Mode

- Execute the following commands to check hostapd:

```
# ps | grep hostapd
   2345          /etc/misc/wifi/hostapd . conf
```

- Execute the following commands in one Terminal

```
# interface=softap0

# pkill -9 wpa_supplicant

# hostapd /etc/misc/wifi/hostapd-open.conf -dddd &
```

- Execute the following commands in another terminal

```
# ifconfig wlan0 192.168.43.1 netmask 255.255.255.0 up # pkill
-9 dnsmasq
 # dnsmasq -i wlan0 --dhcp-range=192.168.43.10,192.168.43.100,12h --dhcp-
leasefile=/data/dnsmasq_d.lease
```

- Use phone to connect this Wi-Fi named: AndroidAP-sdmsteppe
- Excecute the following command to check the status of hostapd

```
#hostapd_cli -i wlan0
```

12.2.3 Streaming over Wi-Fi

- Run the following command in the device debug console or adb shell

```
gst-launch-1.0 -e qtiqmmfsrc name=qmmf ! video/x-raw\(memory:GBM\), width=1920, height=1080,
framerate=30/1,format=NV12 ! omxh264enc target-bitrate=600000 control-rate=3 min-quant-i-
frames=25 min-quant-p-frames=25 max-quant-i-frames=36 max-quant-p-frames=36 interval-
intraframes=149 num-ltr-frames=1 ! video/x-h264, profile=high ! h264parse config-interval=1 !
mpegtsmux name=muxer ! queue ! tcpserversink port=8900 host=x.x.x.x
```

"host=x.x.x.x", the IP address must be the same as configured for the device.

- Start VLC-Media-Player on PC

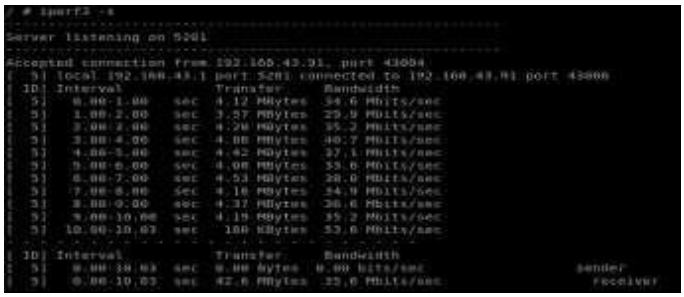
  Open VLC media play and please turn off the Windows Firewall!
  Go to "Media" —> "Open Network Stream".
  Enter "tcp:/ 127.0.0.1:8900" for network URL

Note: Make sure the device(board) and host(PC) are in the same network

12.2.4 IPERF Test

12.2.4.1 Device as Server

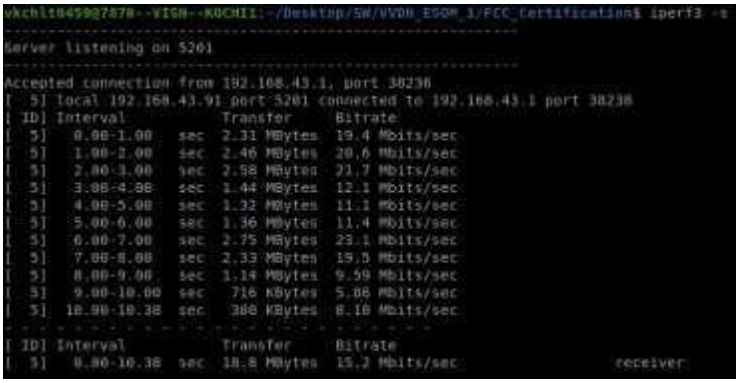
- On Device enter the command **iperf3 -s**
- On PC enter the command **iperf3 -c <board ip> -t 60**
  Then the data rate as shown below



While connecting the antenna

12.2.4.2 Device as Client

- On PC enter the command **iperf3 -s**
- On Device enter the command**iperf3 -c <laptop ip> -t 60**
- Then the data rate as shown below



While connecting the antenna

12.3 Bluetooth

12.3.1 Bluetooth Generic Access Profile (GAP)

- Before running btapp, run the btproperty in the background (run only once at the beginning

```
btproperty &
 btapp
```

- After running btapp, type gap_menu and press Enter.

gap_menu

```
**********Menu**********
 enable
 disable
 inquiry
 cancel_inquiry
 pair<space><bt_address><space><transport>  eg. pair 00:11:22:33:44:55
0(auto)/1(BREDR)/2(BLE)
 unpair<space><bt_address>  eg. unpair 00:11:22:33:44:55
 inquiry_list
 bonded_list
 get_state
 get_bt_name
 get_bt_address
 set_bt_name<space><bt name>  eg. set_bt_name MDM_Fluoride
 set_scan_mode<space><scan mode value (range 0-2)>  eg. set_scan_mode 0 --0-
BT_SCAN_MODE_NONE,1- BT_SCAN_MODE_CONNECTABLE, 2-
BT_SCAN_MODE_CONNECTABLE_DISCOVERABLE
set_afh<space><AFH_Host_Channel_Classification>  eg.  set_afh  1122334455667789f0
send_hci_cmd<space><hci_cmd>  eg.  send_hci_cmd  01,04,05,33,8b,9e,0a,00  -  For  Inquiry
main_menu
```

12.3.2 Enable Bluetooth

After running btapp, input “enable” and press “Enter”enable

```
enable
killall: wcnssfilter: no process killed
killall: btsnoop: no process killed killall:
qcbtdaemon: no process killed
/bin/sh: qcbtdaemon: not found
/data/vendor/ssrdump is created
diag: Unable to get remote processor info. Continuing with just the local processor
 Load Audio HAL started
 BT State is ON
```

12.3.3 Start inquiry

After running enable, type inquiry and press Enter to start inquiry.

```
inquiry
        Inquiry Started
        Device Found details:
        Found device Addr: 50:29:f5:ee:7d:ca
        Found device Name: Realme 2 Pro Device
        Type is: 1
        Device Found details:
        Found device Addr: 50:2f:9b:ce:91:3f
        Found device Name: 7881--VISN--KOCHI1
        Device Type is: 1
        Inquiry Stopped automatically
```

To cancel inquiry, issue the following command while the inquiry in progress: cancel_inquiry

12.3.4 Check the inquiry list

After running inquiry, type inquiry_list and press Enter to check the list.
inquiry_list

```
        ****************************** Inquiry List ********************************
        1       Realme 2 Pro                    50:29:f5:ee:7d:ca
        2       7881--VISN--                    50:2f:9b:ce:91:3f
        ****************************** End of List *********************************
```

12.3.5 Pair outgoing SSP

- Use the following command to pair outgoing SSP

  ```
  pair<bd_address>
  ```

- To accept or reject the outgoing pairing for the following example (pair e4:ba:d9:10:00:c9), type Yes or No and press Enter.

  ```
  pair 50:29:f5:ee:7d:ca 0
   Auto select in the stack
   ACL state:0 change with reason 00 for device: 50:29:f5:ee:7d:ca

  ************************************************
   BT pairing request::Device Realme 2 Pro::Pairing Code:: 716151
  ************************************************
   ** Please enter yes / no ** yes

  ************************************************
   Pairing state for Realme 2 Pro is BONDED
  ```

To unpair the device, issue the following command- unpair<space><bt_address>

```
unpair 50:29:f5:ee:7d:ca
```

12.3.6 Check the bonded list

- After running btapp, type bonded_list and press Enter to check the bonded list:

```
bonded_list
***************************** Bonded Device List *****************************
Realme 2 Pro                        50:29:f5:ee:7d:ca
***************************** End of List ************************************
```

- Disconnect bonded, type disable and press Enter disable:

```
disable
 current State = 1, new state = 0 killall:
qcbtdaemon: no process killed killall:
wcnssfilter: no process killed
 BT State is OFF
```

- To exit from btapp, navigate to the main menu and enter the following command:

```
exit
BT is Already OFF, Just exiting APP
Killed
# |
```

12.4 IMX462 Camera Board Connection to QCS610 EVK

Please refer to “VVDN_IMX462_CAMERA BOARD_DATASHEET_A1” for Camera connection details.

Below are the steps for validating the Camera Capturing and Playback on the device. All the commands shown below need to be executed on the device debug console/ADB.

12.4.1. Prerequisites

- **Disable rootfs write-protect**

```
adb root
adb remount
adb disable-verity
adb reboot
adb wait-for-device
adb root
adb shell mount -o remount,rw,exec /
```

- Software Installed - VLC (on PC).

- **Pre Configuration for camera working properly**

```
adb shell mkdir -p /etc/camera
adb shell touch /etc/camera/camxoverridesettings.txt
adb shell "echo IFEDualClockThreshold=600000000 >> /etc/camera/camxoverridesettings.txt"
```

- Run adb reboot to reboot the device.

- **Set up the GStreamer environment**: (requires on every session)

```
adb shell
source /etc/gstreamer1.0/set_gst_env.sh
```

12.4.2. Steps for validating camera sensor - Using Recording pipeline

Single H264/H265 stream encoding with MP4 file dump

- Run below GStreamer pipeline for 1080p video recording on device.

$ gst-launch-1.0 -e qtiqmmfsrc name=camsrc ! video/x-raw\(memory:GBM\), format= ?gst-launch-1.0 -e qtiqmmfsrc name=camsrc camera=1 ! video/x-raw\(memory:GBM\), format= NV12,width=3840,height=2160,framerate=30/1 ! queue ! omxh264enc ! queue ! h264parse ! mp4mux ! queue ! filesink location="/data/mux_cam0.mp4"

12.4.3. Steps for validating camera sensor - Using streaming pipeline

12.4.3.1. Streaming over USB

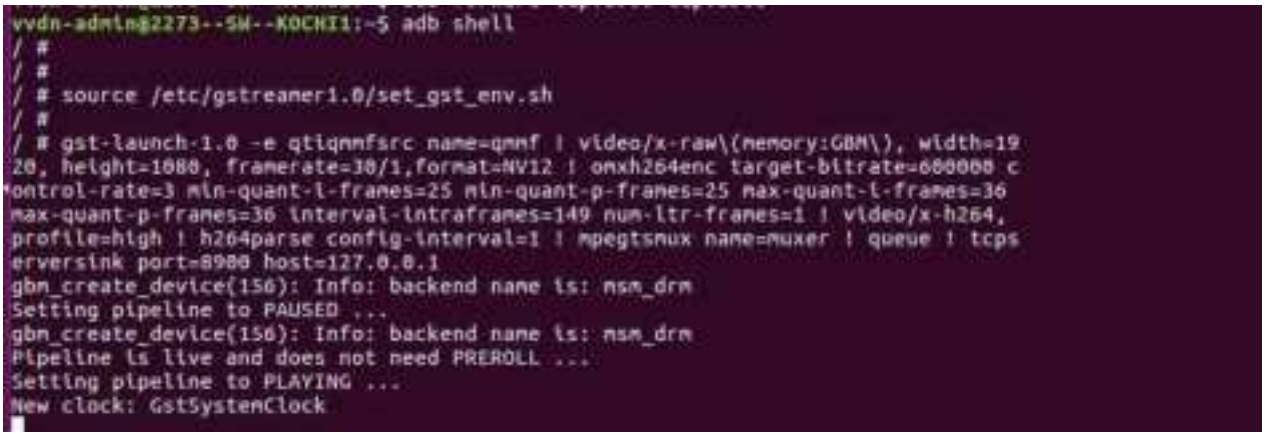
- Run below tcp port forwarding command in the PC terminal where the device is connected over USB.

adb forward tcp:8900 tcp:8900

- Run the following command in the device debug console or adb shell

$ gst-launch-1.0 -e qtiqmmfsrc name=qmmf ! video/x-raw\(memory:GBM\), width=1920, height=1080, framerate=30/1,format=NV12 ! omxh264enc target-bitrate=600000 control-rate=3 min-quant-i-frames=25 min-quant-p-frames=25 max-quant-i-frames=36 max-quant-p-frames=36 interval-intraframes=149 num-ltr-frames=1 ! video/x-h264, profile=high ! h264parse config-interval=1 ! mpegtsmux name=muxer ! queue ! tcpserversink port=8900 host=127.0.0.1

- For taking live stream in PC, install video player app and take network stream from below link tcp://127.0.0.1:8900



Single Camera streaming over USB

Dual camera streaming over USB

Dual n/w stream test:

Device:

- source /etc/gstreamer1.0/set_gst_env.sh

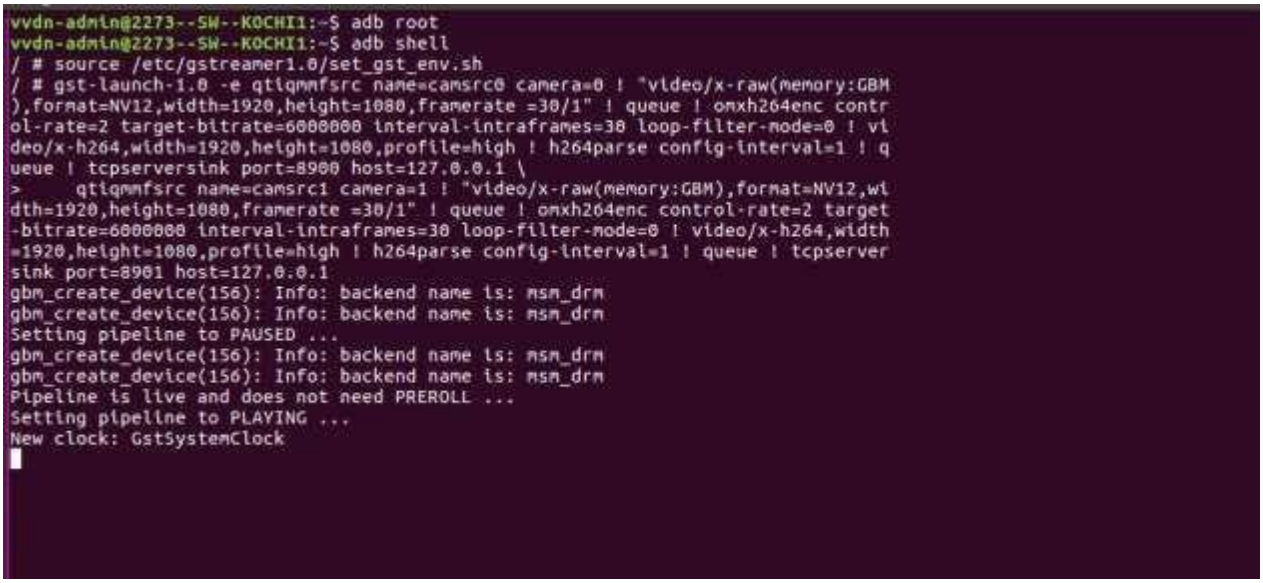
- gst-launch-1.0 -e qtiqmmfsrc name=camsrc0 camera=0 ! "video/x-raw(memory:GBM),format=NV12,width=1920,height=1080,framerate =30/1" ! queue ! omxh264enc control-rate=2 target-bitrate=6000000 interval-intraframes=30 loop-filter-mode=0 ! video/x-h264,width=1920,height=1080,profile=high ! h264parse config-interval=1 ! queue ! tcpserversink port=8900 host=127.0.0.1 \ qtiqmmfsrc name=camsrc1 camera=1 ! "video/x-raw(memory:GBM),format=NV12,width=1920,height=1080,framerate =30/1" ! queue ! omxh264enc control-rate=2 target-bitrate=6000000 interval-intraframes=30 loop-filter-mode=0 ! video/x-h264,width=1920,height=1080,profile=high ! h264parse config-interval=1 ! queue ! tcpserversink port=8901 host=127.0.0.1

PC :

- adb forward tcp:8900 tcp:8900
- adb forward tcp:8901 tcp:8901

PC:

- ffplay tcp://127.0.0.1:8900
- ffplay tcp://127.0.0.1:8901



Dual Camera streaming over USB

12.4.3.2. Streaming over Network

- Run the following command in the device debug console or adb shell

gst-launch-1.0 -e qtiqmmfsrc name=qmmf ! video/x-raw\(memory:GBM\), width=1920, height=1080,

framerate=30/1,format=NV12 ! omxh264enc target-bitrate=600000 control-rate=3 min-quant-i-frames=25 min-quant-p-frames=25 max-quant-i-frames=36 max-quant-p-frames=36 interval-intraframes=149 num-ltr-frames=1 ! video/x-h264, profile=high ! h264parse config-interval=1 ! mpegtsmux name=muxer ! queue ! tcpserversink port=8900 host=x.x.x.x

"host=x.x.x.x",the IP address must be the same as configured for the device.

- Start VLC-Media-Player on PC

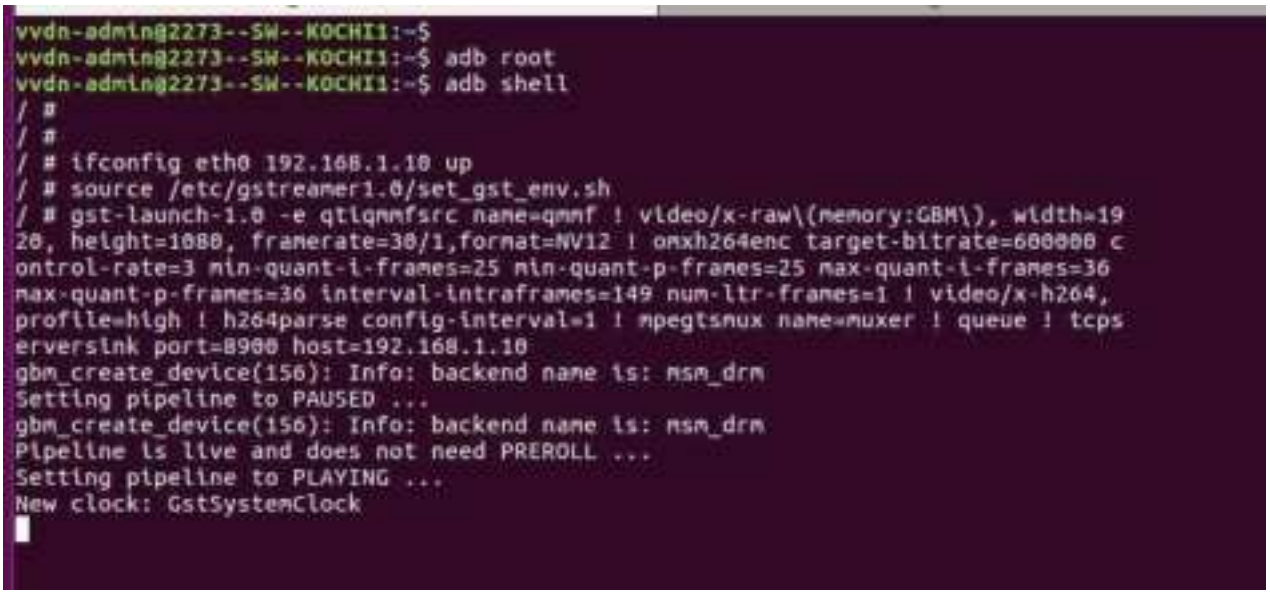
Windows :

Open VLC media play and please turn off the Windows Firewall! Go to "Media" → "Open Network Stream". Enter "tcp://127.0.0.1:8900" for network URL.

Ubuntu :

vlc -vvv tcp://x.x.x.x:8900

Note: Make sure the device(board) and host(PC) are in the same network segment!



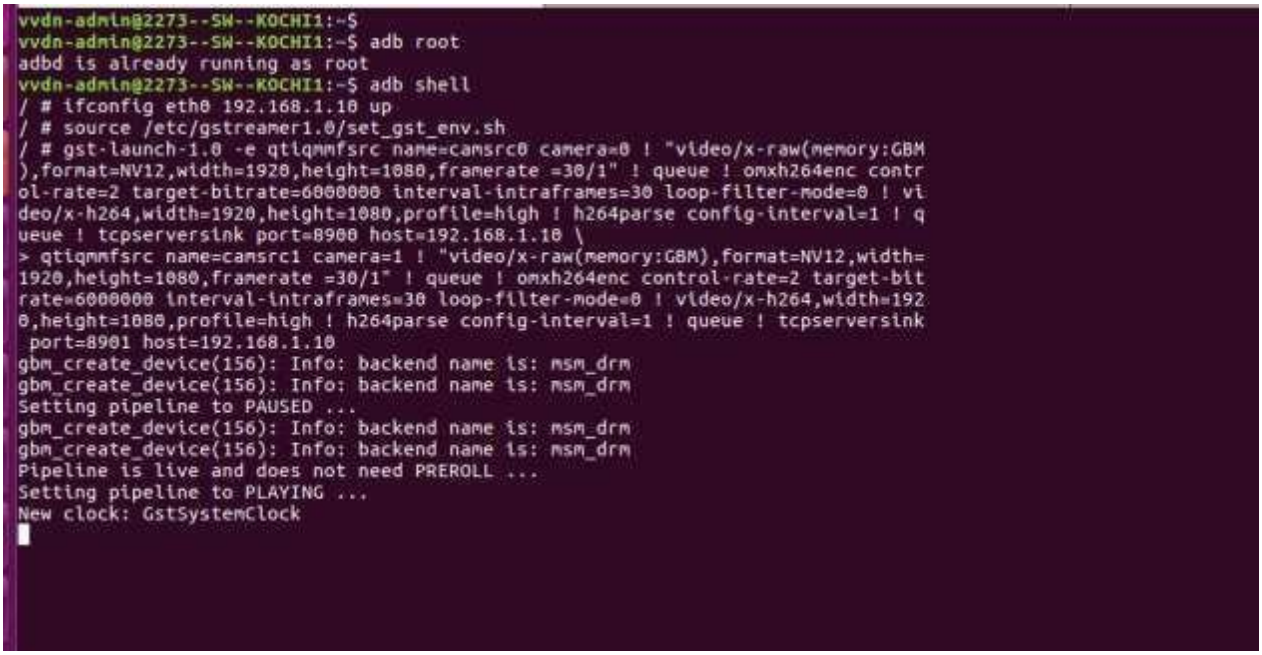
Single Camera streaming over Network

Dual Camera Streaming through WiFi or Ethernet

- source /etc/gstreamer1.0/set_gst_env.sh

- gst-launch-1.0 -e qtiqmmfsrc name=camsrc0 camera=0 ! "video/x-raw(memory:GBM),format=NV12,width=1920,height=1080,framerate =30/1" ! queue ! omxh264enc control-rate=2 target-bitrate=6000000 interval-intraframes=30 loop-filter-mode=0 ! video/x-

h264,width=1920,height=1080,profile=high ! h264parse config-interval=1 ! queue ! tcpserversink port=8900 host=<*HOST_IP*> \ qtiqmmfsrc name=camsrc1 camera=1 ! "video/x-raw(memory:GBM),format=NV12,width=1920,height=1080,framerate =30/1" ! queue ! omxh264enc control-rate=2 target-bitrate=6000000 interval-intraframes=30 loop-filter-mode=0 ! video/x-h264,width=1920,height=1080,profile=high ! h264parse config-interval=1 ! queue ! tcpserversink port=8901 host=<*HOST_IP*>



Dual Camera streaming over Network

12.4.3.3. Streaming over HDMI Display

Start the weston

- Run the following command to check if Weston is running

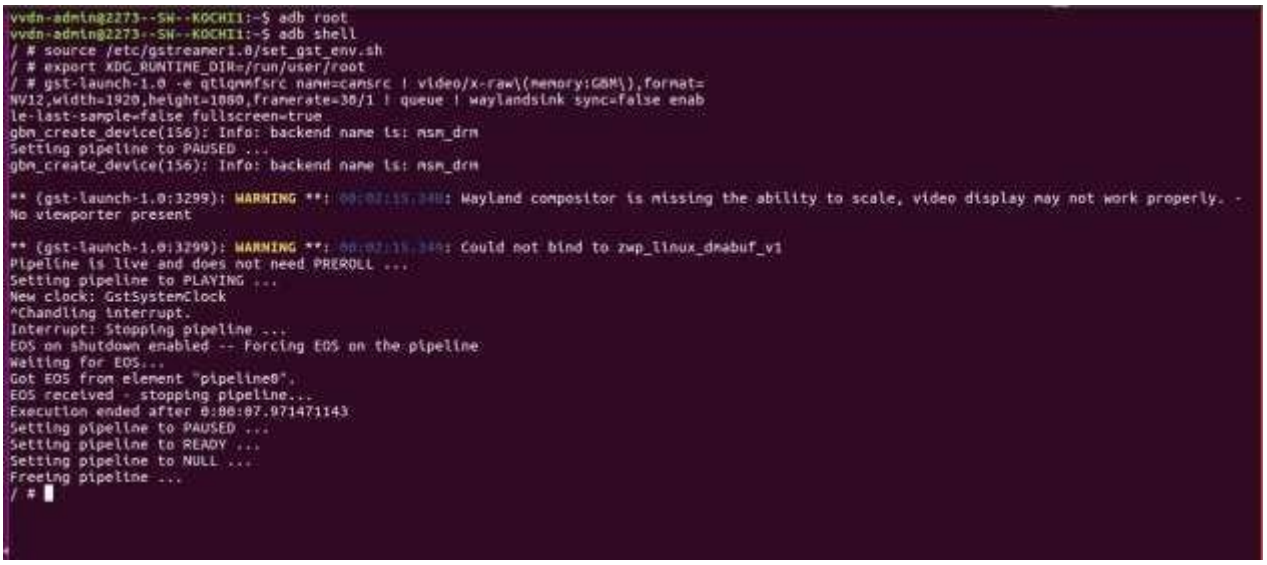
/# ps | grep weston

- Ensure that the external display monitor (1080p) is connected via HDMI.
- Set up the environment variable requires on every session

export XDG_RUNTIME_DIR=/run/user/root

Run the following command in the device debug console or adb shell

$ gst-launch-1.0 -e qtiqmmfsrc name=camsrc camera=0 ! video/x-raw\(memory:GBM\),format=NV12,width=1920,height=1080,framerate=30/1 ! queue ! waylandsink sync=false enable-last-sample=false fullscreen=true



Single Camera streaming over HDMI Display

Multi-stream playback

- Stream over two pipelines

$ gst-launch-1.0 -e qtiqmmfsrc name=camsrc0 camera=1 ! video/x-raw\(memory:GBM\), format=NV12, width=1920,height=1080,framerate=30/1 ! queue ! waylandsink name=d1 x=45 y=300 width=864 height=486 qtiqmmfsrc name=camsrc1 camera=0 ! video/x-raw\(memory:GBM\), format=NV12, width=1920,height=1080,framerate=30/1 ! queue ! waylandsink name=d2 x=1000 y=300 width=864 height=486

Dual Camera streaming over HDMI Display

13. Interface Validation Summary

Following interfaces are validated on QCS610 EVK,

1. UART Console(USB 2.0 Micro-B)
2. USB 3.1 Type-C
3. Ethernet
4. Wi-Fi
5. Bluetooth
6. Camera Connectors
7. Mini PCIe Connector
8. Micro-SD Card
9. USB2.0Type-A
10. HDMI

13.1 Ongoing developments

- Audio Codec
- LCD Module

13.2 Known bugs

- HDMI - Weston Stability Issue

The QC610-410EVK contains FCC ID: 2A8AKQC610-410EVK. This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation. This equipment complies with FCC RF radiation exposure limits set forth for an uncontrolled environment. This equipment should be installed and operated with a minimum distance of 20 centimeters between the radiator and your body.

Caution: Any changes or modifications not expressly approved by VVDN void the user's authority to operate the equipment.

FCC Interference Statement (Part 15.105 (b))

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.