

VVDN QCS 610 EVK User Guide

Revision History:

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

Table of Contents

1.	Introduction	5
2.	QCS610 + Carrier Board	5
2.1	Power Source Selection	6
3.	Debug Console (Micro B)	6
4.	USB 3.1 Type -C	7
5.	UART Console setup	7
6.	Powering up the system	9
7.	ADB setup:-	11
8.	DC Power Jack Details	12
9.	QDL & FASTBOOT Flashing Guide (Linux)	13
10.	Expansion Header details (H7)	17
11.	Mini PCIe Connector Details (J11)	18
12.	Performance validation of interfaces	19
12.1	Ethernet	19
12.2	Wi-Fi	20
12.3	Bluetooth	23
12.4	IMX462 Camera Board Connection to QCS610 EVK	25
12.4.1.	Prerequisites	26
12.4.2.	Steps for validating camera sensor - Using Recording pipeline	26
12.4.3.	Steps for validating camera sensor - Using streaming pipeline	27
12.4.3.1.	Streaming over USB	27
12.4.3.2.	Streaming over Network	28
12.4.3.3.	Streaming over HDMI Display	30
13.	Interface Validation Summary	32
13.1	Ongoing developments	32
13.2	Known bugs	32

List of Figures

1.	Introduction	5
2.	QCS610 + Carrier Board	5
2.1	Power Source Selection	6
3.	Debug Console (Micro B)	6
4.	USB 3.1 Type -C	7
5.	UART Console setup	7

6.	Powering up the system	9
7.	ADB setup:-	11
8.	DC Power Jack Details	12
9.	QDL & FASTBOOT Flashing Guide (Linux)	13
10.	Expansion Header details (H7)	17
11.	Mini PCIe Connector Details (J11)	18
12.	Performance validation of interfaces	19
12.1	Ethernet	19
12.2	Wi-Fi	20
12.3	Bluetooth	23
12.4	IMX462 Camera Board Connection to QCS610 EVK	25
12.4.1.	Prerequisites	26
12.4.2.	Steps for validating camera sensor - Using Recording pipeline	26
12.4.3.	Steps for validating camera sensor - Using streaming pipeline	27
12.4.3.1.	Streaming over USB	27
12.4.3.2.	Streaming over Network	28
12.4.3.3.	Streaming over HDMI Display	30
13.	Interface Validation Summary	32
13.1	Ongoing developments	32
13.2	Known bugs	32

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)

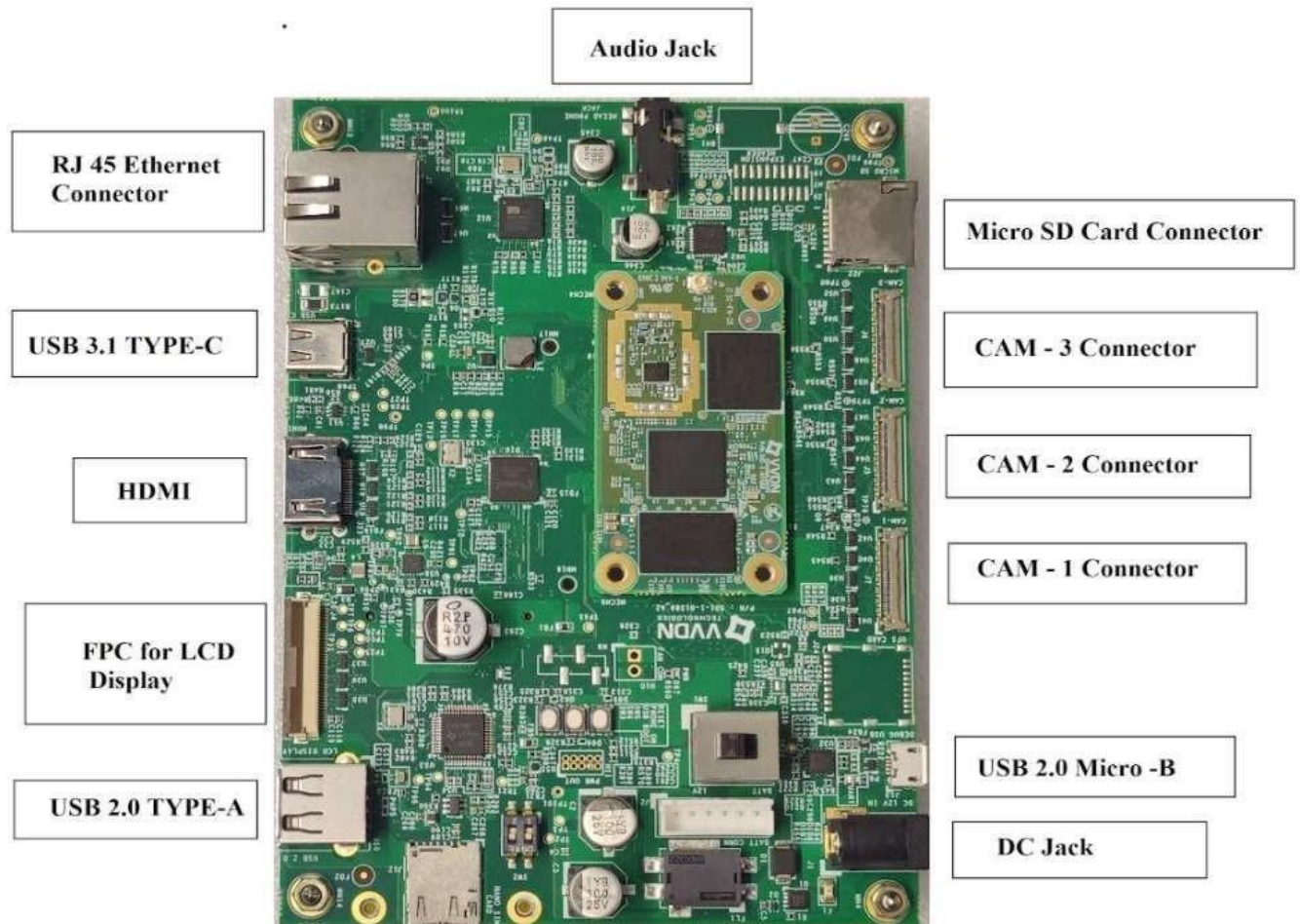


Figure 1: QCS610 SOM + Carrier Board

2.1 Power Source Selection

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

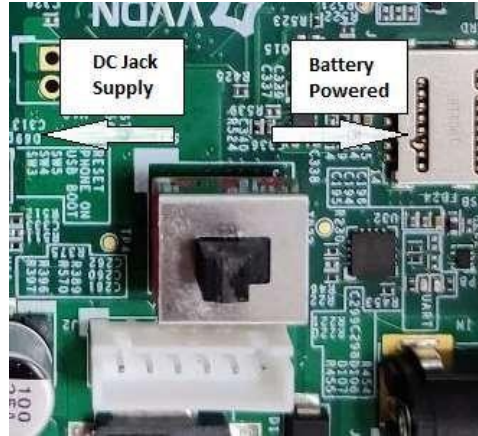


Figure 2: Position of Switch

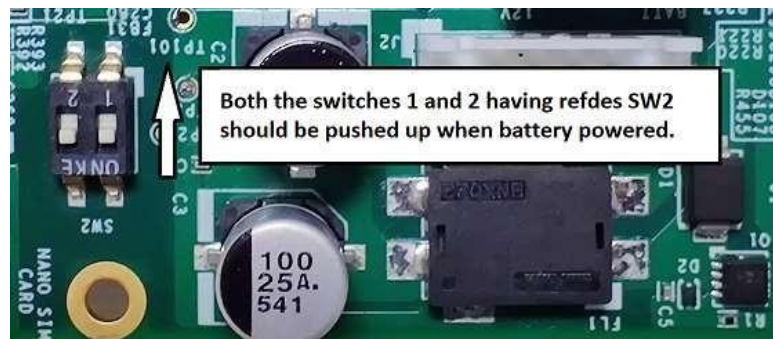


Figure 3: Positioning of Switch (SW2)

3. Debug Console (Micro B)

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

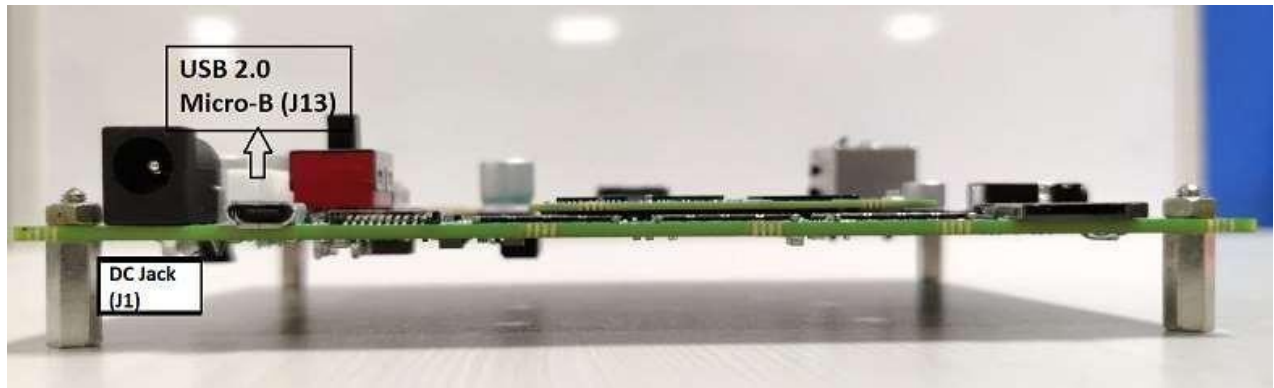


Figure 4: Micro-B (J13)

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.

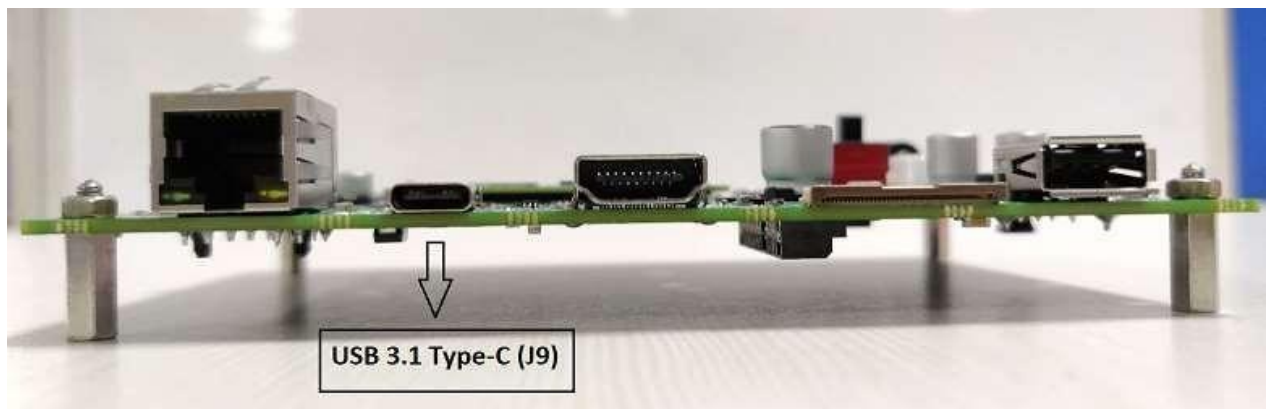


Figure 5: USB 3.1 Type-C (J9)

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:

```

+-----+
| A -   Serial Device       : /dev/ttyUSB0 |
| B - Lockfile Location    : /var/lock     |
| C -   Callin Program     :               |
| D -   Callout Program    :               |
| E -   Bps/Par/Bits       : 115200 8N1    |
| F - Hardware Flow Control : No           |
| G - Software Flow Control: Yes          |
|                                         |
| Change which setting? █                |
+-----+
| Screen and keyboard |
| Save setup as dfl   |
| Save setup as..     |
| Exit                |
| Exit from Minicom   |
+-----+

```

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

```

root@5955--SW--CHN:/home/vvdn/Music# dmesg | grep tty | tail -1
[74632.034250] usb 1-3: FTDI USB Serial Device converter now attached to ttyUSB0
root@5955--SW--CHN:/home/vvdn/Music# █

```

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.

```

root@5955--SW--CHN:/home/vvdn/Music# picocom -b 115200 /dev/ttyUSB0
picocom v1.7

port is           : /dev/ttyUSB0
flowcontrol       : none
baudrate is       : 115200
parity is         : none
databits are      : 8
escape is         : C-a
local echo is     : no
noinit is         : no
noreset is        : no
nolock is         : no
send_cnd is       : sr -vv
receive_cnd is    : rz -vv
imap is           :
omap is           :
enap is           : crcr1f,delbs,
Terminal ready

```

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.

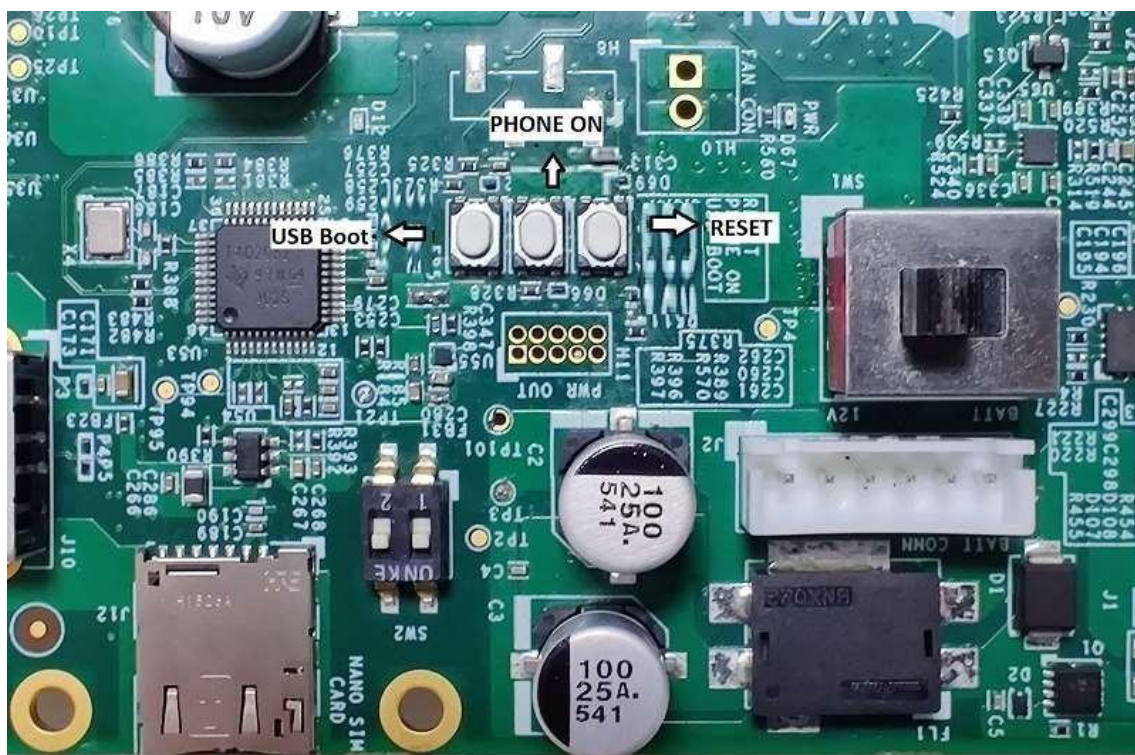


Figure 9: Phone ON Switch

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

```

Log Type: B - Since Boot(Power On Reset), D - Delta, S - Statistic
S: QX Image Version: STRONG-BOOT_V0.1.1-00007-3M110L2B-2
S: IMAGE VARIANT: STRONG-SOM110L2B-A
S: QEM Image Version: STRONG-ESOM
S: Boot Interface: efuse
S: Secure Boot: Off
S: Boot Config @ 0x00780078 = 0x00000001
S: TAG ID @ 0x00780130 = 0x10110001
S: DEM ID @ 0x00780130 = 0x00000000
S: Serial Number @ 0x00780130 = 0x02004700
S: QEM Config Row 0 @ 0x00780130 = 0x0000000000000000
S: QEM Config Row 1 @ 0x00780130 = 0x0000000000000000
S: Feature Config Row 0 @ 0x00780130 = 0x0000200010300000
S: Feature Config Row 1 @ 0x00780130 = 0x0010777700037777
S: Core 0 Frequency: 1510 MHz
S: PBL Patch Ver: 3
S: PBL Freq: 600 MHz
D: 5633 - pbl_apps_init_timestamp
D: 56871 - bootable_media_detect_timestamp
D: 1793 - bl_elf_metadata_loading_timestamp
D: 432 - bl_hash_seg_auth_timestamp
D: 54830 - bl_elf_loadable_segment_loading_timestamp
D: 4908 - bl_elf_seg_hash_verify_timestamp
D: 7009 - bl_sec_hash_seg_auth_timestamp
D: 421 - bl_sec_seg_hash_verify_timestamp
D: 39 - pbl_populate_shared_data_and_exit_timestamp
S: 133787 - PBL_End
D: 134393 - SBL1_Start
D: 277123 - SBL1_Hello @ 10:35:00 on Oct 27 2021
D: 277794 - usb: hs_phy_nondrive_start
D: 279808 - usb: hs_phy_nondrive_finish
D: 283467 - boot_flash_init_start
D: 30 - boot_flash_init_delta
D: 291001 - xblconfig_init_start
D: 978 - Auth_Metadata
D: 28243 - xblconfig_init_delta
D: 324520 - boot_config_data_table_init_start
D: 327997 - Using default CDT
D: 4636 - boot_config_data_table_init_delta - (0 Bytes)
D: 325774 - CDT_Versions.Platform ID:32, Major ID:0, Minor ID:0, subtype:0
D: 348676 - pm_device_init_start

```

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.

```

17.871493 sm6150-asoc-snd soc:qcom,msm-audio-apr:qcom,qcom-audio:sound: ASoc: no source widget found for MIC_BIAS3
17.882511 sm6150-asoc-snd soc:qcom,msm-audio-apr:qcom,qcom-audio:sound: ASoc: Failed to add route MIC_BIAS3 -> direct -> TX_DMIC2
17.894868 [drm-dp] dp_aux_cnd_fifo_tx: aux err: DP_AUX_ERR_TOUT
17.901438 sm6150-asoc-snd soc:qcom,msm-audio-apr:qcom,qcom-audio:sound: ASoc: no sink widget found for MIC_BIAS3
17.912284 sm6150-asoc-snd soc:qcom,msm-audio-apr:qcom,qcom-audio:sound: ASoc: Failed to add route Digital_Mic2 -> direct -> MIC_BIAS3
17.925294 sm6150-asoc-snd soc:qcom,msm-audio-apr:qcom,qcom-audio:sound: ASoc: no source widget found for MIC_BIAS3
17.936317 sm6150-asoc-snd soc:qcom,msm-audio-apr:qcom,qcom-audio:sound: ASoc: Failed to add route MIC_BIAS3 -> direct -> TX_DMIC3
17.948881 sm6150-asoc-snd soc:qcom,msm-audio-apr:qcom,qcom-audio:sound: ASoc: no sink widget found for MIC_BIAS3
17.959744 [drm-dp] dp_aux_cnd_fifo_tx: aux err: DP_AUX_ERR_TOUT
17.966858 sm6150-asoc-snd soc:qcom,msm-audio-apr:qcom,qcom-audio:sound: ASoc: Failed to add route Digital_Mic3 -> direct -> MIC_BIAS3
17.979251 sm6150-asoc-snd soc:qcom,msm-audio-apr:qcom,qcom-audio:sound: ASoc: no sink widget found for IN1_HPHL
17.990036 sm6150-asoc-snd soc:qcom,msm-audio-apr:qcom,qcom-audio:sound: ASoc: Failed to add route HP_HL_OUT -> direct -> IN1_HPHL
18.002475 sm6150-asoc-snd soc:qcom,msm-audio-apr:qcom,qcom-audio:sound: ASoc: no sink widget found for IN2_HPWR
18.013728 sm6150-asoc-snd soc:qcom,msm-audio-apr:qcom,qcom-audio:sound: ASoc: Failed to add route HPWR_OUT -> direct -> IN2_HPWR
18.025496 [drm-dp] dp_aux_cnd_fifo_tx: aux err: DP_AUX_ERR_TOUT
18.03997 sm6150-asoc-snd soc:qcom,msm-audio-apr:qcom,qcom-audio:sound: ASoc: no sink widget found for IN3_AUX
18.042669 sm6150-asoc-snd soc:qcom,msm-audio-apr:qcom,qcom-audio:sound: ASoc: Failed to add route AUX_OUT -> direct -> IN3_AUX
18.054958 sm6150-asoc-snd soc:qcom,msm-audio-apr:qcom,qcom-audio:sound: ASoc: no source widget found for ADC1_OUTPUT
18.066153 sm6150-asoc-snd soc:qcom,msm-audio-apr:qcom,qcom-audio:sound: ASoc: Failed to add route ADC1_OUTPUT -> direct -> TX_SMB_ADC0
18.079146 sm6150-asoc-snd soc:qcom,msm-audio-apr:qcom,qcom-audio:sound: ASoc: no source widget found for ADC2_OUTPUT
18.090381 [drm-dp] dp_aux_cnd_fifo_tx: aux err: DP_AUX_ERR_TOUT
18.096649 sm6150-asoc-snd soc:qcom,msm-audio-apr:qcom,qcom-audio:sound: ASoc: Failed to add route ADC2_OUTPUT -> direct -> TX_SMB_ADC2
18.109975 sm6150-asoc-snd soc:qcom,msm-audio-apr:qcom,qcom-audio:sound: ASoc: no sink widget found for SpkLeft_IN
18.121007 sm6150-asoc-snd soc:qcom,msm-audio-apr:qcom,qcom-audio:sound: ASoc: Failed to add route WSA_SPK1_OUT -> direct -> SpkLeft_IN
18.134073 sm6150-asoc-snd soc:qcom,msm-audio-apr:qcom,qcom-audio:sound: ASoc: no sink widget found for SpkRight_IN
18.145391 sm6150-asoc-snd soc:qcom,msm-audio-apr:qcom,qcom-audio:sound: ASoc: Failed to add route WSA_SPK2_OUT -> direct -> SpkRight_IN
18.158156 [drm-dp] dp_aux_cnd_fifo_tx: aux err: DP_AUX_ERR_TOUT
Starting Save/Restore Sound card state e...
18.314159 sm6150-asoc-snd soc:qcom,msm-audio-apr:qcom,qcom-audio:sound: Sound card sm6150-ldp-snd-card registered
18.316842 type=1400 audit(1633325429.632:251): avc: denied { getattr } for pid=532 comm="systemd-journald" scontext=system_u:system_r:syslog
18.325334 sm6150-asoc-snd soc:qcom,msm-audio-apr:qcom,qcom-audio:sound: soc:qcom,msm-audio-apr:qcom,qcom-audio:sound supply pa-vddio r
18.347842 type=1400 audit(1633325429.632:252): avc: denied { read } for pid=532 comm="systemd-journald" scontext=system_u:system_r:syslog

QTI Linux full reference stack distro with debug capabilities. 1430 qcs610-odk-64 ttyMSM0.

qcs610-odk-64 login: root
Password:
- #
- #
- #

```

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. adbroot
2. adbshell

The figure below show the obtained terminal and login

```

vvdn@5955--SW--CHN:~$ adb wait-for-device
vvdn@5955--SW--CHN:~$ adb devices
List of devices attached
4b108fd3          device

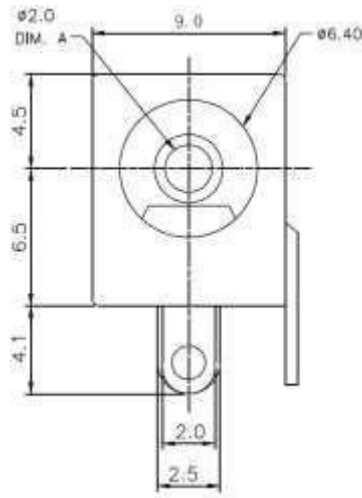
vvdn@5955--SW--CHN:~$ adb root
vvdn@5955--SW--CHN:~$ adb shell
/ #
/ #

```

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

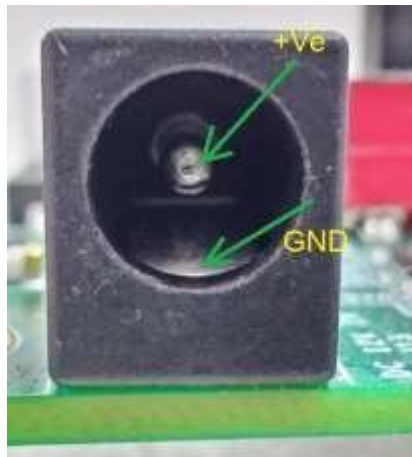


Figure 13: DC Power Jack Details

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)

```
adevi8@ESOM:~/VVDN_ESOM/QFIL/qdl$ ls
firehose.c LICENSE patch.c patch.o program.h qdl qdl.h README sahara.o ufs.h util.c
firehose.o Makefile patch.h program.c program.o qdl.c qdl.o sahara.c ufs.c ufs.o util.o
adevi8@ESOM:~/VVDN_ESOM/QFIL/qdl$
```

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

```
cnlib.nbn storsec.mbn userdata_11.img userdata_26.img userdata_40.img userdata_55.img userdata_6.img
devcfg.mbn system_10.img userdata_12.img userdata_27.img userdata_41.img userdata_56.img userdata_70.img
dspso.bin system_11.img userdata_13.img userdata_28.img userdata_42.img userdata_57.img userdata_71.img
dtbo.img system_12.img userdata_14.img userdata_29.img userdata_43.img userdata_58.img userdata_72.img
gpt_backup0.bin system_1.img userdata_15.img userdata_2.img userdata_44.img userdata_59.img userdata_73.img
gpt_main0.bin system_2.img userdata_16.img userdata_30.img userdata_45.img userdata_60.img userdata_74.img
hyp.mbn system_3.img userdata_17.img userdata_31.img userdata_46.img userdata_61.img userdata_75.img
kn4.mbn system_4.img userdata_18.img userdata_32.img userdata_47.img userdata_62.img userdata_76.img
NON-HLOS.bin system_5.img userdata_19.img userdata_33.img userdata_48.img userdata_63.img userdata_77.img
patch0.xml system_6.img userdata_20.img userdata_34.img userdata_49.img userdata_64.img userdata_78.img
persist_1.img system_7.img userdata_21.img userdata_35.img userdata_50.img userdata_65.img userdata_79.img
adevi8@ESOM:~/esom_qfil/emmc$
```

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**

```
vvdn@vvsal-Latitude-3400:~$ adb devices
List of devices attached
4a909a4b device
```

```
vvdn@vvsal-Latitude-3400:~$
```

(or)

- **lsusb**

```
vvdn@vvsal-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
```

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@vvs-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@vvs-Latitude-3400:~$ adb root
restarting adbd as root
vvdn@vvs-Latitude-3400:~$ adb reboot edl
vvdn@vvs-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_unsparsed0.xml patch0.xml
```

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

```
root@vvs-Latitude-3400:/home/vvdn/Documents/VVDN_ESDM/Images/emmc# sudo ./qdl --debug --storage emmc prog_firehose_ddr.elf rawprogram_unsparsed0.xml patch0.xml
Waiting for EDL device
```

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

```
root@vvs-Latitude-3400:/home/vvdn/Documents/VVDN_ESDM/Images/emmc# sudo ./qdl --debug --storage emmc prog_firehose_ddr.elf rawprogram_unsparsed0.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: 0x30
READ64 image: 13 offset: 0x1000 length: 0x1000
READ64 image: 13 offset: 0x2000 length: 0xcd8
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
vkchlt0459@7870--VISN--KOCHI1:~/Esom_fastboot$ |
```

After flashing, restart the device.

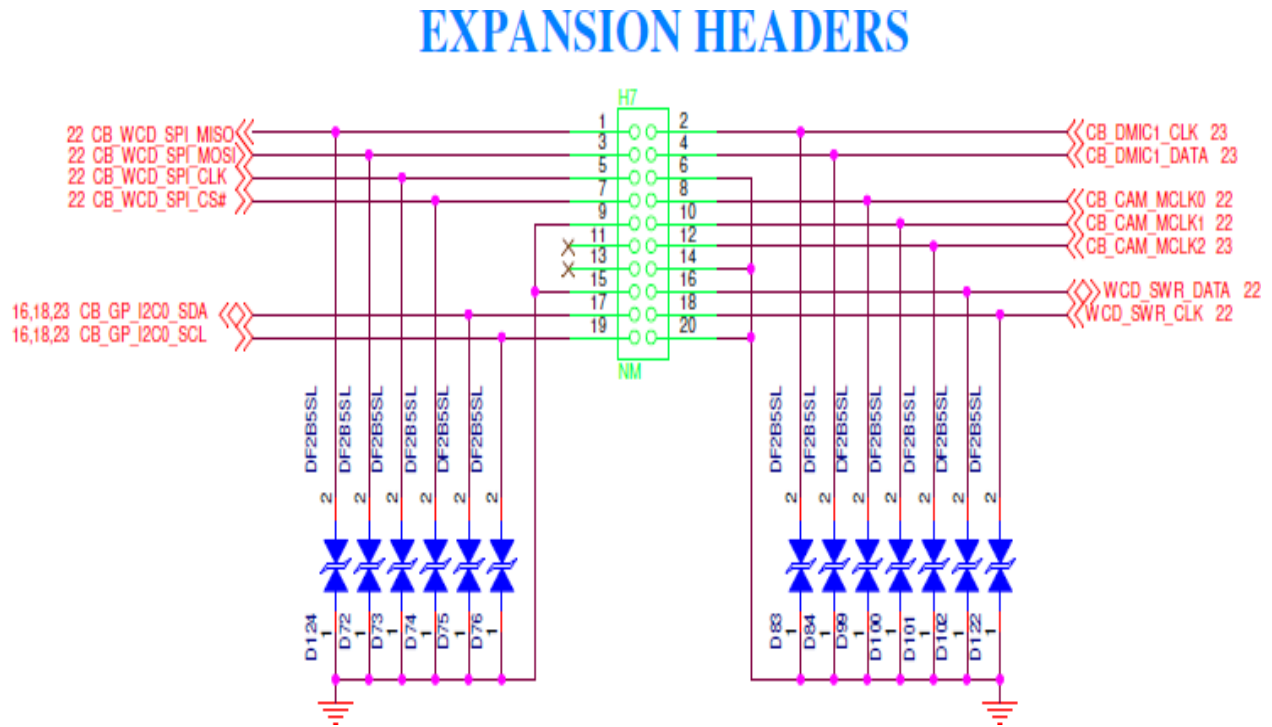
9.3 QFIL Flashing (using Windows PC)

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)



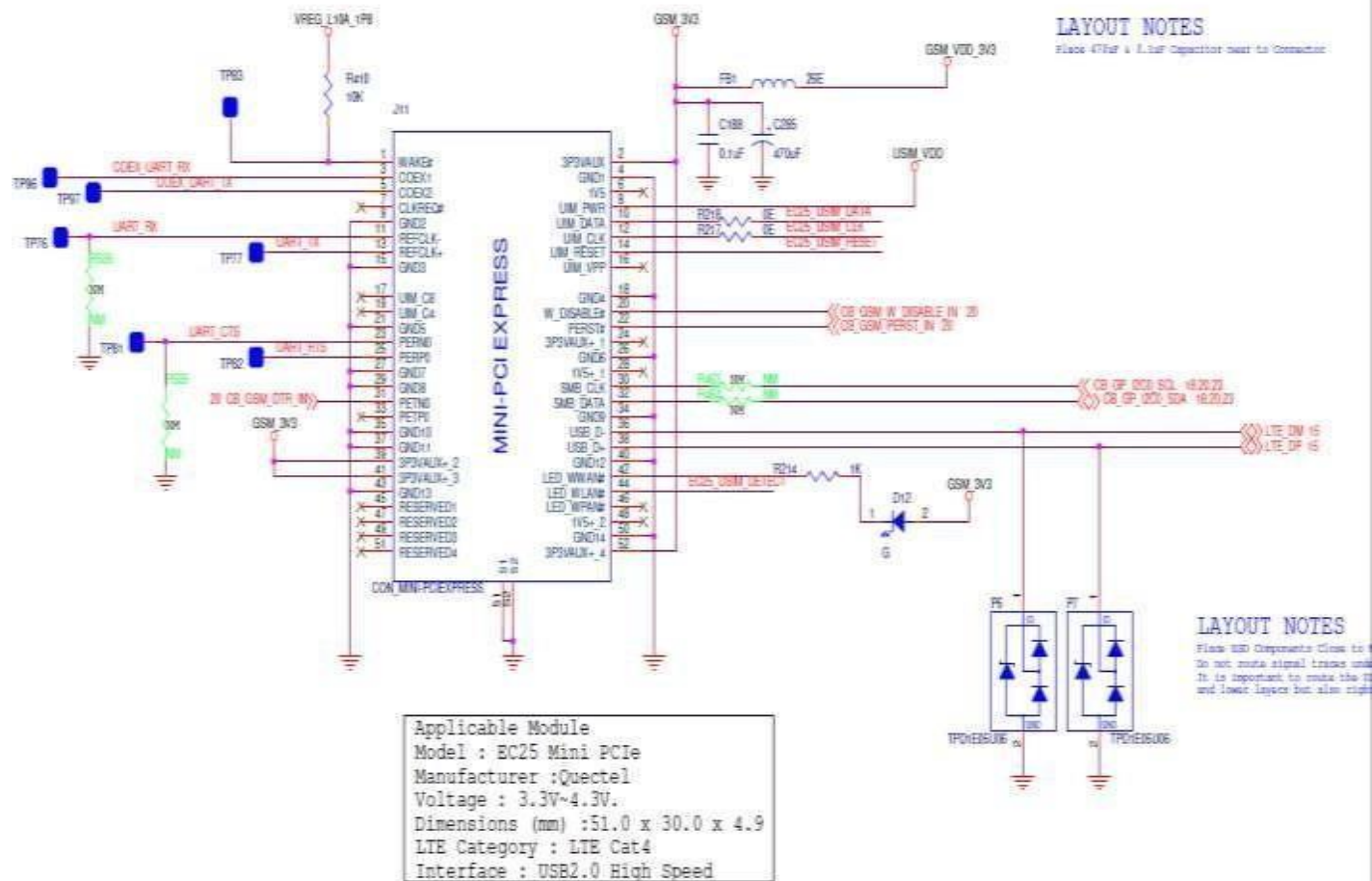
All these signals are coming under 1.8V IO level.

Pullups 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 the test.

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

```
/ # iperf3 -s
.....
Server listening on 5201
.....
Accepted connection from 192.168.1.5, port 50348
[ 5] local 192.168.1.10 port 5201 connected to 192.168.1.5 port 50350
[ ID] Interval      Transfer    Bandwidth
[ 5] 0.00-1.00    sec    97.8 MBytes  819 Mbits/sec
[ 5] 1.00-2.00    sec    99.0 MBytes  833 Mbits/sec
[ 5] 2.00-3.00    sec    98.5 MBytes  825 Mbits/sec
[ 5] 3.00-4.00    sec    99.4 MBytes  835 Mbits/sec
```

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

```
VKCH10452@7881--VISH--KUCH11: $ iperf3 -s
.....
Server listening on 5201
.....
Accepted connection from 192.168.1.10, port 42157
[ 5] local 192.168.1.5 port 5201 connected to 192.168.1.10 port 42158
[ ID] Interval      Transfer    Bitrate
[ 5] 0.00-1.00    sec    106 MBytes  886 Mbits/sec
[ 5] 1.00-2.00    sec    112 MBytes  941 Mbits/sec
[ 5] 2.00-3.00    sec    112 MBytes  941 Mbits/sec
[ 5] 3.00-4.00    sec    112 MBytes  941 Mbits/sec
[ 5] 4.00-5.00    sec    112 MBytes  941 Mbits/sec
[ 5] 5.00-6.00    sec    112 MBytes  941 Mbits/sec
[ 5] 6.00-7.00    sec    112 MBytes  941 Mbits/sec
[ 5] 7.00-8.00    sec    112 MBytes  941 Mbits/sec
```

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 ! tcpserver sink 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.

- ScanNearby Access 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 -dd #
```

```
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 -ddddd &
```

- 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
- Execute the following command to check the status of hostapd

```
#hostapd_cli -i wlan0
Streaming over Wi-Fi
```

12.2.3

- 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 ! tcpserver sink 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

```
# iperf3 -s
Server listening on 5201
Accepted connection from 192.168.43.91, port 43004
[ 5] local 192.168.43.1 port 5201 connected to 192.168.43.91 port 43004
[ ID] Interval      Transfer      Bandwidth
[ 5] 0.00-1.00 sec  4.12 MBytes  34.6 Mbits/sec
[ 5] 1.00-2.00 sec  3.27 MBytes  27.9 Mbits/sec
[ 5] 2.00-3.00 sec  4.20 MBytes  35.2 Mbits/sec
[ 5] 3.00-4.00 sec  4.00 MBytes  33.7 Mbits/sec
[ 5] 4.00-5.00 sec  4.42 MBytes  37.1 Mbits/sec
[ 5] 5.00-6.00 sec  4.00 MBytes  33.6 Mbits/sec
[ 5] 6.00-7.00 sec  4.23 MBytes  35.6 Mbits/sec
[ 5] 7.00-8.00 sec  4.18 MBytes  34.9 Mbits/sec
[ 5] 8.00-9.00 sec  4.37 MBytes  36.6 Mbits/sec
[ 5] 9.00-10.00 sec  4.19 MBytes  35.2 Mbits/sec
[ 5] 10.00-10.00 sec  100 KBytes  0.9 Mbits/sec
[ ID] Interval      Transfer      Bandwidth
[ 5] 0.00-10.00 sec  42.0 MBytes  35.0 Mbits/sec
sender
receiver
```

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

```
vkchit0459q787b@-VISHU-KOCHI:~/Desktop/SW/VVDN_ESOM_1/FCC_Certifications$ iperf3 -s
Server listening on 5201
Accepted connection from 192.168.43.1, port 38236
[ 5] local 192.168.43.91 port 5201 connected to 192.168.43.1 port 38236
[ ID] Interval      Transfer      Bitrate
[ 5] 0.00-1.00 sec  2.31 MBytes  19.4 Mbits/sec
[ 5] 1.00-2.00 sec  2.46 MBytes  20.6 Mbits/sec
[ 5] 2.00-3.00 sec  2.58 MBytes  21.7 Mbits/sec
[ 5] 3.00-4.00 sec  2.44 MBytes  20.1 Mbits/sec
[ 5] 4.00-5.00 sec  2.32 MBytes  19.1 Mbits/sec
[ 5] 5.00-6.00 sec  2.36 MBytes  19.4 Mbits/sec
[ 5] 6.00-7.00 sec  2.75 MBytes  23.1 Mbits/sec
[ 5] 7.00-8.00 sec  2.33 MBytes  19.5 Mbits/sec
[ 5] 8.00-9.00 sec  2.14 MBytes  17.9 Mbits/sec
[ 5] 9.00-10.00 sec  716 KBytes  5.96 Mbits/sec
[ 5] 10.00-10.00 sec  360 KBytes  3.00 Mbits/sec
[ ID] Interval      Transfer      Bitrate
[ 5] 0.00-10.00 sec  18.8 MBytes  15.2 Mbits/sec
receiver
```

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 112233445566778899f0
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
current State = 1, new state = 0 killall:
qcbtdaemon: no process killed killall:
wcnsfilter: 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/camoverridesettings.txt
adb shell "echo IFEDualClockThreshold=600000000 >> /etc/camera/camoverridesettings.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 ! tcpserver sink 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

```
vvdn-admin@2273--SW--KOCHI1:~$ adb shell
/ #
/ #
/ # source /etc/gstreamer1.0/set_gst_env.sh
/ #
/ # gst-launch-1.0 -e qtiqmmfsrc name=qmmf ! video/x-raw(memory:GBM\), width=19
20, height=1080, framerate=30/1,format=NV12 ! omxh264enc target-bitrate=600000 c
ontrol-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 ! tcpse
rversink port=8900 host=127.0.0.1
gbm_create_device(156): Info: backend name is: msm_drm
Setting pipeline to PAUSED ...
gbm_create_device(156): Info: backend name is: msm_drm
Pipeline is live and does not need PREROLL ...
Setting pipeline to PLAYING ...
New clock: GstSystemClock
```

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 ! tcpserver sink 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 ! tcpserver sink 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`

```
vvdn-admin@2273--5W--K0CHI1:~$ adb root
vvdn-admin@2273--5W--K0CHI1:~$ adb shell
/ # 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 ! tcpserver sink 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 ! tcpserver sink port=8901 host=127.0.0.1
gbm_create_device(156): Info: backend name is: nsm_drm
gbm_create_device(156): Info: backend name is: nsm_drm
Setting pipeline to PAUSED ...
gbm_create_device(156): Info: backend name is: nsm_drm
gbm_create_device(156): Info: backend name is: nsm_drm
Pipeline is live and does not need PREROLL ...
Setting pipeline to PLAYING ...
New clock: GstSystemClock
```

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 ! tcpserver sink 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!

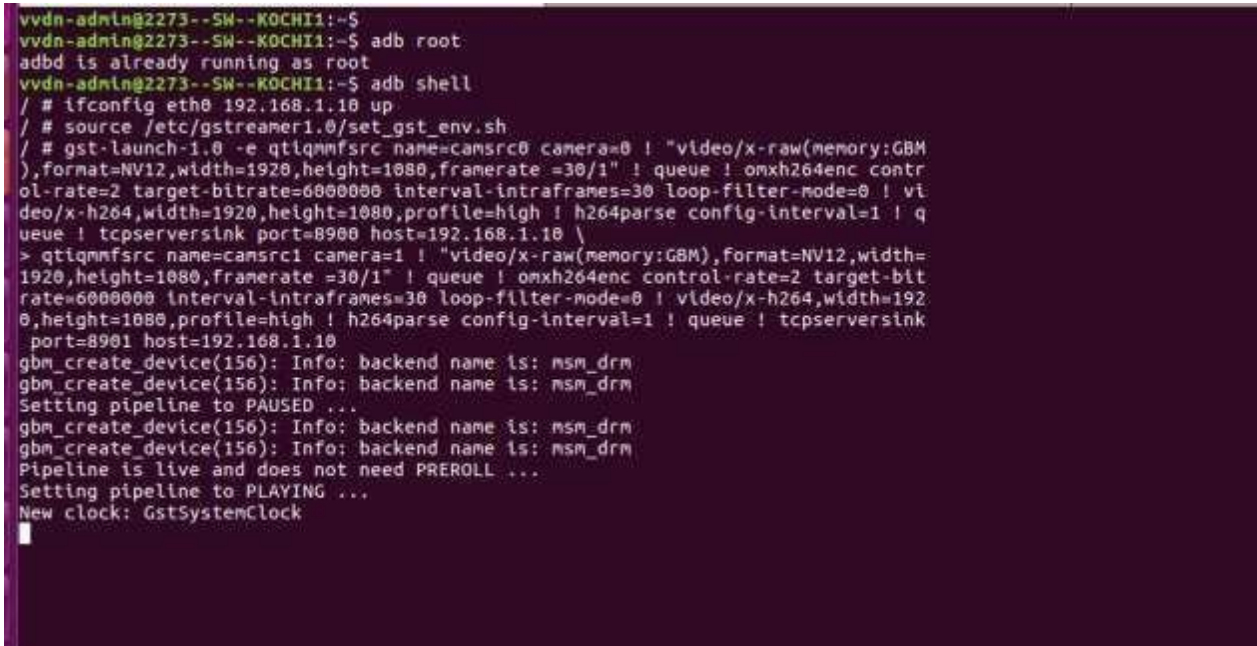
```
vvdn-admin@2273--SW--KOCHI1:~$
vvdn-admin@2273--SW--KOCHI1:~$ adb root
vvdn-admin@2273--SW--KOCHI1:~$ adb shell
/ #
/ #
/ # ifconfig eth0 192.168.1.10 up
/ # source /etc/gstreamer1.0/set_gst_env.sh
/ # gst-launch-1.0 -e qtiqmmfsrc name=qmmf ! video/x-raw(memory:GBM), width=1920, height=1080, framerate=30/1,format=NV12 ! 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 ! tcpserver sink port=8900 host=192.168.1.10
gbm_create_device(156): Info: backend name is: msm_drm
Setting pipeline to PAUSED ...
gbm_create_device(156): Info: backend name is: msm_drm
Pipeline is live and does not need PREROLL ...
Setting pipeline to PLAYING ...
New clock: GstSystemClock
```

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 ! tcpserver sink
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 ! tcpserver sink
port=8901 host=<HOST_IP>
```



```
vvdn-admin@2273--SW--K0CHI1:~$
vvdn-admin@2273--SW--K0CHI1:~$ adb root
adb is already running as root
vvdn-admin@2273--SW--K0CHI1:~$ adb shell
/ # ifconfig eth0 192.168.1.10 up
/ # 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 contr
ol-rate=2 target-bitrate=6000000 interval-intraframes=30 loop-filter-mode=0 ! vi
deo/x-h264,width=1920,height=1080,profile=high ! h264parse config-interval=1 ! q
ueue ! tcpserver sink port=8900 host=192.168.1.10 \
> 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-bit
rate=6000000 interval-intraframes=30 loop-filter-mode=0 ! video/x-h264,width=192
0,height=1080,profile=high ! h264parse config-interval=1 ! queue ! tcpserver sink
port=8901 host=192.168.1.10
gbm_create_device(156): Info: backend name is: msm_drm
gbm_create_device(156): Info: backend name is: msm_drm
Setting pipeline to PAUSED ...
gbm_create_device(156): Info: backend name is: msm_drm
gbm_create_device(156): Info: backend name is: msm_drm
Pipeline is live and does not need PREROLL ...
Setting pipeline to PLAYING ...
New clock: GstSystemClock
```

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
```

```
vvdn-admin@2273:~$ adb root
vvdn-admin@2273:~$ adb shell
/ # source /etc/gstreamer-0.0/set_gst_env.sh
/ # export XDG_RUNTIME_DIR=/run/user/0
/ # gst-launch-1.0 -e qtiqmmfsrc name=camsrc ! video/x-raw(memory:GBM),format=
NV12,width=1920,height=1080,framerate=30/1 ! queue ! waylandsink sync=false enab
le-last-sample=false fullscreen=true
gdn_create_device(156): Info: backend name is: msm_drm
Setting pipeline to PAUSED ...
gdn_create_device(156): Info: backend name is: msm_drm

** (gst-launch-1.0:3299): WARNING **: 00:02:15.340: Wayland compositor is missing the ability to scale, video display may not work properly. -
No viewporter present

** (gst-launch-1.0:3299): WARNING **: 00:02:15.340: Could not bind to zwf_linux_dnabuf_v1
Pipeline is live and does not need PREROLL ...
Setting pipeline to PLAYING ...
New clock: GstSystemClock
^CHandling interrupt...
Interrupt: Stopping pipeline ...
EOS on shutdown enabled -- Forcing EOS on the pipeline
Waiting for EOS...
Got EOS from element "pipeline0".
EOS received - stopping pipeline...
Execution ended after 0:00:07.071471143
Setting pipeline to PAUSED ...
Setting pipeline to READY ...
Setting pipeline to NULL ...
Freeing pipeline ...
/ #
```

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
```

```
vvdn-admin@2273:~$ adb root
adb is already running as root
vvdn-admin@2273:~$ adb shell
/ # source /etc/gstreamer-0.0/set_gst_env.sh
/ # export XDG_RUNTIME_DIR=/run/user/0
/ # 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 nam
e=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
gdn_create_device(156): Info: backend name is: msm_drm
gdn_create_device(156): Info: backend name is: msm_drm
Setting pipeline to PAUSED ...
gdn_create_device(156): Info: backend name is: msm_drm

** (gst-launch-1.0:3272): WARNING **: 00:02:17.237: Wayland compositor is missing the ability to scale, video display may not work properly. -
No viewporter present

** (gst-launch-1.0:3272): WARNING **: 00:02:17.237: Could not bind to zwf_linux_dnabuf_v1
gdn_create_device(156): Info: backend name is: msm_drm

** (gst-launch-1.0:3272): WARNING **: 00:02:17.238: Wayland compositor is missing the ability to scale, video display may not work properly. -
No viewporter present

** (gst-launch-1.0:3272): WARNING **: 00:02:17.238: Could not bind to zwf_linux_dnabuf_v1
Pipeline is live and does not need PREROLL ...
Setting pipeline to PLAYING ...
New clock: GstSystemClock
```

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. USB2.0 Type-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.