MISTRAL
an AXISCADES company

MRD5165 Platform

Software User Guide

Revision History

Revision	Date	Description
0.1	Nov 28, 2023	Preliminary version
0.2	17 January 2024	Added Section-8
1.0	08 Aug 2024	Baseline Version
1.1	09 Aug 2024	Added Fan Connector Section
1.2	24 Oct 2024	5G module section updated
1.3	16 Jan 2025	Added 5G module USB mode setup instructions

TABLE OF CONTENTS

TABLE OF FIGURES

Privacy Information

Mistral Solution Private Limited. and/or its affiliated companies – May Contain Trade Secrets and information sensitive to restricted audience.

The content of this document is intended solely for the use of the individual or entity to whom it is addressed and may contain privileged or confidential information. If you are not the intended recipient, you are hereby notified that any dissemination, distribution, copying, or unauthorized use of the information contained herein is strictly prohibited.

By accessing or reviewing this document, you agree to the following.

- To treat its contents with the utmost confidentiality and to take all necessary precautions to prevent unauthorized disclosure. Any unauthorized use, disclosure, or distribution of the information in this document may result in legal action and may be subject to applicable laws.
- Not to engage in or support any activities that are illegal or harmful. Any misuse of this document for purposes contrary to the law or public safety is strictly prohibited.

Mistral Solutions disclaims any responsibility for the consequences of unauthorized access or disclosure of the information contained in this document. Further to this, Mistral solutions disclaims any responsibility for the misuse or illegal distribution of this document. Individuals or entities found to be using or distributing this document for unlawful or harmful activities may be subject to legal action.

If you have received this document in error, please notify the sender immediately and delete the original message and all copies from your system.

Confidential Distribution: Use or distribution of this item, in whole or in part, is prohibited except as expressly permitted by written agreement(s) and/or terms with Mistral Solutions and/or its subsidiaries.

Not to be used, copied, reproduced, or modified in whole or in part, nor its contents revealed in any manner to others without the express written permission of Mistral Solution Private Limited.

For any further information/clarifications please contact

Mistral Solutions Private Limited,

#60 Adarsh Regent, 100 Ft. Ring Road,

Domlur Extension, Bengaluru 560071

Karnataka, India

1. Introduction

The MRD5165 platforms (Eagle – Kit & Edge AI – Box) from Mistral are compact, high-performance edge AI Engine built around the Qualcomm QRB5165 SoC, providing 15 TOPS (Trillion Operations Per Second) of artificial intelligence (AI) performance. This enables efficient processing of complex AI and deep learning workloads for on-device edge inferencing. It is designed to deliver high-speed wireless connectivity and high-accuracy AI and machine-learning inferencing technology, accelerating the development of innovative, power-efficient, high-computing robots for enterprise, industrial, and professional service applications.

Packed with cutting-edge technology, this kit empowers developers to create powerful drone solutions. Experience exceptional compute performance, seamless AI integration, and precision with computer vision capabilities.

The MRD5165 platforms include MRD5165 SoM, the core of the Edge AI Box, is a high-compute, AI-enabled, low-power processor with 8GB LPDDR5 PoP memory, 128GB UFS storage, a dedicated high-performance Computer Vision Engine for video analytics, on-board wireless connectivity, and multiple PMICs for power supplies. The MRD5165 platform integrates a powerful Image Signal Processor and provides various peripheral connections such as 6 MIPI CSI cameras, an HDMI Camera, USB 3.0 ports, a Micro USB port for debugging, and a Gigabit Ethernet port for wired connectivity. It also integrates an expandable SD card slot for additional storage.

The MRD5165 Eagle Kit from Mistral is an advanced drone controller built around the Qualcomm QRB5165 SoC and CubePilot's Cube Orange+. Designed with power efficiency in mind, it ensures longer flight times without compromising performance. Explore the skies and revolutionize the drone industry with our optimized development kit – your gateway to a new era of airborne innovation. The MRD5165 Eagle kit delivers high-speed wireless connectivity and high-accuracy artificial intelligence (AI) and machine-learning inferencing technology to facilitate accelerated development of innovative, power-efficient, high-computing robots and drones for enterprise, industrial, and professional service applications.

The MRD5165 Edge AI – box is highly integrated, modular form-factor design tailored for robotics developers, drone manufacturers, and system integrators, empowering users to build intelligent machines customized to their requirements. The kit is meticulously engineered for effortless integration directly into your designs!

This MRD5165 platform also includes multiple software options including support for embedded Linux and a reference Root File-System from Ubuntu. The kit supports Linux Kernel 5.4, Ubuntu, and ROS; and includes a suite of software packages for implementing various applications such as AI/ML, Neural Processing, Auto Pilot, Navigation, Machine Vision, Multimedia and User Interface among others.

This document provides the instructions to setting up the Eagle-Kit / Edge AI-Box and procedure to quickly validate all the peripherals of the platform features.

2. Platform Overview

Eagle Kit Overview

The Mistral MRD5165 Eagle Kit is based on Qualcomm's QRB5165 processor alongside an independent Flight Controller Unit from CubePilot. The MRD5165 Eagle kit provides high-performance compute platform for precise artificial intelligence (AI) and machine-learning inferencing technology to facilitate the accelerated development of innovative, power-efficient, high-computing robots and drones for enterprise, industrial, and professional service applications. It offers readily deployable SDKs, Ubuntu Root-FS and tools for product development to facilitate quick prototyping and proof-of-concept evaluations. Additionally, this kit includes high performance connectivity interfaces making it an ideal platform for connected device in IOT, Drones and Robotics.

The CubePilot provides an independent Microcontroller to perform critical flight control functions including the motor control, telemetry and other flight stabilization functions.

Figure 1 MRD5165 EAGLE KIT

Figure 2 MRD5165 EAGLE KIT LR WIFI

The Mistral MRD5165 Eagle Kit offers a versatile platform for product developers to evaluate system functionality, experiment with sample applications, migrate existing applications, create new features, and integrate with a variety of peripheral devices. This kit provides a Linux software environment for application development, with the option to customize and update the system software using a chosen release of the QRB5165 system software.

The Development Kit Software User Guide provides an overall description of the hardware and software for this platform and includes instructions for setting up the platform and validating all the peripherals.

Edge AI-Box Overview

Sharing the same compute platform as the Eagle-Kit, the MRD5165 Edge AI-Box provides high-performance compute platform for precise artificial intelligence (AI) and machine-learning inferencing technology to facilitate the accelerated development of innovative, power-efficient, high-computing robots. The compact form-factor is conducive to its usage in the industrial environment for machine vision and other AI/ML/DL work-loads.

Figure 3 MRD5165 EDGE AI BOX

Platform Accessories

Basic Accessories:

1. DC Adapter
2. Power Chord
3. Custom power cable
4. I-PEX cable
5. Camera Adapter Board
6. Wi-Fi Antenna
7. USB Type-C Cable

Vision Accessories:

1. IMX577 Camera Module and corresponding Adapter Board
2. OV9282 Camera Module and corresponding Adapter Board

Connectivity Accessories:

1. 5G modem
2. 5G antenna

Note: USB Micro-B cable and Ethernet cable are not part of the kit accessories.

3. Platform Setup

Figure 4 MRD5165 EAGLE KIT HOST PC Connectors

1. Connect the DC adapter using Power cable to J1
2. Connect a USB Type-C cable to J103 (adb connection)
3. Connect a Micro-B cable to J5 (Debug console)

Powering up Platform

The platform is Pre-Flashed with the Linux Ubuntu software.

1. Connect the DC adapter using custom power cable to Power Connector J1 on the MRD5165 Platform.
2. Connect a Type-C cable to USB0 Type-C port J103 of the MRD5165 Platform for adb access.
3. Connect a Micro-B cable to Micro-B port J5 of the MRD5165 Platform for debug console access.
4. Power up the setup

Refer to the "Error! Reference source not found." and "Error! Reference source not found." for the setup.

Host PC setup

- Windows Host PC adb and fast-boot installation
  Download platform tools for Windows from the link provided below.

  https://developer.android.com/tools/releases/platform-tools

- Ubuntu Host PC adb and fast-boot installation
  Use the commands below to install adb and fast-boot.

```
$ sudo apt-get update

$ sudo apt-get install android-tools-adb android-tools-fastboot
```

Serial/Debug Console

- Connect the USB cable to Micro-B connector on the platform to Host PC as shown in Error! Reference source not found..
- Open a Tera Term application on Host PC to observe boot logs and access the board.

  a) Select appropriate Serial Console (Micro-B connection) COM port.
  b) Set Baud Rate to 115200–8-N-1

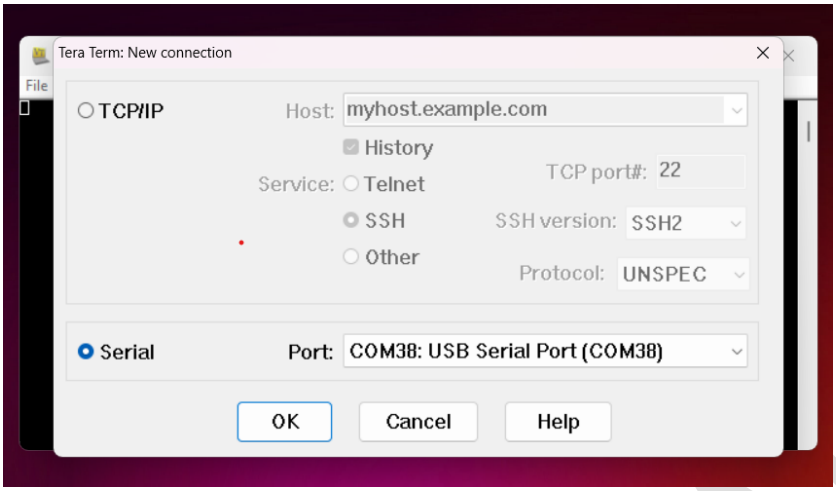

Figure 5 Serial COM port selection

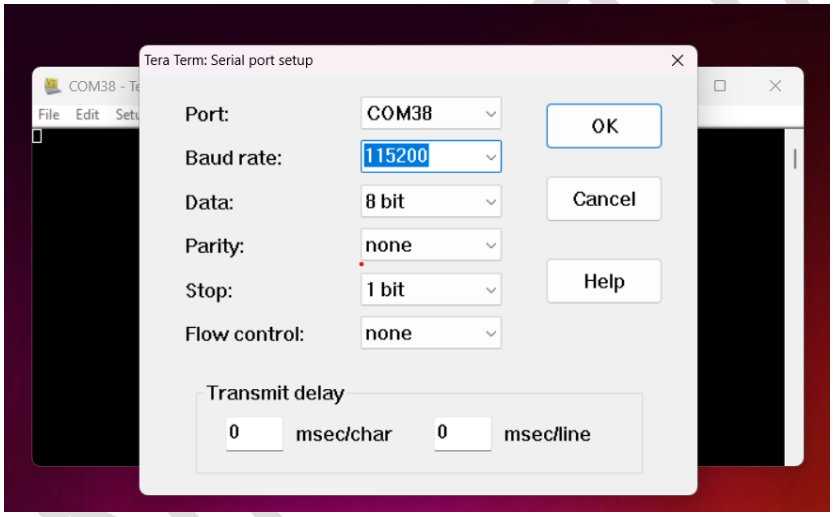

Figure 6 Serial Baud rate set and settings for 115200 -8N1

Figure 7 MRD5165 Platform Boot Logs

```
Serial Console Login:
Username: root
Password: oelinux123
```

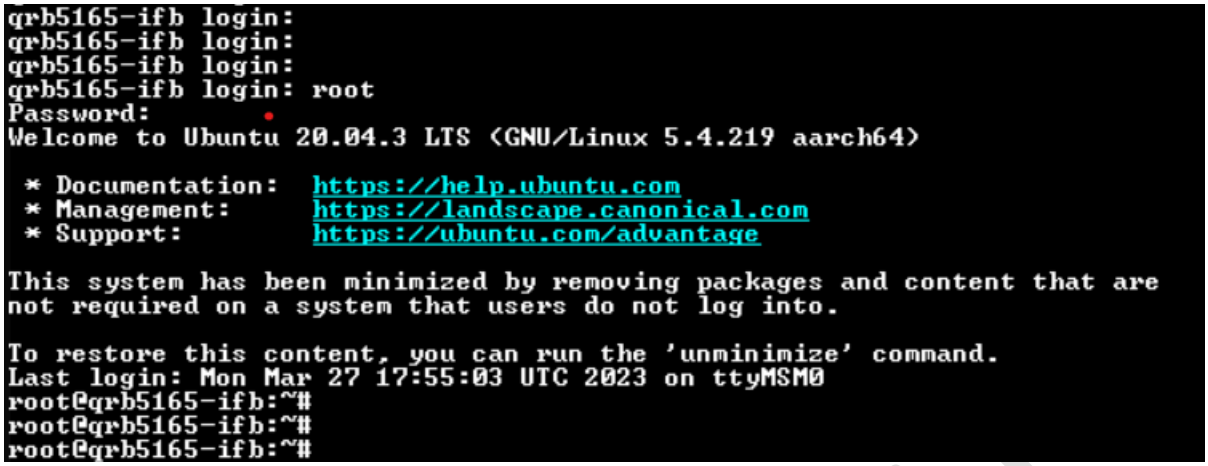

Figure 8 MRD5165 Platform Serial Console Login

Setting up ADB over Type-C

- Connect the USB Type-C cable to platform Type-C connector & Host PC as shown in Error! Reference source not found..
- Power up the platform
- Open a CMD terminal on Host PC and run the following commands.

```
$ adb root
$ adb shell
```

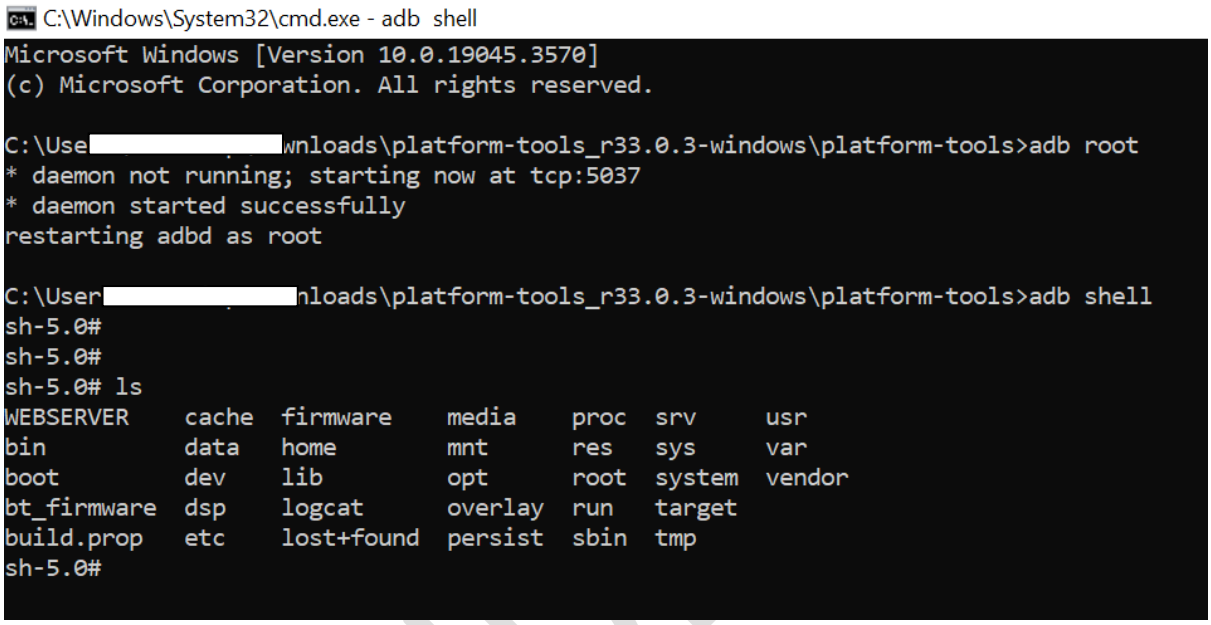

Figure 9 MRD5165 Platfrom ADB Type-C connection

Setting up ADB over Wi-Fi

The Platform device offers a hotspot connection, requiring the Host PC to establish a Wi-Fi connection.

- Connect Host PC Wi-Fi to the hotspot hosted by the MRD5165 Platform. (SSID "MRD5165-XXXXXXXX)
- Open a CMD terminal on Host PC and run the following commands.

```
$ adb connect 192.168.2.1:5555
$ adb root & adb shell
```

Figure 10 MRD5165 Platform Hotspot Wi-Fi Connection

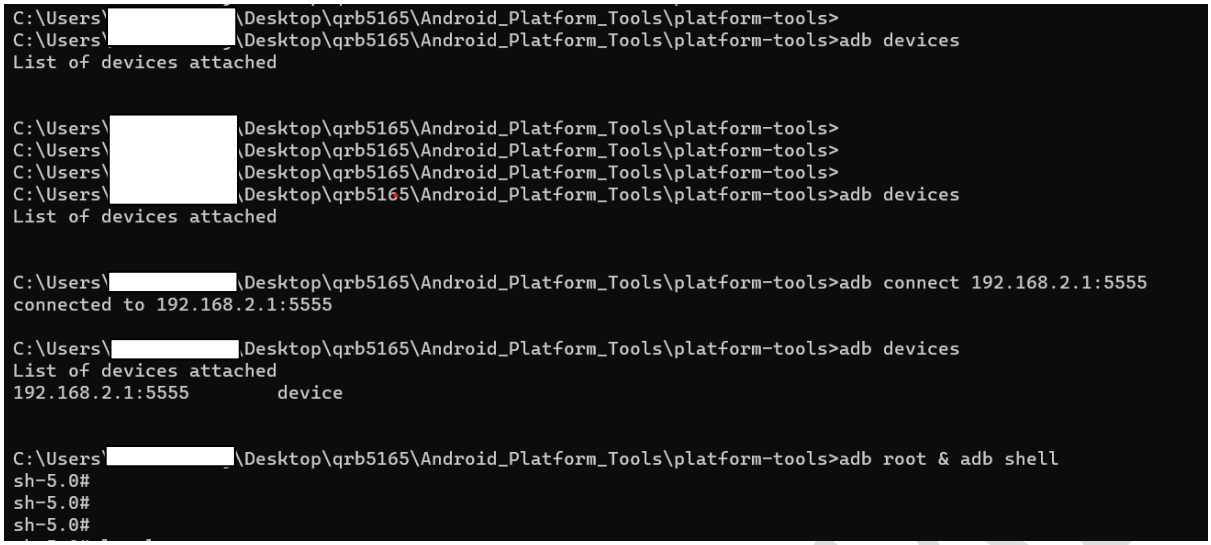

Figure 11 MRD5165 Platform ADB connection Over Wi-Fi

To disconnect the adb device from the Host PC, run the following commands and then disconnect the Wi-Fi connection on Host PC.

```
$ adb disconnect 192.168.2.1:5555
```

4. Peripherals Validation

RGB LEDs

MRD5165 Platform Setup:

NOTE: The LEDs are physically mount on the board. The enclosure may require to be removed for viewing the LED status.

- Open an adb shell over Wi-Fi (Follow the ADB over Wi-Fi steps) and execute the following commands to control the RGB LEDs
  LEDs ON:

```
# echo 255 > /sys/class/leds/red/brightness
# echo 255 > /sys/class/leds/green/brightness
# echo 255 > /sys/class/leds/blue/brightness
```

  LEDs OFF:

```
# echo 0 > /sys/class/leds/red/brightness
# echo 0 > /sys/class/leds/green/brightness
# echo 0 > /sys/class/leds/blue/brightness
```

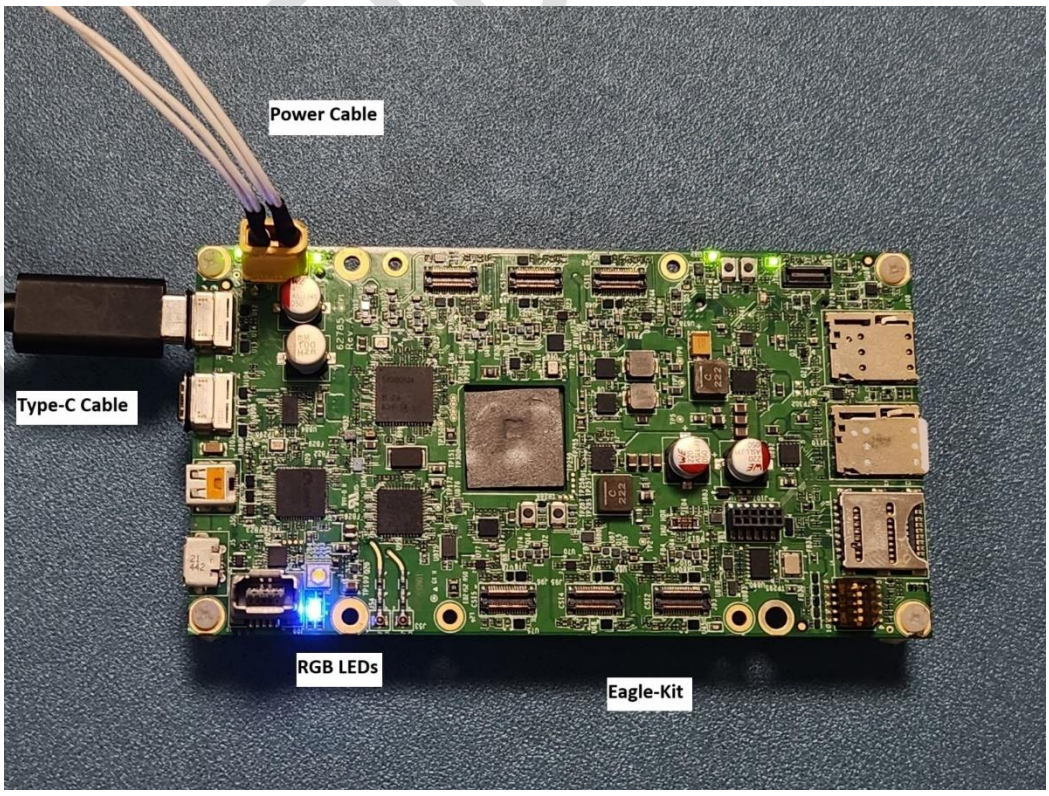

Figure 12 MRD5165 Platform RGB LEDs, Here BLUE LED state is ON

Switches

MRD5165 Platform Setup:

NOTE: The switches are physically mount on the board. The enclosure would have to be removed for accessing these switches / buttons.

- Open an adb shell over Wi-Fi (Follow the ADB over Wi-Fi steps) and execute the following commands to verify the VOL-UP, VOL-DOWN, POWER switches events.

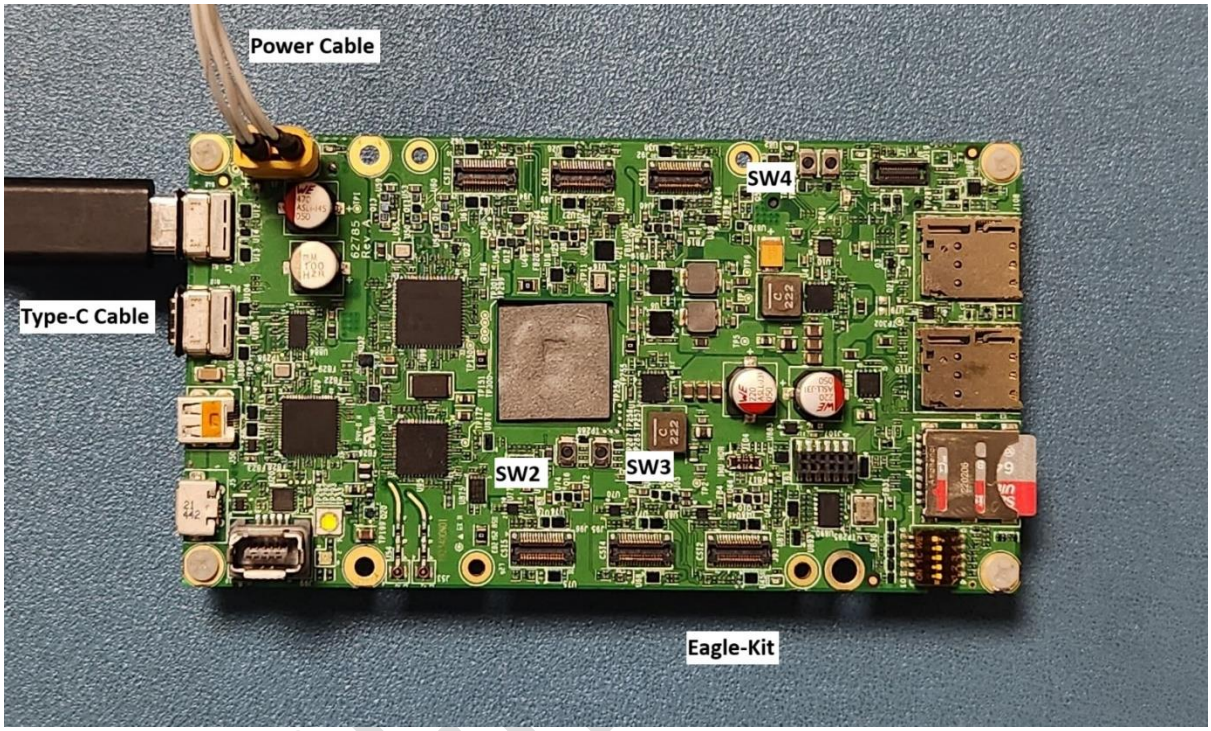

Figure 13 MRD5165 Platform Switches

SW2: VOL-UP, SW3: VOL-DOWN, SW5: POWER

VOL-UP KEY:

- Execute the following command and press the VOL-UP: SW2 button.

```
# evtest /dev/input/event1
```

Observe the event logs for KEY_VOLUMEUP button press.

```
sh-5.0# evtest /dev/input/event1
Input driver version is 1.0.1
Input device ID: bus 0x19 vendor 0x1 product 0x1 version 0x100
Input device name: "gpio-keys"
Supported events:
  Event type 0 (EV_SYN)
  Event type 1 (EV_KEY)
    Event code 115 (KEY_VOLUMEUP)
Properties:
Testing ... (interrupt to exit)
Event: time 1679939996.1679939996, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), value 1
Event: time 1679939996.1679939996, -------------- SYN_REPORT ------------
Event: time 1679939996.1679939996, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), value 0
Event: time 1679939996.1679939996, -------------- SYN_REPORT ------------
```

Figure 14 MRD5165 Platform VOLUME-UP KEY event

VOL-DOWN KEY:

- Execute the following command and press the VOL-DOWN: SW3 button.

  ```
  # evtest /dev/input/event0
  ```

  Observe the event logs for KEY_VOLUMEDOWN button press.

```
sh-5.0# evtest /dev/input/event0
Input driver version is 1.0.1
Input device ID: bus 0x0 vendor 0x0 product 0x0 version 0x0
Input device name: "qpnp_pon"
Supported events:
  Event type 0 (EV_SYN)
  Event type 1 (EV_KEY)
    Event code 114 (KEY_VOLUMEDOWN)
    Event code 116 (KEY_POWER)
Properties:
Testing ... (interrupt to exit)
Event: time 1679940045.1679940045, type 1 (EV_KEY), code 114 (KEY_VOLUMEDOWN), value 1
Event: time 1679940045.1679940045, -------------- SYN_REPORT ------------
Event: time 1679940045.1679940045, type 1 (EV_KEY), code 114 (KEY_VOLUMEDOWN), value 0
Event: time 1679940045.1679940045, -------------- SYN_REPORT ------------
```

Figure 15 MRD5165 Platform VOLUME-DOWN KEY event

POWER KEY:

- Execute the following command and press the POWER: SW4 button.

  ```
  # evtest /dev/input/event0
  ```

  Observe the event logs for KEY_POWER button press.

```
sh-5.0# evtest /dev/input/event0
Input driver version is 1.0.1
Input device ID: bus 0x0 vendor 0x0 product 0x0 version 0x0
Input device name: "qpnp_pon"
Supported events:
  Event type 0 (EV_SYN)
  Event type 1 (EV_KEY)
    Event code 114 (KEY_VOLUMEDOWN)
    Event code 116 (KEY_POWER)
Properties:
Testing ... (interrupt to exit)
Event: time 1679940092.1679940092, type 1 (EV_KEY), code 116 (KEY_POWER), value 1
Event: time 1679940092.1679940092, -------------- SYN_REPORT ------------
Event: time 1679940092.1679940092, type 1 (EV_KEY), code 116 (KEY_POWER), value 0
Event: time 1679940092.1679940092, -------------- SYN_REPORT ------------
```

Figure 16 MRD5165 Platform POWER KEY event

Wi-Fi Station Mode

MRD5165 Platform Setup:

- Open an adb shell over Type-C (Follow the ADB over Wi-Fi steps) and execute the following commands to verify the Wi-Fi Station mode connection.

  a. Add either a “open” network configuration or secured network configuration to the **/data/misc/wifi/wpa_supplicant.conf** file as shown below.

  Open networks connect configuration:

```
network= {
ssid="Open-AP-Wifi-Name"
key_mgmt=NONE
}
```

  Secured networks connect configuration:

```
network= {
ssid="Secured-AP-Wi-Fi-Name"
key_mgmt=WPA-PSK
pairwise=TKIP CCMP
group=TKIP CCMP
psk=”AP-Wi-Fi-password”
}
```

  b. Kill all the running wpa_supplicant and hostapd daemons.

```
# killall wpa_supplicant
# killall hostapd
```

  c. Run the “wpa_supplicant” with network configure wpa_supplicant.conf file

```
# wpa_supplicant -Dnl80211 -iwlan0 -c
/data/misc/wifi/wpa_supplicant.conf &
```

  d. Run “dhcpcd” to request the IP address

```
# dhcpcd
```

  e. Check the “wlan0” iface IP address

```
# ifconfig wlan0
```

```
sh-5.0# ifconfig wlan0
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.43.145  netmask 255.255.255.0  broadcast 192.168.43.255
        inet6 fe80::d243:67c6:eab6:6707  prefixlen 64  scopeid 0x20<link>
        ether 00:03:7f:12:9f:d2  txqueuelen 3000  (Ethernet)
        RX packets 80  bytes 6474 (6.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 84  bytes 6658 (6.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 17 MRD5165 Platform Wi-Fi Station mode IP address

f. Run ping command to test the network connection.

```
# ping www.google.com
```

```
sh-5.0# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: icmp_seq=0 ttl=112 time=329.715 ms
64 bytes from 8.8.8.8: icmp_seq=1 ttl=112 time=54.053 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=112 time=55.244 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=112 time=63.904 ms
^C--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 54.053/125.729/329.715/117.833 ms
```

Figure 18 MRD5165 Platform Wi-Fi Station mode Ping test

Note: **Reboot the board to enable the MRD5165 Platform in Wi-Fi Hotspot mode (Default mode)**

Wi-Fi Access Point Mode

NOTE: By default, MRD5165 Platform boot enables Wi-Fi in Access Point mode.

To override the default AP mode configuration, follow the below steps.

MRD5165 Platform Setup:

- Open an adb shell over Type-C (Follow the ADB over Wi-Fi steps) and execute the following commands to verify the Wi-Fi AP mode connection.

a. Kill all the running wpa_supplicant and hostapd daemons
```
# killall wpa_supplicant
# killall hostapd
```

b. Run the hostapd daemon with the default "/etc/hostapd/hostapd.conf" configuration file.
```
# hostapd /etc/hostapd/hostapd.conf &
```
Modify the ""/etc/hostapd/hostapd.conf" "ssid" parameter for hotspot name and "wpa_passphrase" for WPA password.

c. Setup the "wlan0" iface and dhcp server
```
# ifconfig wlan0 192.168.2.1 netmask 255.255.255.0 up
# dnsmasq  --dhcp-range=192.168.2.10,192.168.2.100,12h --port=5353
```

d. Connect an external Wi-Fi device to Hotspot hosted by MRD5165 Platform
Default WPA password is "1234567890"

Ethernet

MRD5165 Platform Setup:

- Open an adb shell over Type-C (Follow the ADB over Wi-Fi steps) and execute the following commands to verify the Ethernet connection.

1. Connect ethernet cable to the MRD5165 Platform

Figure 19 MRD5165 Platform Ethernet test

2. Verify the ethernet interface "eth0" IP address
   # ifconfig eth0

```
sh-5.0# ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.13.159  netmask 255.255.255.0  broadcast 192.168.13.255
        inet6 fe80::d34f:86ee:97d0:a560  prefixlen 64  scopeid 0x20<link>
        ether 00:1e:c0:e1:2c:8a  txqueuelen 1000  (Ethernet)
        RX packets 824  bytes 71664 (71.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 30  bytes 3012 (3.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 20 Ethernet Interface IP address

3. Delete if any “default” route with gateway IP 0.0.0.0 is present.

```
# route (Will list all the route gateway)
# route del default
```

```
root@qrb5165-ifb:~#
root@qrb5165-ifb:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         0.0.0.0         0.0.0.0         U     0      0        0 eth0
default         192.168.13.1    0.0.0.0         UG    210    0        0 eth0
link-local      0.0.0.0         255.255.0.0     U     312    0        0 wlan0
192.168.2.0     0.0.0.0         255.255.255.0   U     0      0        0 wlan0
192.168.13.0    0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.13.0    0.0.0.0         255.255.255.0   U     210    0        0 eth0
192.168.225.0   0.0.0.0         255.255.255.0   U     0      0        0 bridge0
root@qrb5165-ifb:~#
root@qrb5165-ifb:~# route del default
root@qrb5165-ifb:~#
root@qrb5165-ifb:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         192.168.13.1    0.0.0.0         UG    210    0        0 eth0
link-local      0.0.0.0         255.255.0.0     U     312    0        0 wlan0
192.168.2.0     0.0.0.0         255.255.255.0   U     0      0        0 wlan0
192.168.13.0    0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.13.0    0.0.0.0         255.255.255.0   U     210    0        0 eth0
192.168.225.0   0.0.0.0         255.255.255.0   U     0      0        0 bridge0
root@qrb5165-ifb:~#
```

Figure 21 Ethernet Ping test

Bluetooth

MRD5165 Platform Setup:

- Open an adb shell over Type-C (Follow the ADB over Wi-Fi steps) and execute the following commands to verify the Bluetooth connection.
- Keep an external Bluetooth device in discover mode to pair with the Eagle-kit and find the BT device address with "inquiry" command execution below and provide the same for "pair" command.

1. Delete the below Bluetooth configuration files

```
# rm /data/misc/bluetooth/bt*

# rm /data/misc/bluetooth/interop_database_dynamic.conf
```

2. Edit the "bt_app.conf" as below.

```
# vi /etc/Bluetooth/bt_app.conf
```

Configure the below parameters to false.

```
BtA2dpSinkEnable=false
BtHfClientEnable=false
```

3. Validating Bluetooth

Excute the "btproperty" and "btapp" and run the following commands to pair a BT device.

```
# btproperty & btapp
```

- gap_menu
- enable
- inquiry
- pair <external BT device address>
- yes
- bonded_list

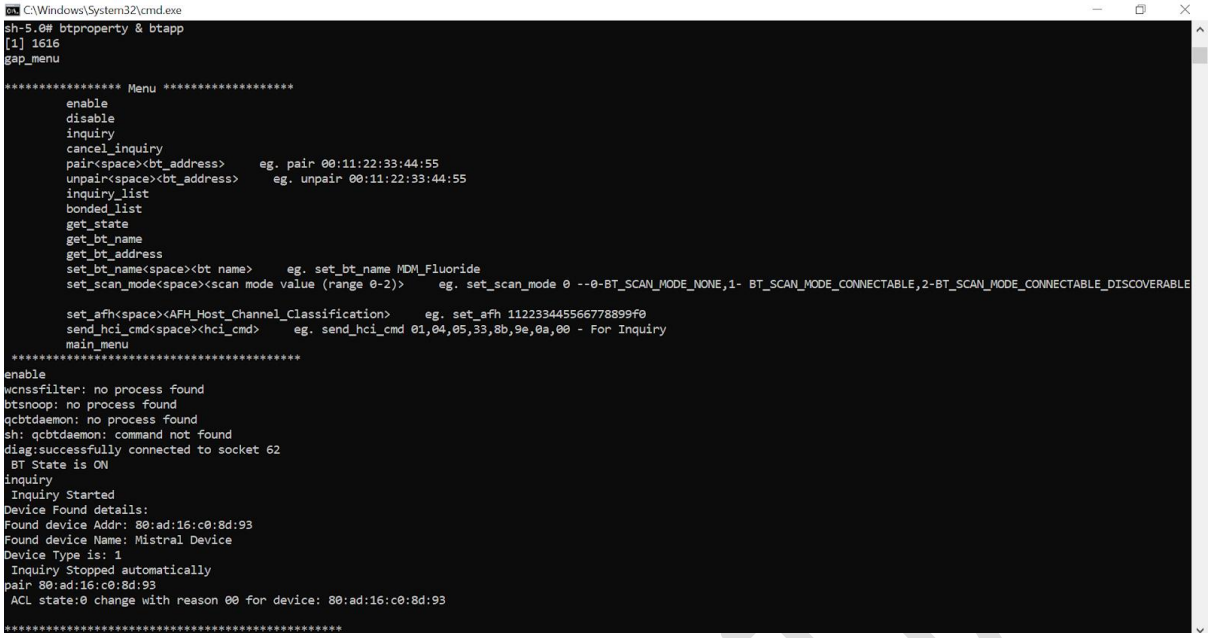

Figure 22 MRD5165 Platform Bluetooth test

Run the below commands to exit the btapp

- `main_menu`
- `exit`

SD Card

MRD5165 Platform Setup:

- Connect a Micro SD card to the MRD5165 Platform board.

Figure 23 MRD5165 Platform SD card

- Open an adb shell over Wi-Fi (Follow the ADB over Wi-Fi steps) and execute the following commands to verify the SD card.

  a. Find the SD card device node
  ```
  # ls -la /dev/mmcblk*
  ```

  b. Mount the SD card partition device node
  ```
  # mkdir -p /mnt/sdcard
  ```

  Example command:
  ```
  # mount /dev/mmcblk2p1 /mnt/sdcard
  ```

  c. Perform file write operation
  ```
  # dd if=/dev/urandom of=/mnt/sdcard/test.txt bs=30M count=2
  conv=fsync
  ```

  Verify the file presence after the above command execution.

d. Unmount the SD card

Example command:

```
# umount /dev/mmcblk2p1
```

```
sh-5.0# ls -la /dev/mmcblk*
brw-rw---- 1 root disk   179, 0 Nov  7 06:11 /dev/mmcblk2
brw-rw---- 1 root system 179, 1 Nov  7 06:11 /dev/mmcblk2p1
brw-rw---- 1 root system 179, 2 Nov  7 06:11 /dev/mmcblk2p2
brw-rw---- 1 root system 179, 3 Nov  7 06:11 /dev/mmcblk2p3
sh-5.0# fdisk -l /dev/mmcblk2
Disk /dev/mmcblk2: 29.74 GiB, 31914983424 bytes, 62333952 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xd35b7e18

Device         Boot   Start      End  Sectors  Size Id Type
/dev/mmcblk2p1 *       2048   411647   409600  200M  b W95 FAT32
/dev/mmcblk2p2       411648  8800255  8388608    4G 83 Linux
/dev/mmcblk2p3      8800256 62332927 53532672 25.5G 83 Linux
sh-5.0# mkdir /mnt/sdcard
sh-5.0# mount /dev/mmcblk2p1 /mnt/sdcard
sh-5.0# dd if=/dev/urandom of=/mnt/sdcard/test.txt bs=30M count=2 conv=fsync
2+0 records in
2+0 records out
62914560 bytes (63 MB, 60 MiB) copied, 4.90529 s, 12.8 MB/s
sh-5.0# umount /dev/mmcblk2p1
sh-5.0#
```

Figure 24 MRD5165 Platform SD card test

Sensors

MRD5165 Platform Setup:

- Open an adb shell over Wi-Fi (Follow the ADB over Wi-Fi steps) and execute the following commands to verify all the sensors.

Accelerometer Sensor:

Run Driver Acceptance Test:

```
# ssc_drva_test -sensor=accel -duration=5 -sample_rate=50
```

Observe the "total sample" count value and "received event" result PASS.

Here duration is set 5 seconds and sample_rate set to 50, expected total samples count would be approximately 5*50=250.

```
sh-5.0# ssc_drva_test -sensor=accel -duration=5 -sample_rate=50
1 ssc drva test version 1.13
1 ssc_drva_test -sensor=accel -duration=5 -sample_rate=50
1 handle_event
1 event_cb attribute event for da_test
1 handle_event
1 event_cb attribute event for da_test
1 using da_test name=da_test, suid = [high addeaddeaddeadde, low addeaddeaddeadde
1 enter send_memory_log_req cookie: 1
1 exit send_memory_log_req
1 enter da_test runner
1 handle_event
1 -time_to_first_event=2258062
1 -time_to_last_event=-156733
1 -sample_ts=11537668924
1 -total_samples=247
1 -avg_delta=378991
1 -recvd_phy_config_sample_rate=50
1 -random_seed_used=2851690000
1 -num_request_sent=2
1 -first_sample_timestamp=11443923159
1 handle_event
1 received event: PASS
1 enter send_memory_log_req cookie: 1
1 exit send_memory_log_req
1 PASS
sh-5.0#
```

Figure 25 MRD5165 Platform Accelerometer Sensor test

Gyro Sensor:

Driver Acceptance Test:

```
# ssc_drva_test -sensor=gyro -duration=5 -sample_rate=50
```

Observe the "total sample" count value and "received event" result PASS.

Here duration is set 5 seconds and sample_rate set to 50, expected total samples count would be approximately 5*50=250.

```
sh-5.0# ssc_drva_test -sensor=gyro -duration=5 -sample_rate=50
2 ssc_drva_test version 1.13
2 ssc_drva_test -sensor=gyro -duration=5 -sample_rate=50
2 handle_event
2 event_cb attribute event for da_test
2 handle_event
2 event_cb attribute event for da_test
2 using da_test name=da_test, suid = [high addeaddeaddeadde, low addeaddeaddeadde
2 enter send_memory_log_req cookie: 2
2 exit send_memory_log_req
2 enter da_test runner
2 handle_event
2 -time_to_first_event=2776890
2 -time_to_last_event=-277916
2 -sample_ts=12247071021
2 -total_samples=245
2 -avg_delta=379476
2 -recvd_phy_config_sample_rate=50
2 -random_seed_used=3561088963
2 -num_request_sent=2
2 -first_sample_timestamp=12153843651
2 handle_event
2 received event: PASS
2 enter send_memory_log_req cookie: 2
2 exit send_memory_log_req
2 PASS
sh-5.0#
```

Figure 26 MRD5165 Platform Gyro Sensor test

Magnetometer Sensor:

Run Driver Acceptance Test:

```
# ssc_drva_test -sensor=mag -duration=5 -sample_rate=10
```

Observe the "total sample" count value and "received event" result PASS.

Here duration is set 5 seconds and sample_rate set to 10, expected total samples count would be approximately 5*10=50.

```
sh-5.0# ssc_drva_test -sensor=mag -duration=5 -sample_rate=10
3 ssc_drva_test version 1.13
3 ssc_drva_test -sensor=mag -duration=5 -sample_rate=10
3 handle_event
3 event_cb attribute event for da_test
3 handle_event
3 event_cb attribute event for da_test
3 using da_test name=da_test, suid = [high addeaddeaddeadde, low addeaddeaddeadde
3 enter send_memory_log_req cookie: 3
3 exit send_memory_log_req
3 enter da_test runner
3 handle_event
3 -time_to_first_event=2146058
3 -time_to_last_event=-1760757
3 -sample_ts=12892869548
3 -total_samples=49
3 -avg_delta=1880816
3 -recvd_phy_config_sample_rate=10
3 -random_seed_used=4206894614
3 -num_request_sent=2
3 -first_sample_timestamp=12799012430
3 handle_event
3 received event: PASS
3 enter send_memory_log_req cookie: 3
3 exit send_memory_log_req
3 PASS
```

Figure 27 MRD5165 Platform Magnetometer Sensor test

Pressure Sensor:

Run Driver Acceptance Test:

```
# ssc_drva_test -sensor=pressure -duration=5 -sample_rate=25
```

Observe the "total sample" count value and "received event" result PASS.

Here duration is set 5 seconds and sample_rate set to 25, expected total samples count would be approximately 5*25=125.

```
sh-5.0# ssc_drva_test -sensor=pressure -duration=5 -sample_rate=25
4 ssc_drva_test version 1.13
4 ssc_drva_test -sensor=pressure -duration=5 -sample_rate=25
4 handle_event
4 event_cb attribute event for da_test
4 handle_event
4 event_cb attribute event for da_test
4 using da_test name=da_test, suid = [high addeaddeaddeadde, low addeaddeaddeadde
4 enter send_memory_log_req cookie: 4
4 exit send_memory_log_req
4 enter da_test runner
4 handle_event
4 -time_to_first_event=2489724
4 -time_to_last_event=-595196
4 -sample_ts=13490701676
4 -total_samples=122
4 -avg_delta=761705
4 -recvd_phy_config_sample_rate=25
4 -random_seed_used=509759345
4 -num_request_sent=2
4 -first_sample_timestamp=13397188127
4 handle_event
4 received event: PASS
4 enter send_memory_log_req cookie: 4
4 exit send_memory_log_req
4 PASS
sh-5.0#
```

Figure 28 MRD5165 Platform Pressure Sensor test

SOM Thermistors

MRD5165 Platform Setup:

- Open an adb shell over Wi-Fi (Follow the ADB over Wi-Fi steps) and execute the following commands to verify all the sensors.

pm8250-wifi-usr:

```
# cat /sys/class/thermal/thermal_zone63/temp
```

pm8150l-therm-usr:

```
# cat /sys/class/thermal/thermal_zone64/temp
```

pm8250-xo-therm-usr:

```
# cat /sys/class/thermal/thermal_zone65/temp
```

pm8150l-skin-step:

```
# cat /sys/class/thermal/thermal_zone66/temp
```

Example:

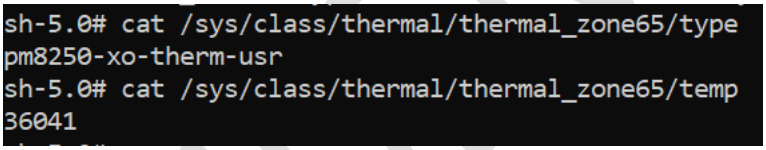

Figure 29 MRD5165 Platform Thermistors test

SOM Current and Voltage Monitor Sensor

MRD5165 Platform Setup:

- Open an adb shell over Wi-Fi (Follow the ADB over Wi-Fi steps) and execute the following commands to verify all the sensors.

INA sensor is a current shunt and power monitor sensor. This sensor measures the MRD5165 SOM current consumption and voltage level.

INA Sensor device sysfs entry check:

```
# cat /sys/class/hwmon/hwmon5/name
```

Current consumption value in mA:

```
# cat /sys/class/hwmon/hwmon5/curr1_input
```

Voltage Drop value in mV:

```
# cat /sys/class/hwmon/hwmon5/in0_input
```

Voltage in value in mV:

```
# cat /sys/class/hwmon/hwmon5/in1_input
```

```
sh-5.0# cat /sys/class/hwmon/hwmon5/name
ina231
sh-5.0# cat /sys/class/hwmon/hwmon5/curr1_input
303
sh-5.0# cat /sys/class/hwmon/hwmon5/in0_input
3
sh-5.0# cat /sys/class/hwmon/hwmon5/in1_input
3619
```

Figure 30 MRD5165 Platform INA Sensor test

FAN Connector

MRD5165 Platform Setup:

Open an adb shell over Wi-Fi (Follow the ADB over Wi-Fi steps) and execute the following commands to verify fan connector.

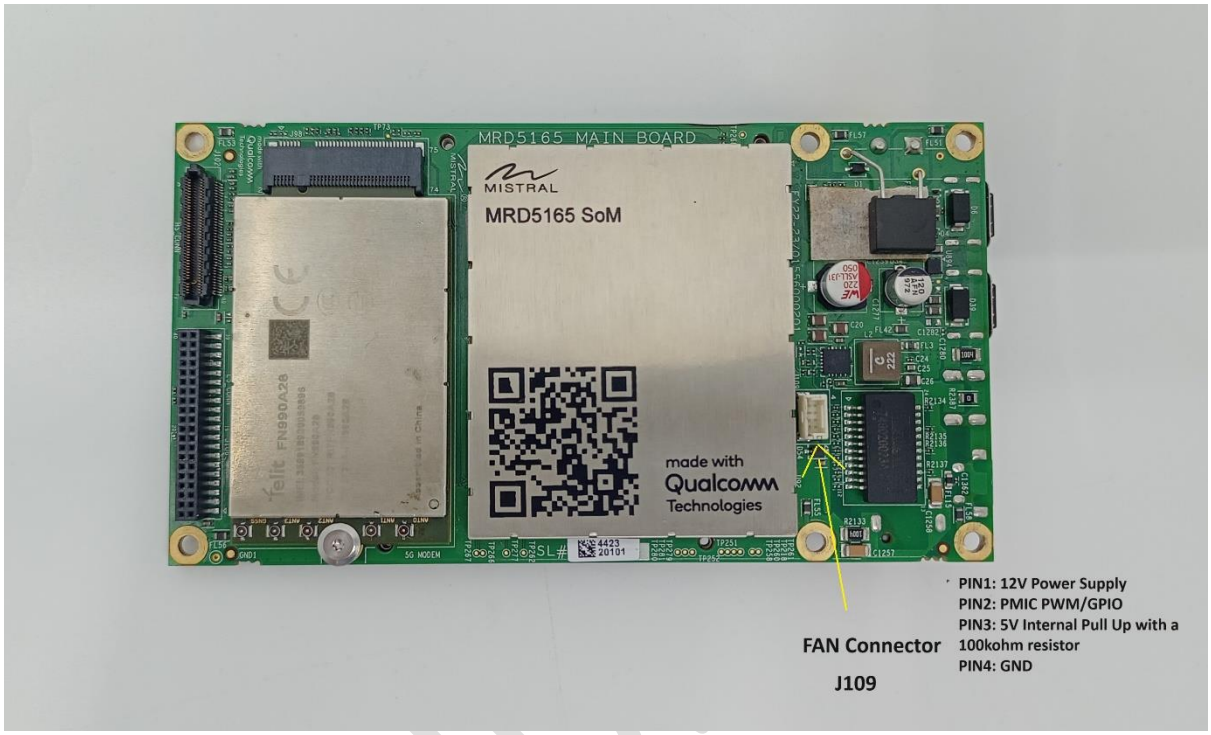

Figure 31 MRD5165 Fan Connector

Note: Fan speed control feature (using PWM) is not enabled for the MRD5165 Platform.

The external FAN state (ON/OFF) can be controlled by exporting the PMIC GPIO terminating at the fan connector PIN2 on the MRD5165 Platform.

```
sh-5.0# cd /sys/class/gpio/
sh-5.0# echo 303 > export   /* Export PMIC GPIO */
sh-5.0# cd gpio303/
sh-5.0# echo out > direction /* Set “out” direction */
```

Turning FAN ON State:

```
sh-5.0# echo 0 > value
```

Turning FAN OFF State:

```
sh-5.0# echo 1 > value
```

5G Module

MRD5165 Platform supports connecting the PCIe/USB based cellular connectivity modems. The default platform (Eagle-Kit as well as Edge AI-Box) does not include the module. The module may need to be installed manually. Please refer to the integration guide for instructions on installing the 5G modem.

NOTE: - Depending on the ordering options, the module may be pre-installed. Verify whether the 5G module is configured in PCIe mode or USB mode and follow the appropriate instructions below for network setup.

MRD5165 Platform Setup:ss

- Connect a 5G module with antenna to the MRD5165 Platform and power up the setup.
- Connect a SIM card to SIM1 slot.
- Open an adb shell over Wi-Fi (Follow the ADB over Wi-Fi steps) and execute the following commands to verify the 5G module connection.

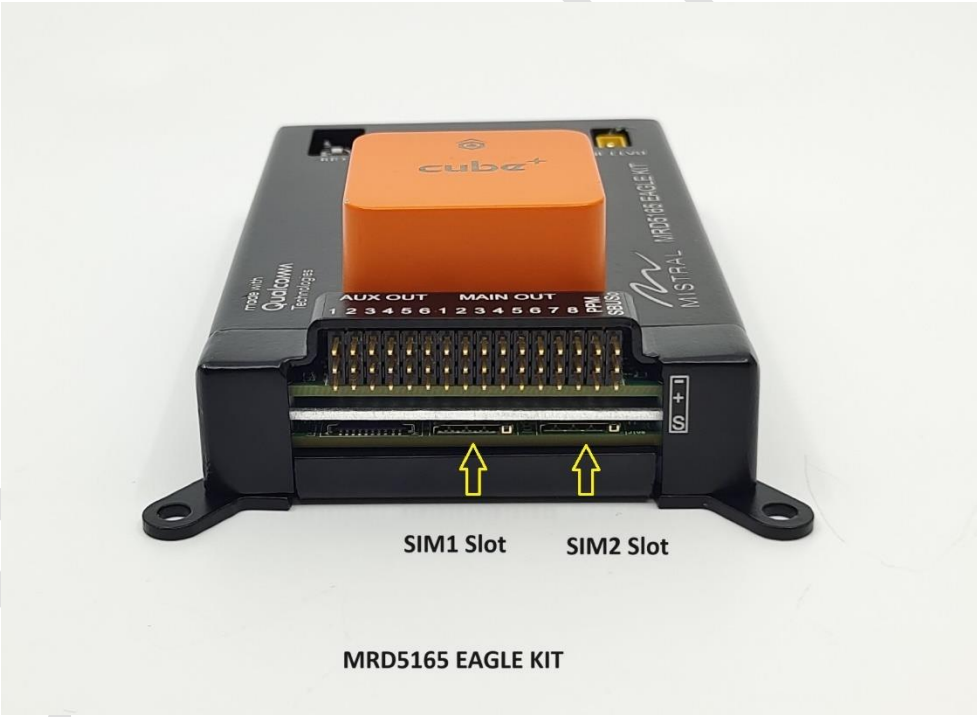

Figure 32 MRD5165 Platform SIM card slots

Figure 33 MRD5165 Platform 5G Module

PCIe mode:

1. Verify the 5G Module presence over PCIe list
   # lspci
   Expected logs for the Telit FN990 5G module is connected:
   ```
   0001:00:00.0 PCI bridge: Qualcomm Device 010b
   0001:01:00.0 Unassigned class [ff00]: Qualcomm Device 0308
   ```

   Expected logs for the Telit FN980 or Quectel RM502Q 5G module is connected:
   ```
   0001:00:00.0 PCI bridge: Qualcomm Device 010b
   0001:01:00.0 Unassigned class [ff00]: Qualcomm Device 0306
   ```

2. Run the following commands to start the network

   If the Telit FN990 5G module is connected run the below command:
   ```
   # qmi_dev=/dev/mhi_0308_01.01.00_pipe_14
   ```

   Or

   If the Telit FN980 5G or Quectel RM502 module is connected, use this command:
   ```
   # qmi_dev=/dev/mhi_0306_01.01.00_pipe_14


   # qmicli -d $qmi_dev --device-open-proxy --wds-set-ip-family=4 --
   client-no-release-cid --device-open-qmi

   # qmicli -d $qmi_dev --device-open-proxy --uim-get-card-status --
   device-open-qmi

   # qmicli -d $qmi_dev --device-open-proxy --wds-start-network="ip-
   type=4,apn=fast.t-mobile.com" --client-no-release-cid --client-cid=15
   --device-open-qmi
   <Configure the apn based on the SIM service provider>

   # qmicli -d $qmi_dev --wds-get-current-settings --device-open-proxy -
   -client-no-release-cid --client-cid=15 --device-open-qmi

   # udhcpc -q -f -n -i rmnet_mhi0

   # ifconfig rmnet_mhi0

   # ping www.google.com
   ```

USB mode:

1. Verify the 5G Module presence over USB devices list
   # lsusb
   Bus 002 Device 004: ID 1bc7:1075 Telit Wireless Solutions

2. Run the following commands to start the network

```
# ifconfig wwan0 down

# echo 'Y' | tee /sys/class/net/wwan0/qmi/raw_ip

# ifconfig wwan0 up

# qmicli -d /dev/cdc-wdm0 --set-expected-data-format=raw-ip

# ifconfig wwan0 up

# sleep 5

# qmicli --device=/dev/cdc-wdm0 --device-open-proxy --wds-start-
network="ip-type=4,apn=fast.t-mobile.com" --client-no-release-cid

# qmicli --device=/dev/cdc-wdm0 --device-open-proxy --uim-get-card-
status --device-open-qmi

# udhcpc -q -f -n -i wwan0

# ping www.google.com
```

HDMI-in Camera Preview on Type-C Display

MRD5165 Platform Setup:

- Connect the Type-C DP port to a display monitor to USB - 0.
- Connect a HDMI camera to Micro HDMI-in port.

Figure 34 MRD5165 Platform HDMI-in Camera Setup

Open an adb shell over Wi-Fi (Follow the ADB over Wi-Fi steps) and execute the following command to preview the HDMI-In camera on Type-C Display.

```
# export XDG_RUNTIME_DIR=/run/user/root && gst-launch-1.0 -e qtiqmmfsrc
camera=0 name=camsrc ! video/x-
raw,format=NV12,width=3840,height=2160,framerate=60/1 ! waylandsink
fullscreen=true async=true sync=false
```

```
sh-5.0#
sh-5.0#
sh-5.0#
sh-5.0# export XDG_RUNTIME_DIR=/run/user/root && gst-launch-1.0 -e qtiqmmfsrc camera=0 name=camsrc ! video/x-raw,format=NV12,width=3840,height=2160,framerat
e=60/1 ! waylandsink fullscreen=true async=true sync=false
gbm_create_device(192): Info: backend name is: msm_drm
Setting pipeline to PAUSED ...
gbm_create_device(192): Info: backend name is: msm_drm
Pipeline is live and does not need PREROLL ...
Setting pipeline to PLAYING ...
New clock: GstSystemClock
```

Figure 35 MRD5165 Platform HDMI-in Camera Preview Command Execution

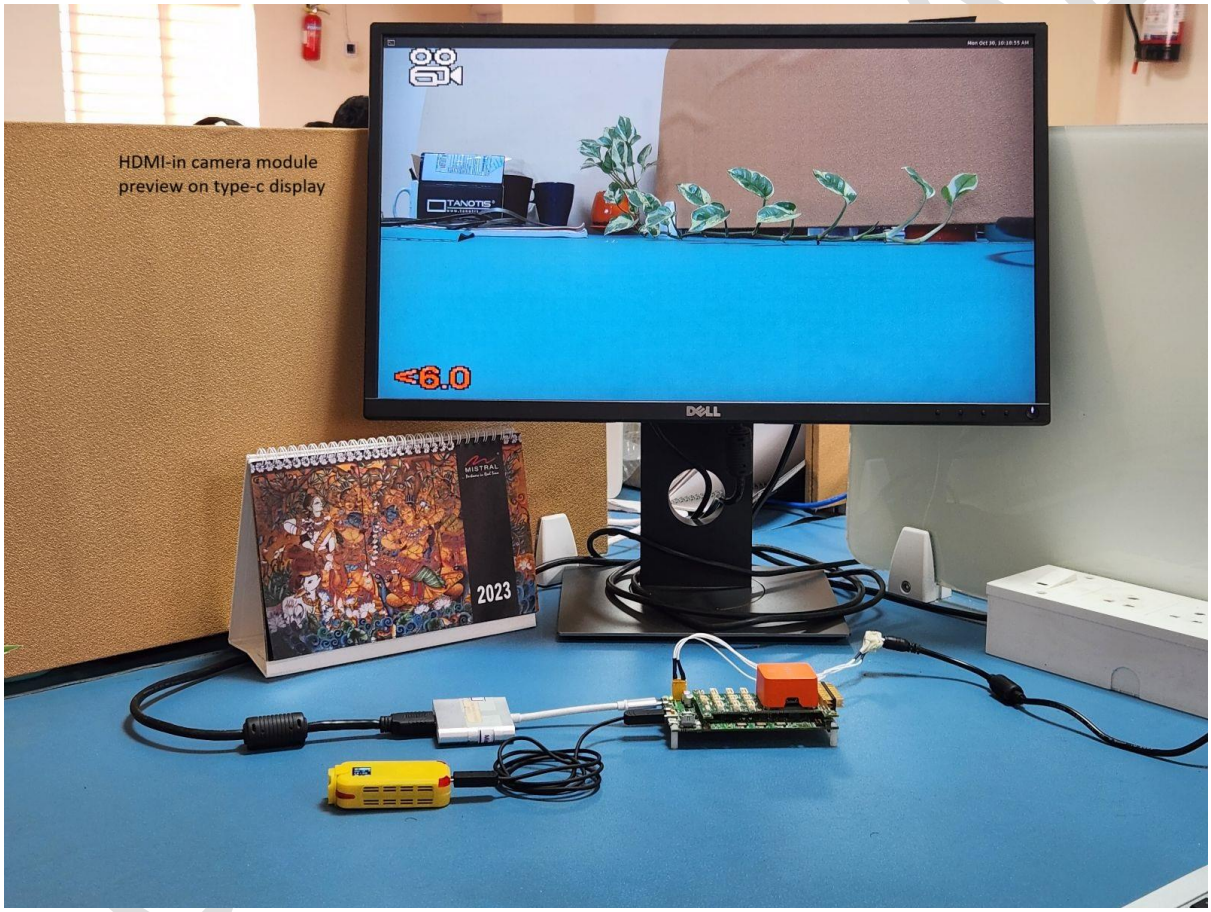

Figure 36 MRD5165 Platform HDMI-in Camera Preview on Type-C Display Monitor

HDMI-in Camera Stream Over Wi-Fi

MRD5165 Platform Setup:

- Connect the HDMI-in port to an HDMI-in camera as shown in the Fig 10 image.

Open an adb shell over Wi-Fi (Follow the ADB over Wi-Fi steps) and execute the following command to stream HDMI-In camera video over Wi-Fi.

```
# gst-launch-1.0 -e qtiqmmfsrc camera=0 name=camsrc ! video/x-raw,format=NV12,width=3840,height=2160,framerate=60/1 ! qtic2venc ! queue ! h264parse ! rtph264pay config-interval=1 pt=96 ! udpsink host = <Host PC IP Address> port=5502
```

Example Command:

```
# gst-launch-1.0 -e qtiqmmfsrc camera=0 name=camsrc ! video/x-raw,format=NV12,width=3840,height=2160,framerate=60/1 ! qtic2venc ! queue ! h264parse ! rtph264pay config-interval=1 pt=96 ! udpsink host = 192.168.2.52 port=5502
```

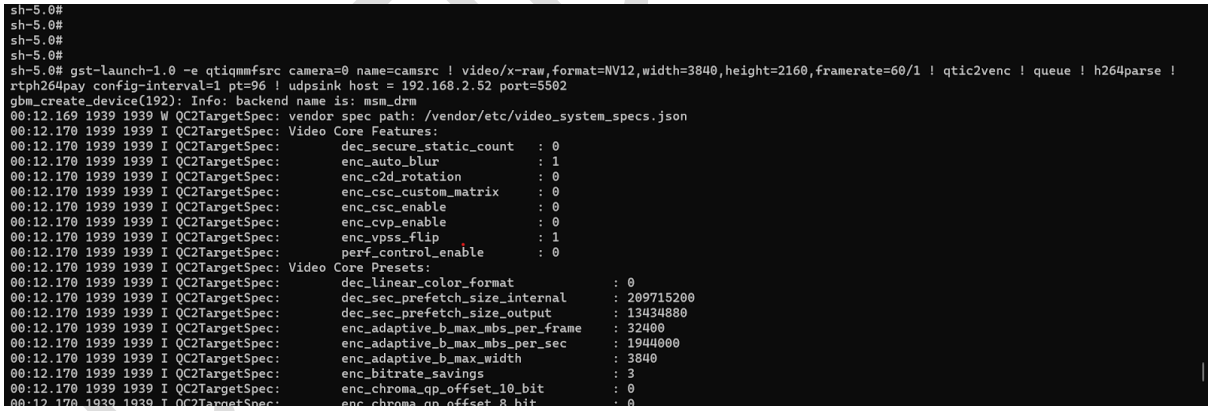

Figure 37 MRD5165 Platform HDMI-in Camera Wi-Fi Stream Command Execution

```
gbm_create_device(192): Info: backend name is: msm_drm
00:12.295 1939 1939 W StandardCaps: Preconditions for b-frame didn't meet. Disabling b-frame!
00:12.295 1939 1961 I QC2Registry: Build pipelined codec for session
00:12.295 1939 1961 I QC2Registry: Getting stages for pipelined codec
00:12.296 1939 1961 I QC2Registry: Pipelining not enabled
00:12.296 1939 1961 I QC2Registry: Doesn't support pipelining. Create standalone codec
00:12.297 1939 1961 I QC2V4l2Driver: [avcE_0] Device /dev/video33 opened with fd: 16
00:12.298 1939 1961 E QC2V4l2Driver: [avcE_0] failed to set buffer size limit to 4
00:12.299 1939 1961 E QC2V4l2Caps: c2 format not found for v4l2 format 0x34363248
00:12.299 1939 1961 E QC2V4l2Caps: c2 format not found for v4l2 format 0x43564548
00:12.300 1939 1961 W QC2V4l2Codec: [avcE_0] unknown/unsupported param coded.p-frame-count index = 0x5200C001
00:12.300 1939 1961 W QC2V4l2Codec: [avcE_0] unknown/unsupported param qti-ext-enc-b-frame-preconditions index = 0x9200C043
00:12.300 1939 1961 W QC2V4l2Codec: [avcE_0] unknown/unsupported param qti-ext-enc-adaptive-b-preconditions index = 0x9200C044
00:12.300 1939 1961 W QC2V4l2Codec: [avcE_0] unknown/unsupported param qti-ext-enc-hier-b-preconditions index = 0x9200C045
Pipeline is live and does not need PREROLL ...
Setting pipeline to PLAYING ...
New clock: GstSystemClock
00:12.406 1939 1961 E QC2Interface: Failed to query parameters
00:12.429 1939 1961 E QC2Interface: Failed to query parameters
00:12.801 1939 1961 I QC2Comp: [avcE_0] Stats: Pending(0) i/p-done(0) Works: Q: 11/Done 11|Work-Rate: Q(21.7/s Avg=21.7/s) Done(21.687/s Avg=21.687/s)| Stre
am: 30.00fps 43.5Mbps
Mem-usage:  [In-2D - 8 bufs 120.000 MB] [1D-0 - 15 bufs 179.297 MB] [1D-0 - 1 bufs 0.004 MB]
Total Mem-usage: 299.301 MB
00:17.801 1939 1961 I QC2Comp: [avcE_0] Stats: Pending(0) i/p-done(0) Works: Q: 161/Done 161|Work-Rate: Q(30.0/s Avg=29.2/s) Done(29.999/s Avg=29.233/s)| St
ream: 30.00fps 43.3Mbps
Mem-usage:  [In-2D - 9 bufs 135.000 MB] [1D-0 - 65 bufs 776.953 MB] [1D-0 - 1 bufs 0.004 MB]
Total Mem-usage: 911.957 MB
00:22.802 1939 1961 I QC2Comp: [avcE_0] Stats: Pending(0) i/p-done(0) Works: Q: 311/Done 311|Work-Rate: Q(30.0/s Avg=29.6/s) Done(29.996/s Avg=29.596/s)| St
ream: 30.00fps 43.3Mbps
Mem-usage:  [In-2D - 9 bufs 135.000 MB] [1D-0 - 65 bufs 776.953 MB] [1D-0 - 1 bufs 0.004 MB]
Total Mem-usage: 911.957 MB
```

Figure 38 MRD5165 Platform HDMI-in Camera Wi-Fi Stream Command Execution Logs

Host PC Wi-Fi Video Playback:

Open VLC player with the following Wifi-Stream.sdp file.

Wifi-Stream.sdp file content is as follows.

```
v=0
m=video 5502 RTP/AVP 96
c=IN IP4 127.0.0.1
a=rtpmap:96 H264/90000
```

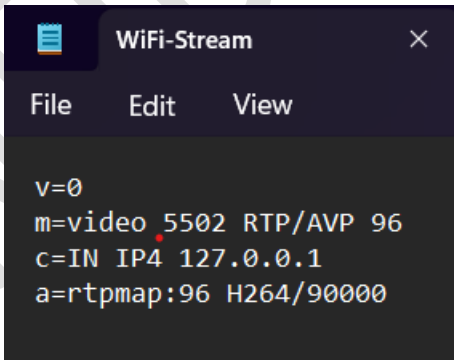

Figure 39 WiFi-Stream.sdp file content

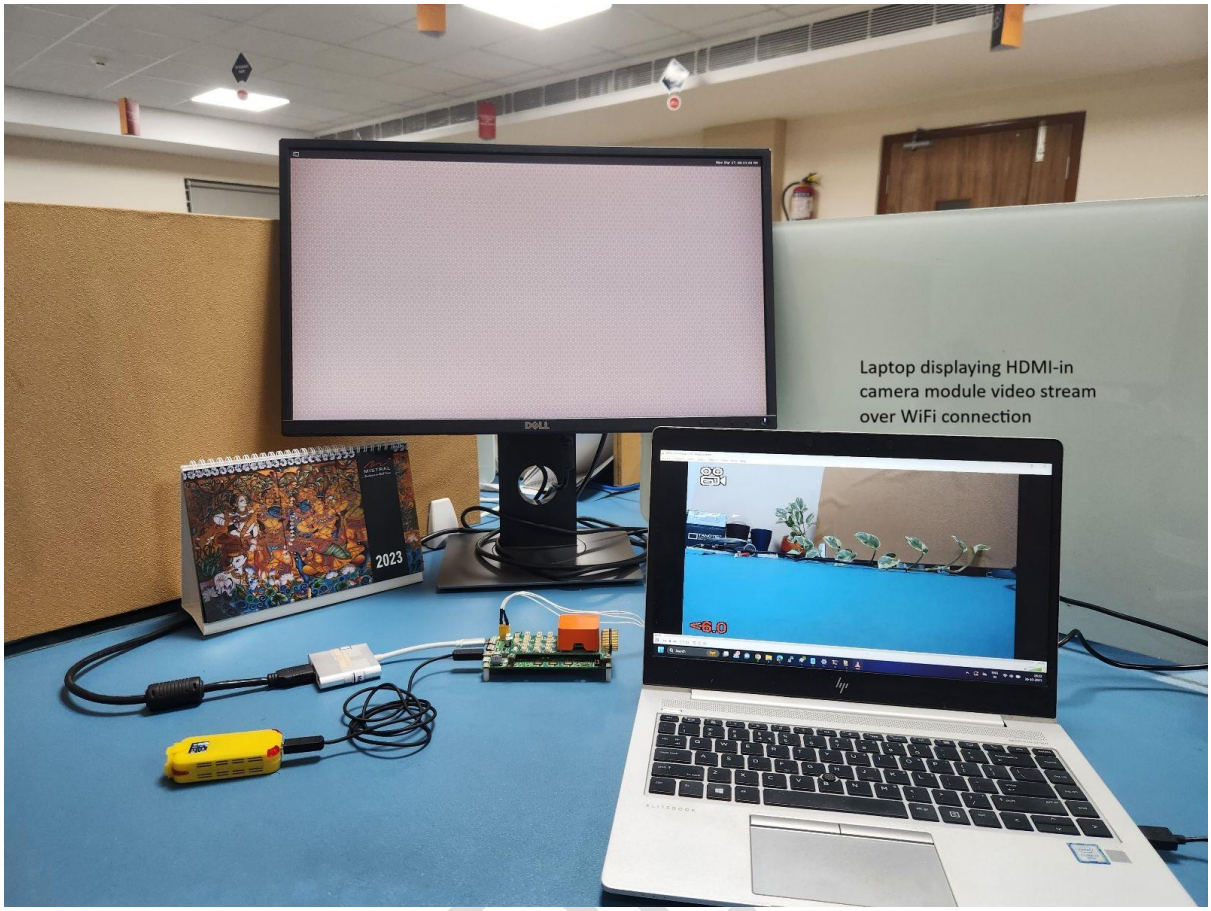

Figure 40 HDMI-in Camera Video Stream over Wi-Fi on Host PC

MRD5165 Platform Camera Ports`

MRD5165 Platform Camera Ports:

Figure 41 MRD5165 Platform Camera Ports Side-1

Figure 42MRD5165 Platform Camera Ports Side-2

Note:

- HDMI-in and CSI-0 MIPI port are using the same CSI-0 signal. HDMI-in and CSI-0 MIPI connector are mutually exclusive. By default, HDMI-in port is enabled. To enable the CSI-0 MIPI connector, drive the camera control GPIO number 88 to low.

- CSI-2 and CSI-4 are using the common CCI-2 bus and CSI-3 and CSI-5 are using the common CCI-3 bus.

- The present software doesn't support OV9282 on CSI-2 MIPI connector.

- Presently, the camera module driver on the software side supports CSI0 to CSI5. Both IMX577 and OV9282 camera modules can be connected, with the exception that CSI2 does not support the OV9282 camera module.

MRD5165 Platform CSI adapter board configuration

The CSI adapter board enables the connection of various camera modules using an IPEX cable.

Regarding the IMX577 and OV9282 camera modules, the MRD5165 Platform CSI adapter board facilitates the connection of camera modules in the following manner only:

The centre I-PEX connector is configured according to the IMX577 camera module power specifications, while the other two I-PEX connectors are configured for the OV9282 camera module power specifications.

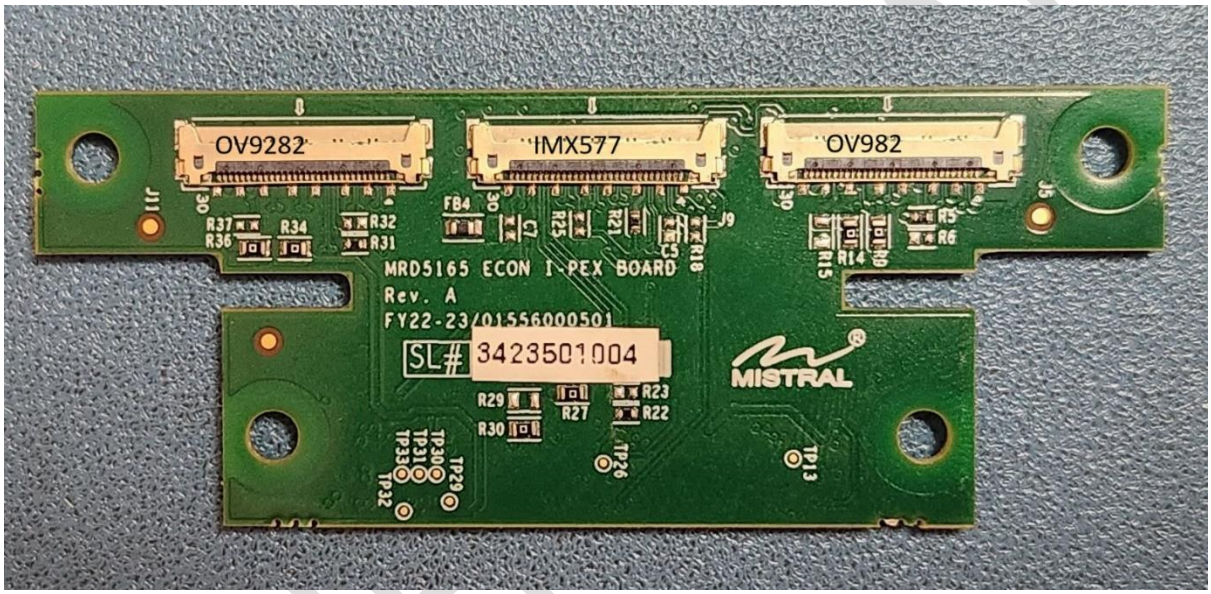

Figure 43 MRD5165 Platform CSI Camera Adapter

MRD5165 Platform Camera Setup 1

1. CSI-0 port connected to IMX577 camera module.
2. CSI-1 port connected to OV9282 camera module.

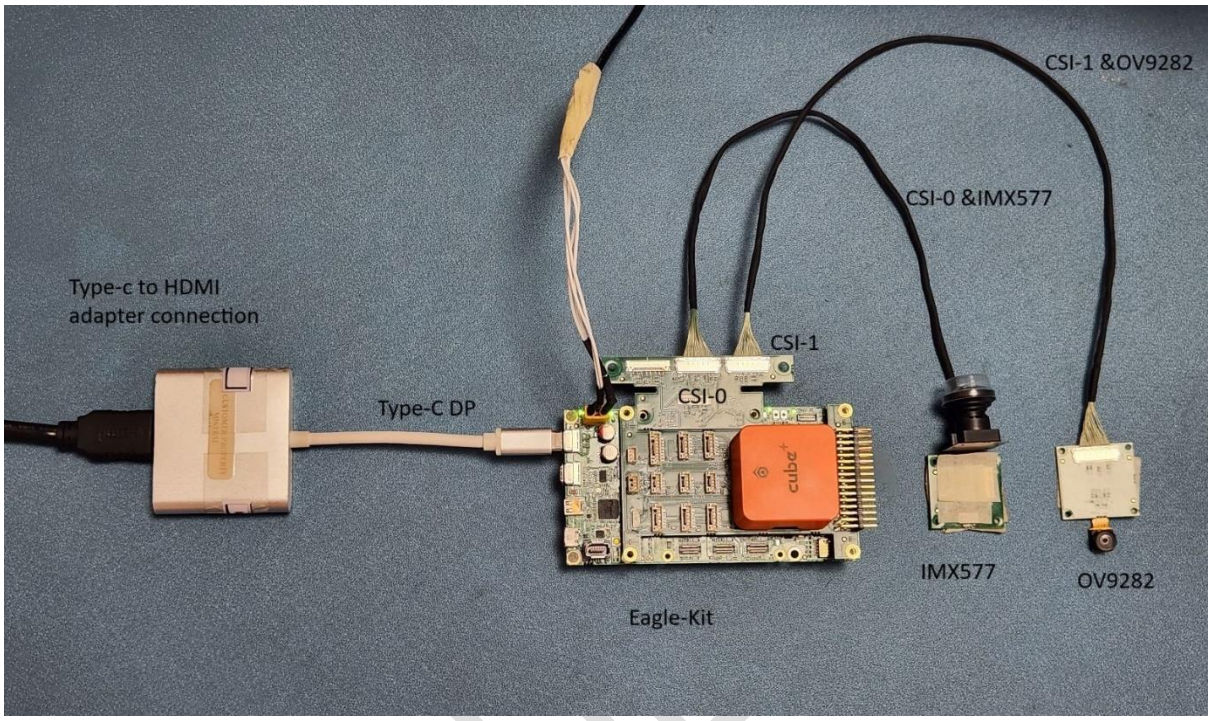

Figure 44 MRD5165 Platform Camera Setup 1

CSI-0 IMX577 Camera Preview:

Open an adb shell over Wi-Fi (Follow the ADB over Wi-Fi steps) and execute the following command to preview CSI-0 IMX577 camera on Type-c Display monitor.

```
# echo 416 > /sys/class/gpio/export; echo out >
/sys/class/gpio/gpio416/direction; echo 0 >
/sys/class/gpio/gpio416/value; echo 416 >
/sys/class/gpio/unexport
```

(To enable the CSI-0 MIPI connector CSI signals)

```
#export XDG_RUNTIME_DIR=/run/user/root && gst-launch-1.0 -e
qtiqmmfsrc camera=0 name=camsrc ! video/x-
raw,format=NV12,width=3840,height=2160,framerate=30/1 !
waylandsink fullscreen=true async=true sync=false
```

```
sh-5.0#
sh-5.0#
sh-5.0# echo 416 > /sys/class/gpio/export; echo out > /sys/class/gpio/gpio416/direction; echo 0 > /sys/class/gpio/gpio416/value; echo 416 > /sys/class/gpio/
unexport
sh-5.0#
sh-5.0# export XDG_RUNTIME_DIR=/run/user/root && gst-launch-1.0 -e qtiqmmfsrc camera=0 name=camsrc ! video/x-raw,format=NV12,width=3840,height=2160,framerat
e=30/1 ! waylandsink fullscreen=true async=true sync=false
gbm_create_device(192): Info: backend name is: msm_drm
Setting pipeline to PAUSED ...
gbm_create_device(192): Info: backend name is: msm_drm
Pipeline is live and does not need PREROLL ...
Setting pipeline to PLAYING ...
New clock: GstSystemClock
```

Figure 45 CSI-0 IMX577 Camera Setup 1 Preview Command Execution

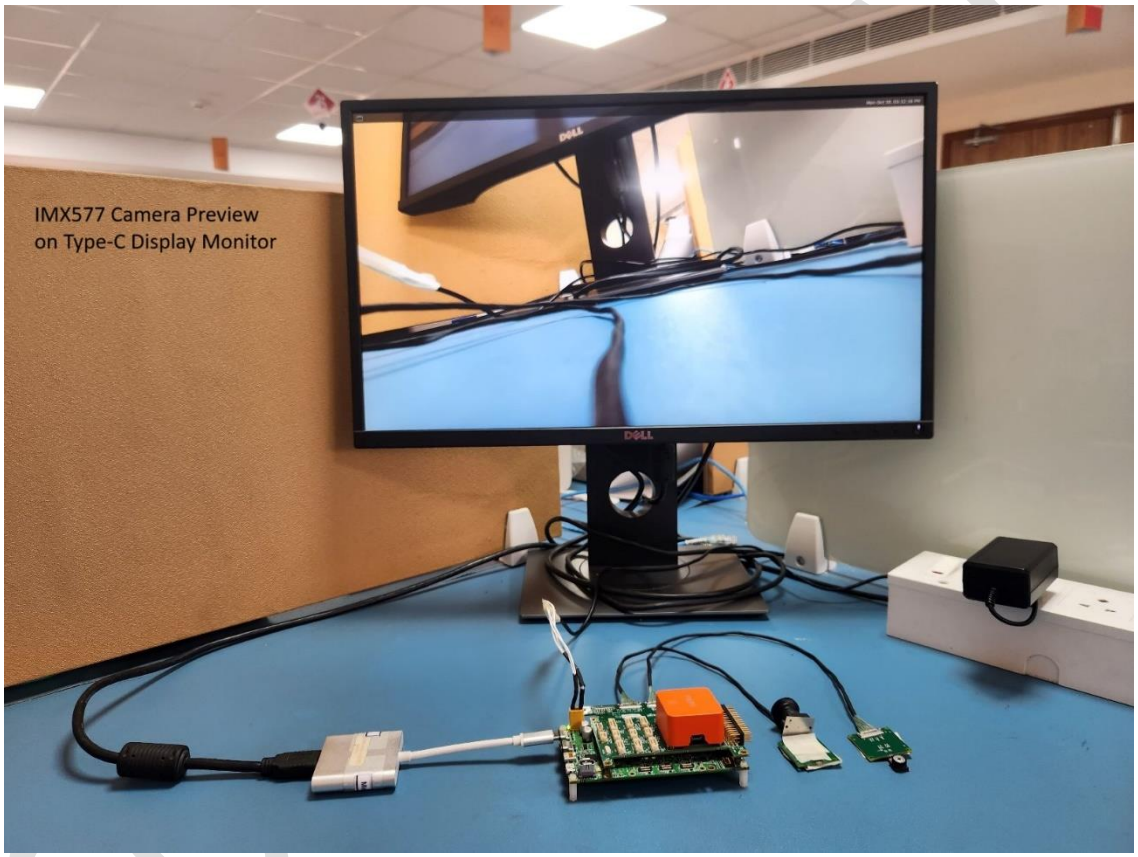

Figure 46 IMX577 Camera Setup 1 Preview on Type-C Display Monitor

CSI-1 OV9282 Camera Preview:

Open an adb shell over Wi-Fi (Follow the ADB over Wi-Fi steps) and execute the following command to preview CSI-1 OV9282 camera on Type-c Display monitor.

```
# export XDG_RUNTIME_DIR=/run/user/root && gst-launch-1.0 -e
qtiqmmfsrc camera=1 name=camsrc ! video/x-
raw,format=NV12,width=1280,height=720,framerate=90/1 !
waylandsink fullscreen=true async=true sync=false
```

```
sh-5.0#
sh-5.0#
sh-5.0#
sh-5.0# export XDG_RUNTIME_DIR=/run/user/root && gst-launch-1.0 -e qtiqmmfsrc camera=1 name=camsrc ! video/x-raw,format=NV12,width=1280,height=720,framerate
=90/1 ! waylandsink fullscreen=true async=true sync=false
gbm_create_device(192): Info: backend name is: msm_drm
Setting pipeline to PAUSED ...
gbm_create_device(192): Info: backend name is: msm_drm
Pipeline is live and does not need PREROLL ...
Setting pipeline to PLAYING ...
New clock: GstSystemClock
```

Figure 47 CSI-1 OV9282 Camera Setup 1 Preview Command Execution

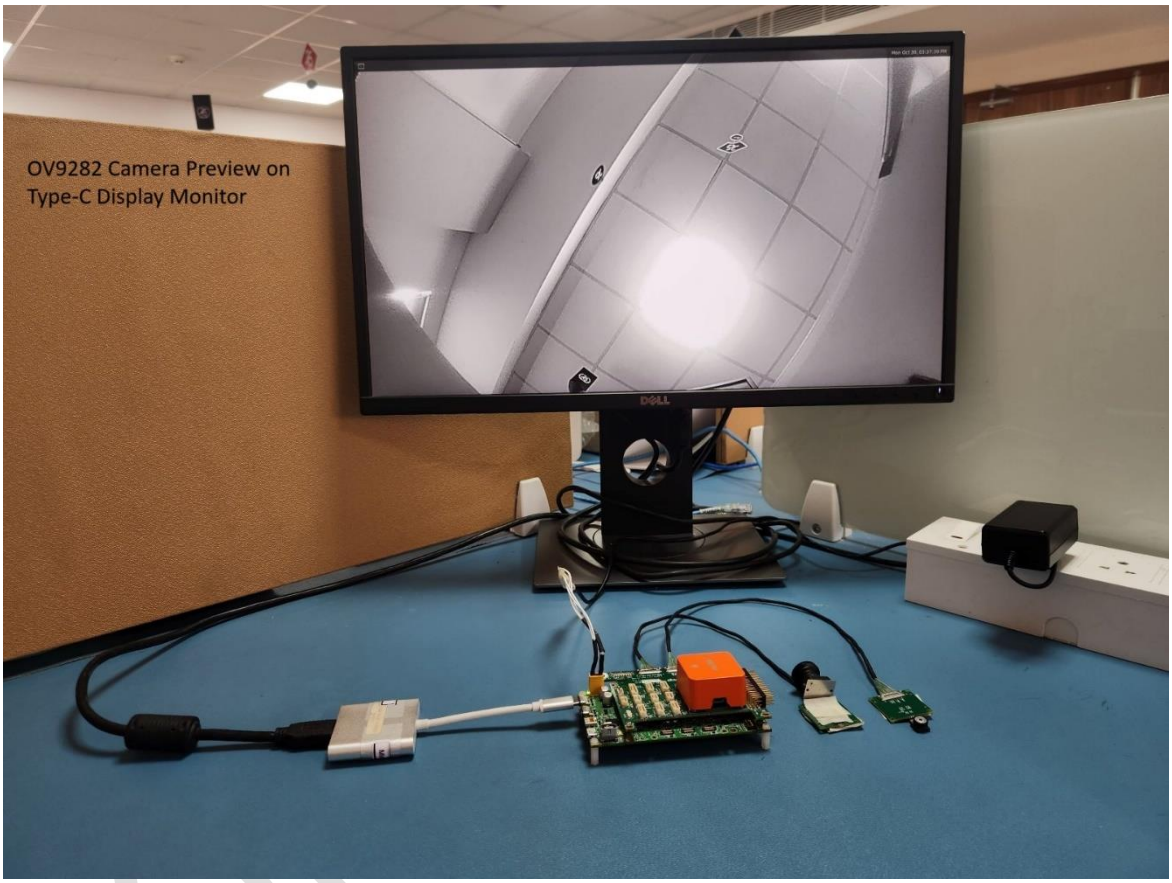

Figure 48 OV9282 Camera Setup 1 Preview on Type-C Display Monitor

CSI-0 IMX577 Camera Wi-Fi Stream:

Open an adb shell over Wi-Fi (Follow the ADB over Wi-Fi steps) and execute the following command to stream CSI-0 IMX577 camera video on Wi-Fi.

```
# echo 416 > /sys/class/gpio/export; echo out >
/sys/class/gpio/gpio416/direction; echo 0 >
/sys/class/gpio/gpio416/value; echo 416 >
/sys/class/gpio/unexport
```

(To enable the CSI-0 MIPI connector CSI signals)

```
# gst-launch-1.0 -e qtiqmmfsrc camera=0 name=camsrc ! video/x-
raw,format=NV12,width=3840,height=2160,framerate=30/1 !
qtic2venc ! queue ! h264parse ! rtph264pay config-interval=1
pt=96 ! udpsink host = 192.168.2.52 port=5502
```

Host PC Wi-Fi Video Playback:

Open VLC player with the following Wifi-Stream.sdp file.

Wifi-Stream.sdp file content is as follows.

```
v=0
m=video 5502 RTP/AVP 96
c=IN IP4 127.0.0.1
a=rtpmap:96 H264/90000
```

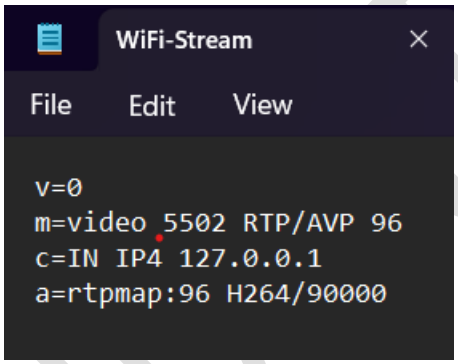

Figure 49 WiFi-Stream.sdp file content

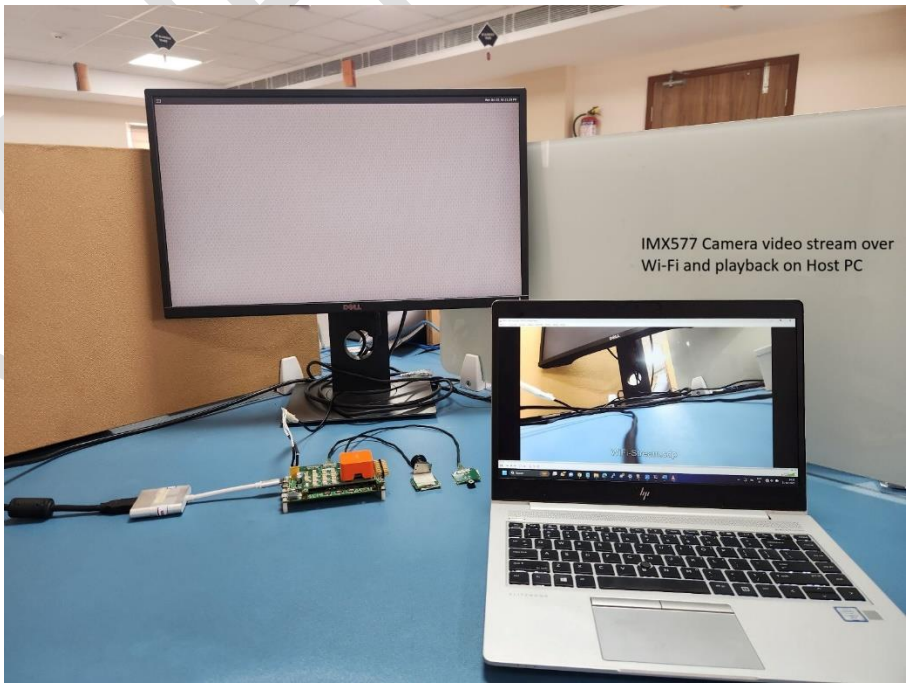

Figure 50 Camera Setup 1: CSI-0 IMX577 Camera Video Stream over Wi-Fi and Playback on Host PC

MRD5165 Platform Camera Setup 2

1. CSI-0 port connected to IMX577 camera module.
2. CSI-3 port connected to OV9282 camera module.

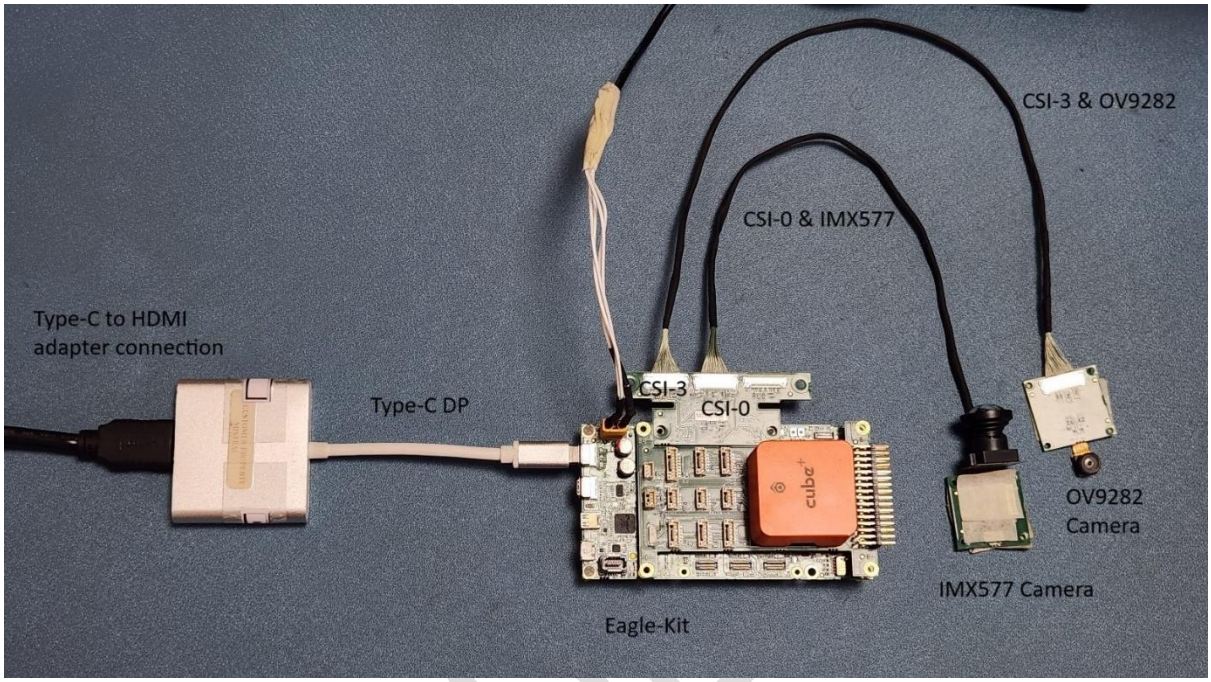

Figure 51 MRD5165 Platform Camera Setup 2

CSI-0 IMX577 and CSI-3 OV9282 Camera Preview: The procedure is identical to the MRD5165 Platform Camera Setup 1

CSI-0 IMX577 Wi-Fi Stream: The procedure is identical to the MRD5165 Platform Camera Setup 1

MRD5165 Platform Camera Setup 3

1. CSI-4 port connected to IMX577 camera module.
2. CSI-5 port connected to OV9282 camera module.

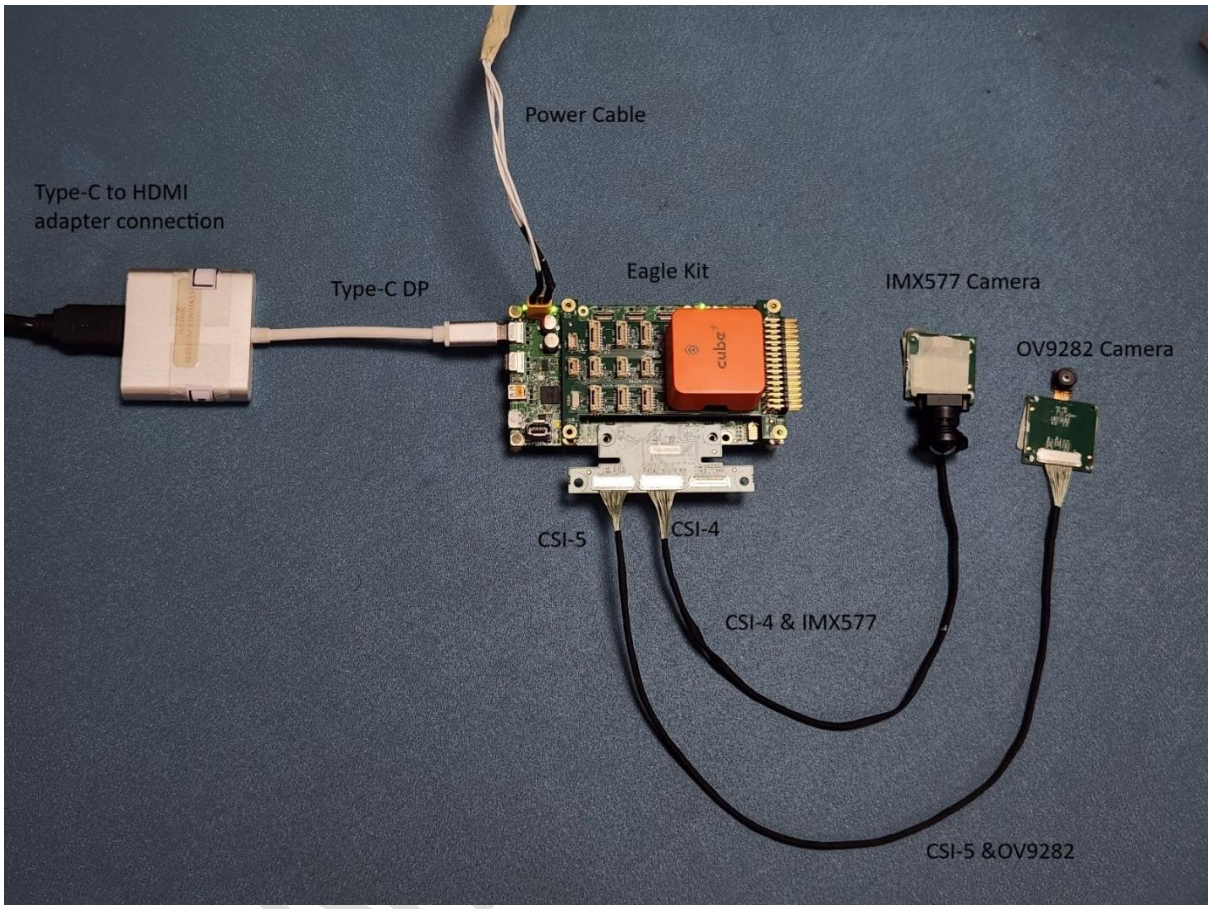

Figure 52 MRD5165 Platform Camera Setup 3

CSI-4 IMX577 Camera Preview:

Open an adb shell over Wi-Fi (Follow the ADB over Wi-Fi steps) and execute the following command to preview CSI-4 IMX577 camera on Type-c Display monitor.

```
#export XDG_RUNTIME_DIR=/run/user/root && gst-launch-1.0 -e
qtiqmmfsrc camera=2 name=camsrc ! video/x-
raw,format=NV12,width=3840,height=2160,framerate=30/1 !
waylandsink fullscreen=true async=true sync=false
```

```
sh-5.0#
sh-5.0# export XDG_RUNTIME_DIR=/run/user/root && gst-launch-1.0 -e qtiqmmfsrc camera=2 name=camsrc ! video/x-raw,format=NV12,width=3840,height=2160,framerat
e=30/1 ! waylandsink fullscreen=true async=true sync=false
gbm_create_device(192): Info: backend name is: msm_drm
Setting pipeline to PAUSED ...
gbm_create_device(192): Info: backend name is: msm_drm
Pipeline is live and does not need PREROLL ...
Setting pipeline to PLAYING ...
New clock: GstSystemClock
```

Figure 53 Camera Setup 3, CSI-4 IMX577 Camera Preview Command Execution

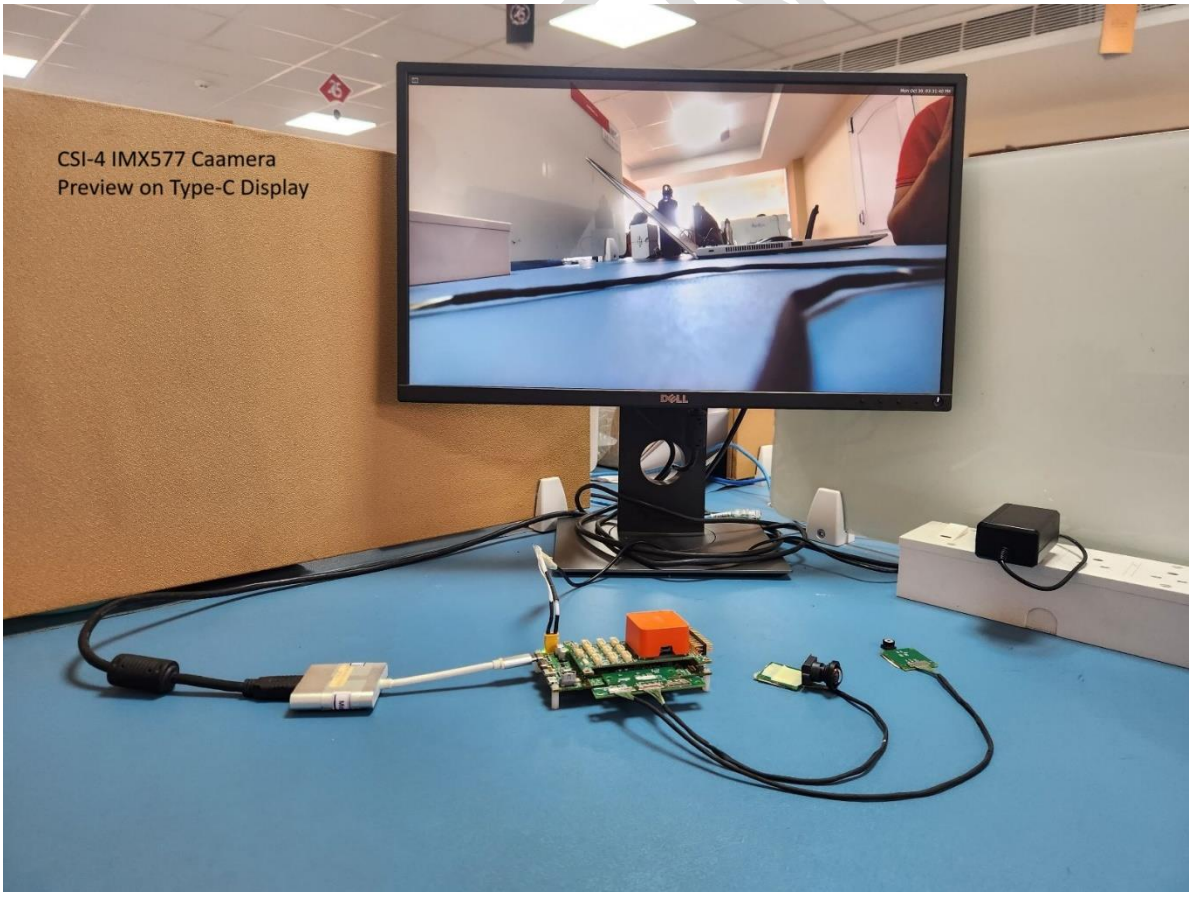

Figure 54 Camera Setup 3, CSI-4 IMX577 Camera Preview

CIS-5 OV9282 Camera Preview: The procedure is identical to the MRD5165 Platform Camera Setup 1

LR Wi-Fi (Long Range Wi-Fi) Station Mode

NOTE: By default, Eagle-Kit boot enables LR Wi-Fi in Access Point mode. To override the default AP mode configuration, follow the below steps.

Eagle-Kit Long Range Setup:

Figure 55 MRD5165 EAGLE KIT LR WIFI

- Open an adb shell over Type-C (Follow the ADB over Type-C) and execute the following commands to verify the Long-Range Wi-Fi Station mode connection.

a. Run the following commands to connect to a AP network available

```
# sh /data/misc/wifi/start_sta
```

b. Wait for the script to finish, then run “wpa_cli_w” command-line tool to add the network

```
# wpa_cli_w -i ath1
```

c. Add the network configurations

```
add_network
set_network 1 ssid "AP-Wi-Fi-Name"
set_network 1 psk "AP-Wi-Fi-password"
enable_network 1
```

Once after the connection is established, give ctrl+c to exit from "wpa_cli_w"

d. Check the “wlan0” iface IP address

```
# ifconfig ath1
```

```
sh-5.0# ifconfig ath1
ath1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.20.86  netmask 255.255.255.0  broadcast 192.168.20.255
        inet6 2409:408c:beb3:8a12:e81e:eb5d:37c3:57b0  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::114:e137:1633:bdb9  prefixlen 64  scopeid 0x20<link>
        inet6 2409:408c:beb3:8a12:37e2:bb2a:8ed8:f6cd  prefixlen 64  scopeid 0x0<global>
        ether 00:30:1a:50:73:49  txqueuelen 0  (Ethernet)
        RX packets 32  bytes 9149 (9.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 29  bytes 2709 (2.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 56 Eagle Kit LR Wi-Fi Station mode IP address

e. Run ping command to test the network connection.

```
# ping www.google.com
```

```
sh-5.0# ping www.google.com
PING www.google.com (142.250.183.228): 56 data bytes
64 bytes from 142.250.183.228: icmp_seq=0 ttl=112 time=79.307 ms
64 bytes from 142.250.183.228: icmp_seq=1 ttl=112 time=77.532 ms
64 bytes from 142.250.183.228: icmp_seq=2 ttl=112 time=85.859 ms
64 bytes from 142.250.183.228: icmp_seq=3 ttl=112 time=78.105 ms
64 bytes from 142.250.183.228: icmp_seq=5 ttl=112 time=170.160 ms
^C--- www.google.com ping statistics ---
6 packets transmitted, 5 packets received, 16% packet loss
round-trip min/avg/max/stddev = 77.532/98.193/170.160/36.107 ms
```

Figure 57 Eagle Kit LR Wi-Fi Station mode Ping test

LR Wi-Fi (Long Range Wi-Fi) Access Point Mode

NOTE: By default, Eagle-Kit boot enables LR Wi-Fi in Access Point mode. To override the default AP mode configuration, follow the below steps.

- Open an adb shell over Type-C (Follow the ADB over Type-C) and execute the following commands to verify the Long-Range Wi-Fi Station mode connection.

a. Disable the default AP mode configuration and reboot the board

```
# vim /etc/init.d/start_npm.sh
```

Comment the line: “sh /data/misc/wifi/start_sap &”

```
# reboot
```

b. Run the hostapd daemon with the default configuration file.

Modify the “/data/misc/wifi/examples/hostapd_he160_ch36.conf” “ssid” parameter for hotspot name and “wpa_passphrase” for WPA password.

```
# sh /data/misc/wifi/start_sap
```

Connect an external Wi-Fi device to Eagle-Kit Hotspot Default WPA password is “1234567890”

5. Additional Assistance

Please contact our support team for further assistance:

Phone : India : +91-80-4562 1100

Email : info@mistralsolutions.com

Web : https://mistralsolutions.com/MRD5165

Note: Information contained in this document is subject to change