MISTRAL

*Engineering the Intelligent Edge*

# *MRD8550 Merlin Kit Software Guide*

## Revision History

| Revision | Date | Description |
|---|---|---|
| 0.1 | July 27, 2025 | Preliminary version |
| 1.0 | Aug 11, 2025 | Baselined version |
| | | |

CONFIDENTIAL

# TABLE OF CONTENTS


1. Merlin Kit Overview ..... 6

Getting Started ..... 6

Merlin Plat orm Accessories ..... 6

2. Merlin Kit Setup ..... 7

Powering up Merlin Kit Board ..... 8

2.1. Serial/Debug Console ..... 8

2.2. Setting up ADB over Type-C ..... 12

3. Peripherals Validation ..... 13

3.1. RGB LEDs ..... 13

3.2. Switches ..... 14

3.3. Wi-Fi Station Mode ..... 16

3.4. Wi-Fi Access Point Mode ..... 17

3.5. Bluetooth ..... 19

3.6. SD Card ..... 22

3.7. Ethernet ..... 24

3.8. Sensors ..... 26

3.9. 5G Modem ..... 30

3.10. USB Port ..... 32

3.11. Speakers and DMICs ..... 33

3.12. GNSS ..... 35

3.13. Type-C Redriver Display ..... 36

3.14. HDMI-in Camera ..... 37

3.15. MRD8550 Platform Camera Ports ..... 39

3.16. MRD8550 LR Wi-Fi (Station Mode) ..... 43

3.17. MRD8550 LR Wi-Fi (Access point Mode) ..... 44

3.18. EEPROM ..... 45

3.19. CAN ..... 46

4. Additional Assistance ..... 47


CONFIDENTIAL

# TABLE OF FIGURES

**Figure 1: MRD8550 Merlin Kit** ..... 7
**Figure 2: MRD8550 Merlin Kit to Host PC connection** ..... 7
**Figure 3: Merlin-Kit Serial port setup** ..... 8
**Figure 4: Serial baud-rate setting** ..... 9
**Figure 5: Merlin-Kit Boot logs** ..... 10
**Figure 6: MRD8550 Merlin Kit shell Login** ..... 11
**Figure 7: Merlin-Kit adb connection** ..... 12
**Figure 8: Merlin Kit's RGB led** ..... 13
**Figure 9:MRD8550 Switches** ..... 14
**Figure 10: Merlin Kit's KEY_VOLUMEUP Event** ..... 14
**Figure 11:Merlin Kit's KEY_VOLUMEDOWN Event** ..... 15
**Figure 12:Merlin Kit's KEY_POWER Event** ..... 15
**Figure 13: Setting up Wi-Fi Station Mode** ..... 16
**Figure 14: If-config ip assignment** ..... 17
**Figure 15: Ping test** ..... 17
**Figure 16: MRD8550 WiFi AP Mode** ..... 18
**Figure 17: Wi-Fi AP Hosted by MRD8550** ..... 18
**Figure 18: Bluetooth app and gap_menu** ..... 20
**Figure 19: BT Inquiry** ..... 20
**Figure 20: BT device pairing** ..... 21
**Figure 21: Merlin Kit with SD Card** ..... 22
**Figure 22: Merlin Kit with SD Card Example** ..... 23
**Figure 23: IX Ethernet Connectors** ..... 24
**Figure 24: IX Ethernet Cable** ..... 24
**Figure 25: Ethernet ifconfig** ..... 25
**Figure 26: Ethernet ping test in MRD8550** ..... 25
**Figure 27: Merlin Kit Accelerometer test** ..... 26
**Figure 28: Merlin Kit Gyro test** ..... 27
**Figure 29: Merlin Kit Magnetometer test** ..... 28
**Figure 30: Merlin Kit Barometer test** ..... 29
**Figure 31:Modem SIM slots** ..... 30
**Figure 32: Merlin-Kit 5G Modem** ..... 30
**Figure 33: 5G Modem ping test** ..... 31
**Figure 34: USB Device Connection** ..... 32
**Figure 35: DMIC and Speakers** ..... 33
**Figure 36: GNSS UFL Antenna Connector** ..... 35
**Figure 37: Merlin-Kit Type-C Display Port** ..... 36
**Figure 38: Merlin-Kit Micro-HDMI Connector** ..... 37
**Figure 39: HDMI-IN Camera demo** ..... 38
**Figure 40: Merlin Kit Camera Ports** ..... 39
**Figure 41: Camera Setup 1** ..... 40
**Figure 42: IMX Cam streamer** ..... 41
**Figure 43: IMX Camera video streaming** ..... 41
**Figure 44: OE camera video stream command** ..... 42
**Figure 45: OE Camera video streaming** ..... 42
**Figure 46: Merlin LR Wi-Fi Module** ..... 43

CONFIDENTIAL

**Figure 47: LR WiFi ping test** ..... 44
**Figure 48: EEPROM hex dump command** ..... 45
**Figure 49: CAN Link show command** ..... 46

CONFIDENTIAL

# 1. Merlin Kit Overview

The Mistral MRD8550 Merlin Kit is built on Qualcomm's QCS8550 processor. Offering high-performance computing platform optimized for artificial intelligence (AI) and machine-learning inferencing accelerated development of innovative, power-efficient, high-computing robots and drones for enterprise, industrial, and professional service applications. It offers readily deployable SDKs 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 Merlin in IOT, Drones and Robotics.

## Getting Started

The Mistral QCS8550 Merlin Kit provides a flexible and robust platform for developers to evaluate system capabilities, test sample applications, migrate legacy software, develop new features, and interface with a wide range of peripheral devices. It supports a Linux-based software environment for application development, with the option to customize and update the system software using a chosen release of the MRD8550 system software.

The Development Kit Software User Guide offers a comprehensive overview o the platform's hardware and software, along with step-by-step instructions for setup and peripheral validation.

## Merlin Platform Accessories

Basic Accessories:

1. DC Adapter with chord
2. Custom B2B power cable with DC Jack
3. Wi-Fi Antenna
4. USB Type-C cable
5. I-PEX cable

Vision Accessories:

1. IMX577 Camera Modules
2. OV9282 Camera Modules

Cable Accessories:

1. HDMI Cable
2. IX Ethernet Cable
3. USB Micro-B Cable

Platform variant Kits:

1. 5G Modem Kit
2. LR Wi-Fi Kit
3. FCU Kit

CONFIDENTIAL

## 2. Merlin Kit Setup

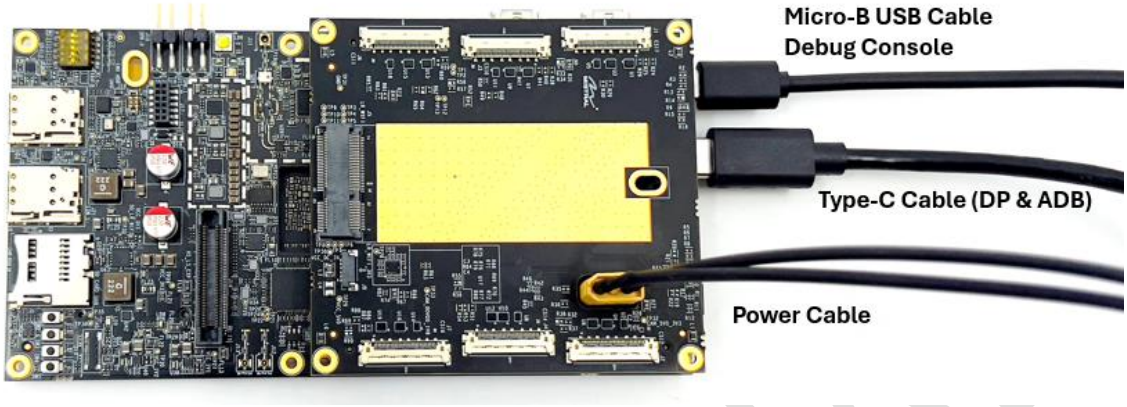


**Figure 1: MRD8550 Merlin Kit**

1. Connect the DC adapter to J6 using power cable
2. Connect a USB Type-C cable to J2 for ADB connection
3. Connect a Micro-B cable to J1 for Debug console access

**Figure 2: MRD8550 Merlin Kit to Host PC connection**

# Powering up Merlin Kit Board

**Note:** MRD8550 Platform must be programmed with a flat build. Please refer to the document MS_MRD8550_Programming_Guide to Program the kit.

## 2.1. Serial/Debug Console

- Connect the Merlin Kit's Micro-B USB connector to Host PC, as illustrated in the Fig 2 image.
- Launch a Tera Term application on Host PC to observe boot logs and access the board.

**In Tera Term**:

a) Select the **Merlin-Kit Serial Console** (Micro-B connection) **COM port**.
b) b) Configure the serial settings to **115200 baud rate**, **8 data bits**, **no parity**, **1 stop bit** (**115200 – 8N1**).

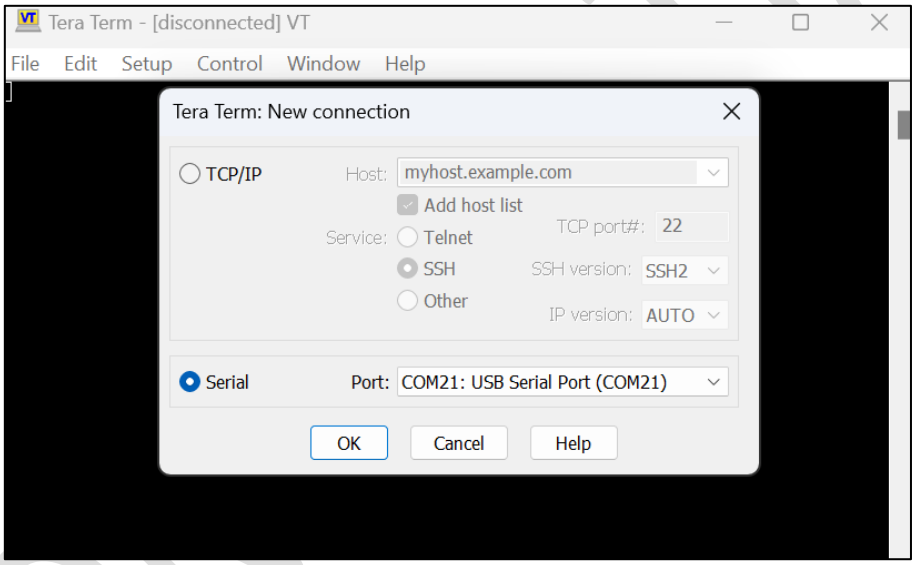


**Figure 3: Merlin-Kit Serial port setup**

CONFIDENTIAL

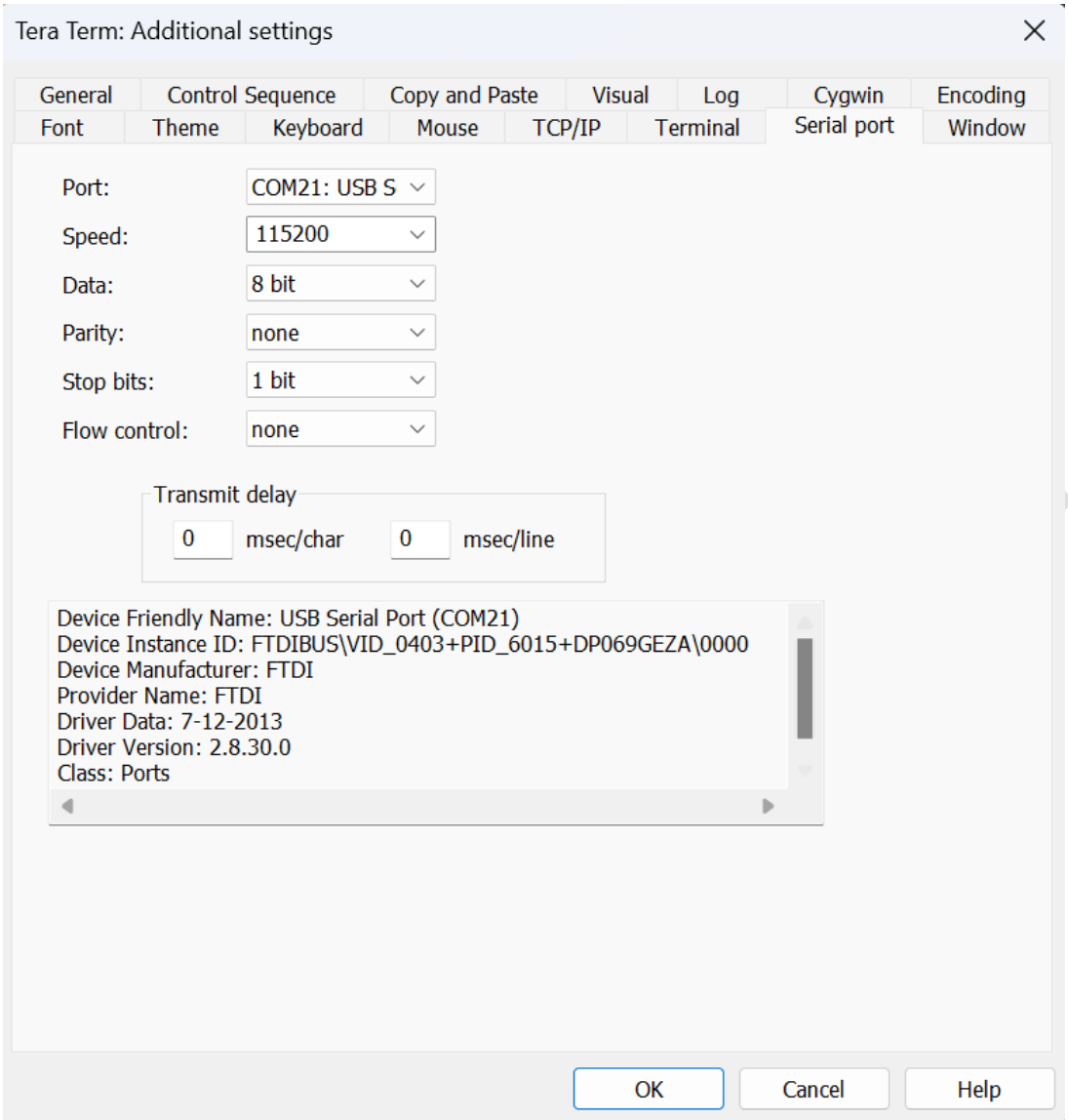


**Figure 4: Serial baud-rate setting**

Merlin Kit Boot Logs:

```
COM21 - Tera Term VT
File  Edit  Setup  Control  Window  Help
Format: Log Type - Time(microsec) - Message - Optional Info
Log Type: B - Since Boot(Power On Reset),  D - Delta,  S - Statistic
S - QC_IMAGE_VERSION_STRING=BOOT.MXF.2.1.1.c1-00019-KAILUA-1
S - IMAGE_VARIANT_STRING=SocKailuaLAA
S - OEM_IMAGE_VERSION_STRING=ip-10-195-202-164
S - Boot Interface: UFS
S - Secure Boot: Off
S - Boot Config @ 0x221c8600 = 0x00000001
S - JTAG ID @ 0x221c8744 = 0x202520e1
S - OEM ID @ 0x221c8700 = 0x00000000
S - Serial Number @ 0x221c8610 = 0x000004377e00c5e7
S - OEM Config Row 0 @ 0x221c2110 = 0x0000000000000000
S - OEM Config Row 1 @ 0x221c2118 = 0x0000000000000000
S - Feature Config Row 0 @ 0x221c2138 = 0x003ffffffffbf00
S - Feature Config Row 1 @ 0x221c2140 = 0x0000000000000000
S - OEM LCS:PROD
S - Core 0 Frequency, 1363 MHz
S - PBL Patch Ver: 3
S - PBL freq: 600 MHZ
S - -------------------------
PBL Timestamps (in usecs)
-------------------------
APBL start: 345
APBL Boot Media Detect Start: 9650
APBL Boot Media Detect End: 53741
APBL SHELF Load+Auth Start: 53743
        APBL ELF MD Load Start: 57545
        APBL ELF MD Load End: 59031
        APBL ELF Seg Load Start: 59031
        APBL ELF Seg Load End: 63891
                APBL SEQ Valid Set: 63935
                APBL SEQ SA ACK Rcvd: 65960
        APBL ELF MD Load Start: 67205
        APBL ELF MD Load End: 70308
        APBL ELF Auth Start: 70318
        APBL ELF Auth End: 80108
        APBL ELF Seg Load Start: 80108
        APBL ELF Seg Load End: 120143
        APBL ELF Seg Verify Start: 120143
        APBL ELF Seg Verify End: 125425
        APBL ELF MD Load Start: 126566
        APBL ELF MD Load End: 127831
        APBL ELF Auth Start: 127835
        APBL ELF Auth End: 137608
        APBL ELF Seg Load Start: 137610
        APBL ELF Seg Load End: 146236
        APBL ELF Seg Verify Start: 146238
        APBL ELF Seg Verify End: 150735
APBL SHELF Load+Auth End: 150928
                APBL TME PBL Exit: 150933
                APBL XBL_SC Exit Check: 151645
                APBL XBL_SC Exit Ack: 163111
APBL Exit: 163116
B -    172965 - SBL1, Start
B -    360205 - SBL1 BUILD @ 19:24:19 on Feb 14 2024
B -    371551 - usb: usb_eud_serial   , 0x7e00c5e7
```

**Figure 5: Merlin-Kit Boot logs**

Debug Serial Console Login credentials are as follows -

```
Username: root
Password: oelinux123
```

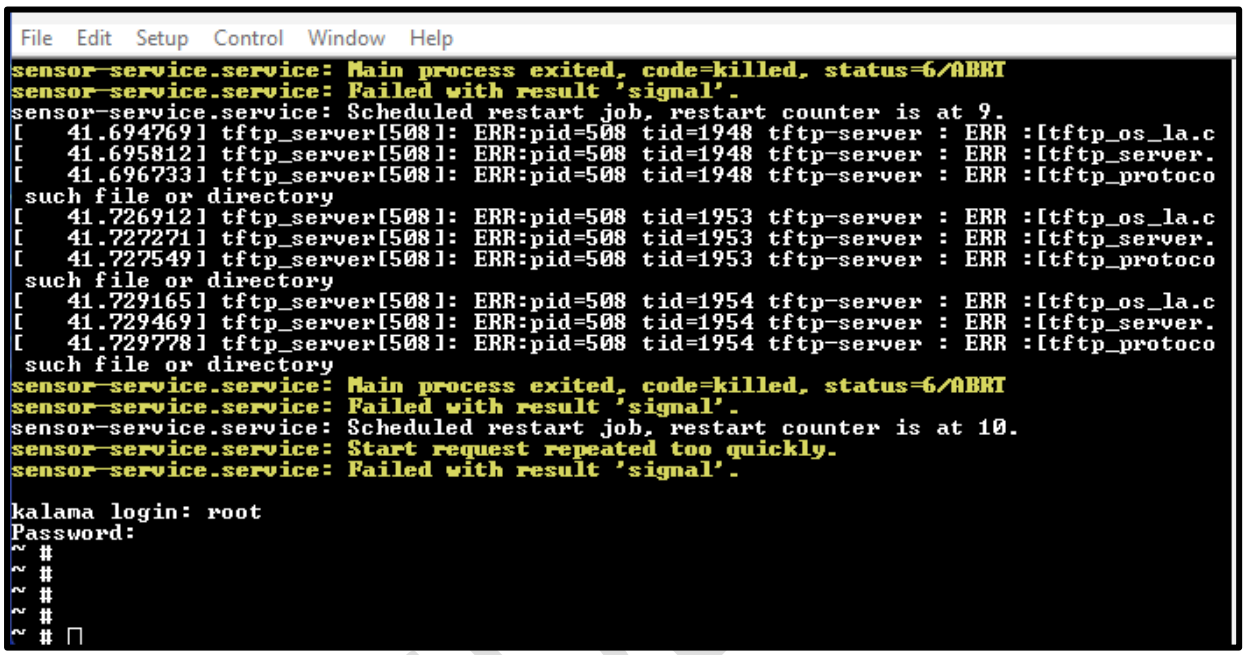


Figure 6: MRD8550 Merlin Kit shell Login

CONFID

## 2.2. Setting up ADB over Type-C

- Connect the Merlin Kit Type-C connector to Host PC as shown in the setup Fig 2 image.
- Power up the Merlin Kit with DC power adapter
- Open a CMD terminal on Host PC and run the following commands.

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

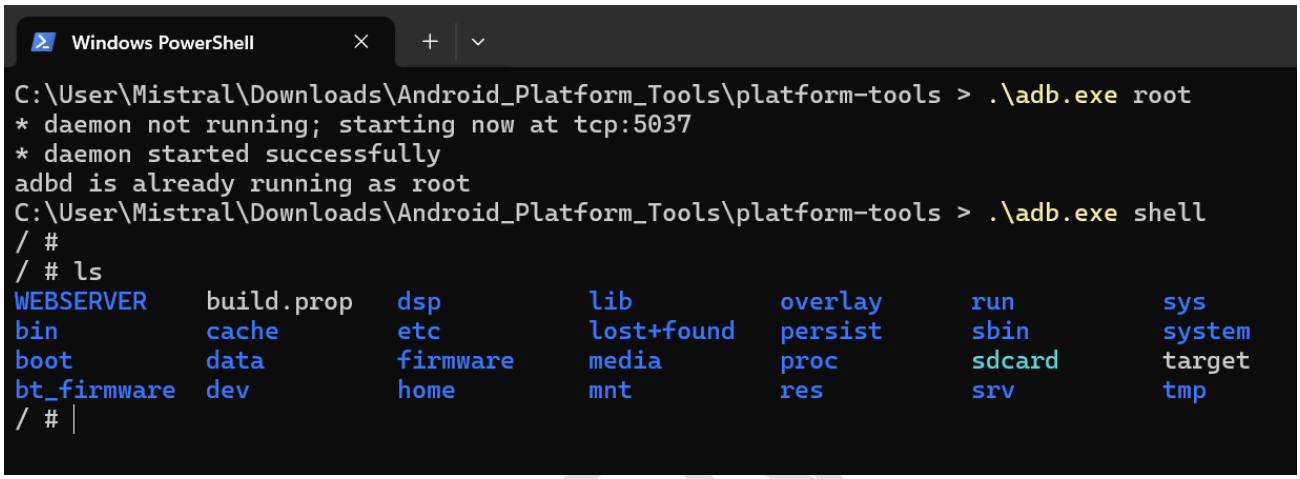


Figure 7: Merlin-Kit adb connection

CONFIDENTIAL

# 3. Peripherals Validation

## 3.1. RGB LEDs

**MRD8550 Kit Setup:**

- Open an adb shell 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
```

**Example:**

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

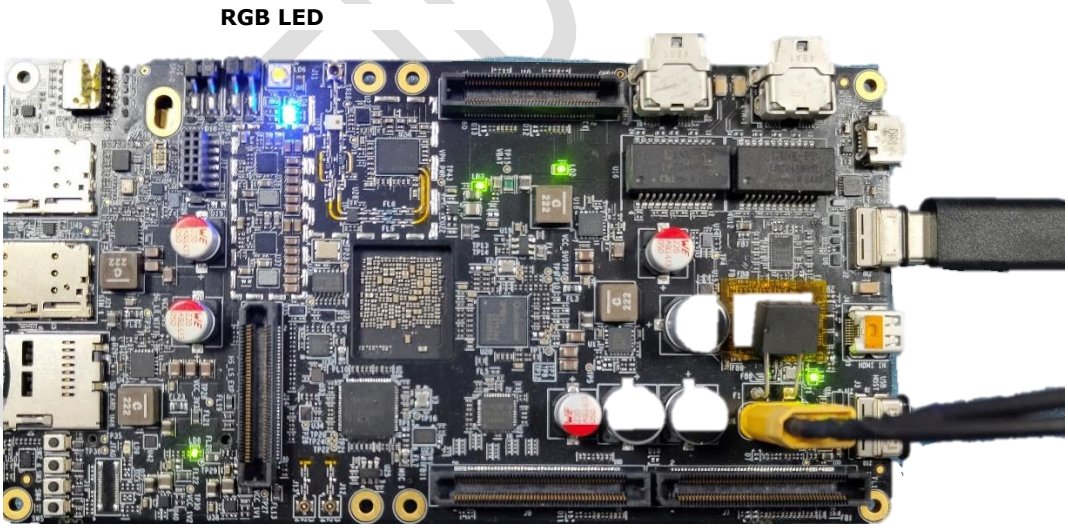


Figure 8: Merlin Kit's RGB led

## 3.2. Switches

**MRD8550 Kit Setup:**

- Open an adb shell and execute the following commands to verify the VOLUME_UP, VOLUME_DOWN, POWER_KEY switches events.

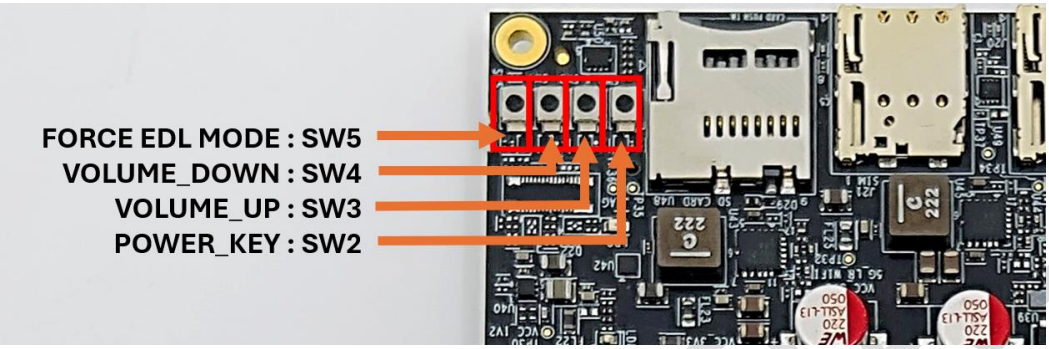


Figure 9:MRD8550 Switches

VOL-UP KEY:

- Execute the following command and press the VOLUME_UP: **SW3** button.

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

Observe the event logs for KEY_VOLUMEUP button press.

```
/ # evtest /dev/input/event0
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 315965015.899672, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), value 1
Event: time 315965015.899672, -------------- SYN_REPORT ------------
Event: time 315965016.229002, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), value 0
Event: time 315965016.229002, -------------- SYN_REPORT ------------
Event: time 315965016.960388, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), value 1
Event: time 315965016.960388, -------------- SYN_REPORT ------------
Event: time 315965017.394779, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), value 0
Event: time 315965017.394779, -------------- SYN_REPORT ------------
```

Figure 10: Merlin Kit's KEY_VOLUMEUP Event

VOL-DOWN KEY:

- Execute the following command and press the VOL-DOWN: **SW4** button.

```
# evtest /dev/input/event2
```

Observe the event logs for KEY_VOLUMEDOWN button press.

```
/ # evtest /dev/input/event2
Input driver version is 1.0.1
Input device ID: bus 0x0 vendor 0x0 product 0x0 version 0x0
Input device name: "pmic_resin"
Supported events:
  Event type 0 (EV_SYN)
  Event type 1 (EV_KEY)
    Event code 114 (KEY_VOLUMEDOWN)
Properties:
Testing ... (interrupt to exit)
Event: time 315965157.723921, type 1 (EV_KEY), code 114 (KEY_VOLUMEDOWN), value 1
Event: time 315965157.723921, -------------- SYN_REPORT ------------
Event: time 315965157.983141, type 1 (EV_KEY), code 114 (KEY_VOLUMEDOWN), value 0
Event: time 315965157.983141, -------------- SYN_REPORT ------------
Event: time 315965158.858220, type 1 (EV_KEY), code 114 (KEY_VOLUMEDOWN), value 1
Event: time 315965158.858220, -------------- SYN_REPORT ------------
Event: time 315965159.396418, type 1 (EV_KEY), code 114 (KEY_VOLUMEDOWN), value 0
Event: time 315965159.396418, -------------- SYN_REPORT ------------
```

**Figure 11:Merlin Kit's KEY_VOLUMEDOWN Event**

POWER KEY:

- Execute the following command and press the POWER: **SW2** button.

  *# evtest /dev/input/event1*

  Observe the event logs for KEY_POWER button press.

```
/ # evtest /dev/input/event1
Input driver version is 1.0.1
Input device ID: bus 0x0 vendor 0x0 product 0x0 version 0x0
Input device name: "pmic_pwrkey"
Supported events:
  Event type 0 (EV_SYN)
  Event type 1 (EV_KEY)
    Event code 116 (KEY_POWER)
Properties:
Testing ... (interrupt to exit)
Event: time 315965094.386178, type 1 (EV_KEY), code 116 (KEY_POWER), value 1
Event: time 315965094.386178, -------------- SYN_REPORT ------------
Event: time 315965095.248150, type 1 (EV_KEY), code 116 (KEY_POWER), value 0
Event: time 315965095.248150, -------------- SYN_REPORT ------------
Event: time 315965096.715557, type 1 (EV_KEY), code 116 (KEY_POWER), value 1
Event: time 315965096.715557, -------------- SYN_REPORT ------------
Event: time 315965096.890551, type 1 (EV_KEY), code 116 (KEY_POWER), value 0
Event: time 315965096.890551, -------------- SYN_REPORT ------------
```

**Figure 12:Merlin Kit's KEY_POWER Event**

## 3.3. Wi-Fi Station Mode

**MRD8550 Setup:**

- Open the adb shell and execute the following commands to verify the Wi-Fi Station mode connection.

**Note:** The default Wi-Fi Mode will be set as AP Mode

a. Enable the wlan0 interface

```
# ifconfig wlan0 up
```

b. Configure the wpa_supplicant with "SSID" and "PASSWORD".

```
# wpa_passphrase "SSID" "Password" | tee
/etc/wpa_supplicant.conf
```

c. Initialize the wpa_supplicant with respective conf file

```
# wpa_supplicant -B -i wlan0 -c
/etc/wpa_supplicant.conf
```

d. Initiate the udhcpc for ip address.

```
# udhcpc -i wlan0
```

```
/ # ifconfig wlan0 up
/ # wpa_passphrase "Mistral" "PASSWORD" | tee /etc/wpa_supplicant.conf
network={
        ssid="Mistral"
        #psk="PASSWORD"
        psk=c6fb43c5d11c2676a9e2c78cdf9be5c9bc62879edc8529825b28375e9511c7a6
}
/ # wpa_supplicant -B -i wlan0 -c /etc/wpa_supplicant.conf
Successfully initialized wpa_supplicant
/ # udhcpc -i wlan0
udhcpc: started, v1.35.0
udhcpc: broadcasting discover
udhcpc: broadcasting select for 10.237.38.157, server 10.237.38.147
udhcpc: lease of 10.237.38.157 obtained from 10.237.38.147, lease time 3599
/etc/udhcpc.d/50default: Adding DNS 10.237.38.147
```

**Figure 13: Setting up Wi-Fi Station Mode**

e. Check the "wlan0" interface IP address

```
# ifconfig wlan0
```

```
sh-5.0# ifconfig wlan0
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.[redacted]  netmask 255.255.255.0  broadcast 192.[redacted]
        inet6 fe80::[redacted]  prefixlen 64  scopeid 0x20<link>
        ether 00:0[redacted]  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 14: If-config ip assignment**

f. Run ping command to test the network connection.

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

```
# ping google.co.in

Pinging google.co.in [142.250.67.35] with 32 bytes of data:
Reply from 142.250.67.35: bytes=32 time=26ms TTL=115
Reply from 142.250.67.35: bytes=32 time=26ms TTL=115
Reply from 142.250.67.35: bytes=32 time=26ms TTL=115
Reply from 142.250.67.35: bytes=32 time=24ms TTL=115

Ping statistics for 142.250.67.35:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 24ms, Maximum = 26ms, Average = 25ms
```

**Figure 15: Ping test**

**Note: Reboot the board to enable the Merlin-Kit in Wi-Fi Hotspot mode / AP Mode (Default)**

## 3.4. Wi-Fi Access Point Mode

**MRD8550 Platform Setup**:

- Open an adb shell 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
# killall dhcpcd
# killall dnsmasq
# ifconfig wlan0 up
```

b. Initialize the hostpad with respective hostpad.conf

```
# hostapd /data/misc/wifi/hostapd.conf &
```

c. Configure the ip address for the device

```
# ifconfig wlan0 192.168.3.1
# ifconfig wlan0
```

d. Enable the dnsmasq

```
# dnsmasq -i wlan0 -l /data/dnsmasq.leases --no-daemon --
no-resolv --no-poll --port=5353 --dhcp-
range=192.168.3.100,192.168.3.200,1h &
```

```
/ #  killall dnsmasq
/ #  killall dhcpcd
killall: dhcpcd: no process killed
/ # hostapd /data/misc/wifi/hostapd.conf &
/ # debug, set loglevel
wlan0: interface state UNINITIALIZED->COUNTRY_UPDATE
wlan0: interface state COUNTRY_UPDATE->HT_SCAN
20/40 MHz operation not permitted on channel pri=6 sec=2 based on overlapping BSSes
Fallback to 20 MHz
wlan0: interface state HT_SCAN->ENABLED
wlan0: AP-ENABLED
wlan0: IEEE 802.11 driver had channel switch: freq=2437, ht=1, vht_ch=0x0, he_ch=0x0, eht_ch=0x0, offset=0, width=1 (2
0
wlan0: CTRL-EVENT-CHANNEL-SWITCH freq=2437 ht_enabled=1 ch_offset=0 ch_width=20 MHz cf1=2437 cf2=0 dfs=0
wlan0: AP-CSA-FINISHED freq=2437 dfs=0

/ # ifconfig wlan0 192.168.3.1
/ # ifconfig wlan0
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.3.1  netmask 255.255.255.0  broadcast 192.168.3.255
        inet6 fe80::203:7fff:fe12:839a  prefixlen 64  scopeid 0x20<link>
        ether 00:03:7f:12:83:9a  txqueuelen 3000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1  bytes 70 (70.0 B)
        TX errors 0  dropped 7 overruns 0  carrier 0  collisions 0

/ # dnsmasq -i wlan0 -l /data/dnsmasq.leases --no-daemon --no-resolv --no-poll --port=5353 --dhcp-range=192.168.3.100,
/ # dnsmasq: started, version 2.81 cachesize 150
dnsmasq: compile time options: IPv6 GNU-getopt no-DBus no-UBus no-i18n no-IDN DHCP DHCPv6 no-Lua TFTP no-conntrack ips
-detect inotify dumpfile
dnsmasq: warning: no upstream servers configured
dnsmasq-dhcp: DHCP, IP range 10.0.0.10 -- 10.0.0.200, lease time 2h
dnsmasq-dhcp: DHCP, IP range 192.168.3.100 -- 192.168.3.200, lease time 1h
dnsmasq: read /etc/hosts - 7 addresses
```

**Figure 16: MRD8550 WiFi AP Mode**

e. Connect an external Wi-Fi device to Hotspot hosted by MRD8550 Platform
Default SSID is "**Merlin-Kit XXXXXXX**" Password is "**1234567890**"

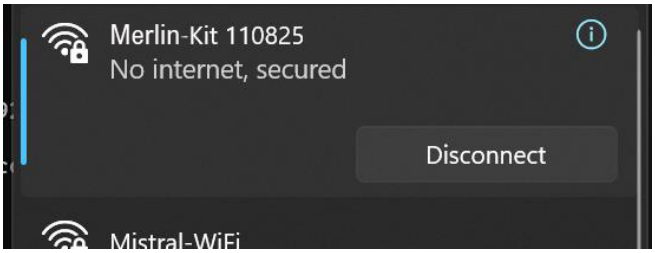


**Figure 17: Wi-Fi AP Hosted by MRD8550**

f. To change the SSID and PASSWORD, edit the /data/misc/wifi/hostapd.conf File, search for **ssid=" Merlin-Kit XXXXXXX"** and **wpa_passphrase="1234567890".**

## 3.5. Bluetooth

**MRD8550 Platform Setup**

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

1. Validating Bluetooth

   Execute the “btproperty” and “btapp” and run the following commands to pair a BT device.

```
# btproperty & btapp

➢ gap_menu
➢ enable
➢ inquiry
➢ pair <address>
➢ yes
➢ bonded_list
```

2. Connecting BT device

```
# enable
# inquiry
# pair <address to be connected>
```

CONFIDENTIAL

```
/ # btapp
enable
 Wrong option selected

***************** Menu *******************
         gap_menu
         pan_menu
         rsp_menu
         test_menu
         a2dp_sink_menu
         hfp_client_menu
         gattctest_menu
         gattstest_menu
         hid_menu
         pbap_client_menu
         opp_menu
/ # btapp
gap_menu

***************** Menu *******************
         enable
         disable
         inquiry
         cancel_inquiry
         pair<space><bt_address>     eg. pair 00:11:22:33:44:55
         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_
         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
 ******************************************
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: Diag_LSM_Init: invoked for pid: 1850 with init_count: 0
diag:successfully connected to socket 91
diag: Diag_LSM_Init: done for pid: 1850 with init_count: 1
diag: Successfully registered commands with the driver
 current State = 0, new state = 1
 BT State is ON
```

Figure 18: Bluetooth app and gap_menu

```
 BT State is ON
inquiry
 Inquiry Started
Device Found details:
Found device Addr: 2c:be:eb:2c:69:e8
Found device Name: mistral
Device Type is: 1
Device Found details:
Found device Addr: 4f:a4:8e:76:9f:e4
Found device Name:
Device Type is: 2
```

Figure 19: BT Inquiry

```
pair 2c:be:eb:2c:69:e8
 ACL state:0 change with reason 00 for device: 2c:be:eb:2c:69:e8

**************************************************
 BT pairing request::Device mistral::Pairing Code:: 324481
**************************************************
 ** Please enter yes / no **
yes

**************************************************
 Pairing state for mistral is BONDED
**************************************************
 ACL state:1 change with reason 13 for device: 2c:be:eb:2c:69:e8
```

**Figure 20: BT device pairing**

3. Run the commands below to exit the btapp

```
# main_menu
# exit
```

CONFIDENTIAL

## 3.6. SD Card

Merlin-Kit Setup:

- Connect a SD card to the Merlin Kit board.

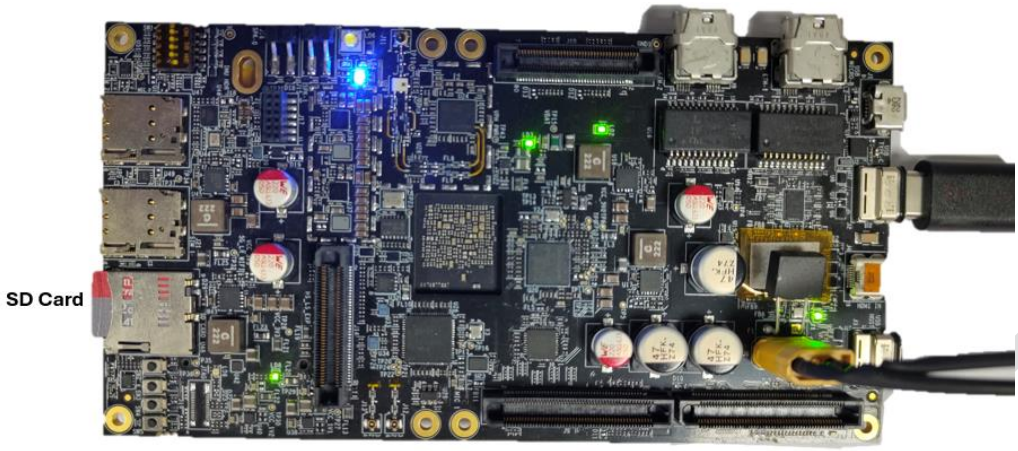


**Figure 21: Merlin Kit with SD Card**

- Open an adb shell and execute the following commands to verify the SD card.

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

  b. Check details of selected node
  ```
  # fdisk -l /mnt/<selected device>
  ```

  c. Mount the SD card partition device node
  ```
  # mkdir -p /mnt/<selected device>
  ```

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

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

CONFIDENTIAL

```
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 22: Merlin Kit with SD Card Example*

e. Unmount the SD card after the verification
   Example command:
   `# umount /dev/mmcblk2p1`

CONFIDE

## 3.7. Ethernet

**Merlin-Kit Setup**:

- Open an adb shell and execute the following commands to verify the Ethernet connection.
- Connect ethernet cable to the MRD8550 Platform

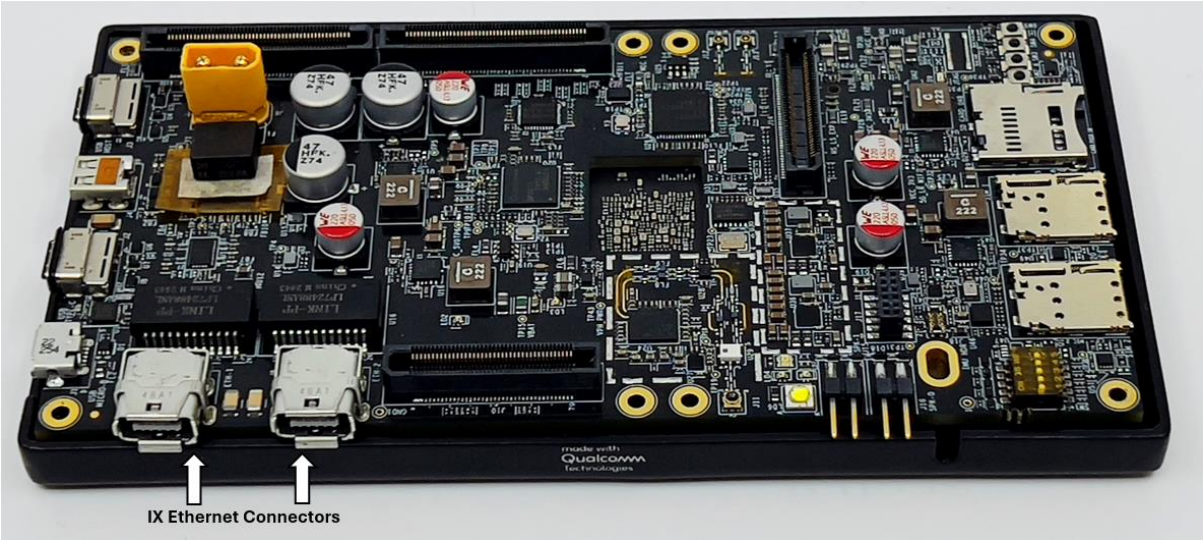


Figure 23: IX Ethernet Connectors

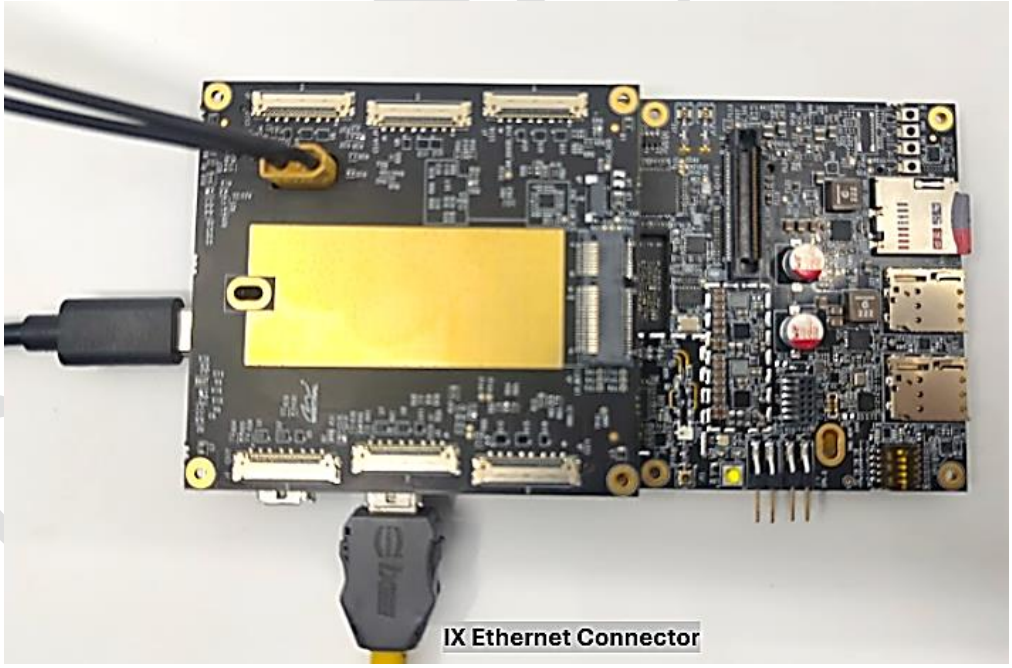


Figure 24: IX Ethernet Cable

```
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 25: Ethernet ifconfig**

1. Verify the ethernet interface "eth0" IP address

   ```
   # ifconfig eth0
   ```

2. Set a static ip address for MRD8550's eth0 interface and enable route.

   ```
   # ifconfig eth0 192.168.13.159 up
   # ip route add default via 192.168.13.1 dev eth0
   ```

3. Set a static ip address for Host PC's eth0 interface.

   ```
   # sudo ifconfig <interface name> 192.168.13.132
   ```

4. Run ping command to on MRD8550 Platform to test connection with Host PC

   ```
   # ping 192.168.13.132
   ```

```
# ping 192.168.13.132

Pinging 192.168.13.132 with 32 bytes of data:
Reply from 192.168.13.132: bytes=32 time<1ms TTL=128
Reply from 192.168.13.132: bytes=32 time<1ms TTL=128
Reply from 192.168.13.132: bytes=32 time<1ms TTL=128
Reply from 192.168.13.132: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.13.132:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
#
```

**Figure 26: Ethernet ping test in MRD8550**

**Note:** After making the eth link down and up, the ping command may stop working, This is a known issue, and we are actively working for a resolution.

## 3.8. Sensors

**MRD8550 Platform Setup:**

- Open an adb shell and execute the following commands to verify all the sensors.

**Accelerometer Sensor Test:**

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 the duration is set 5 seconds and sample rate set to the 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 27: Merlin Kit Accelerometer test

**Gyroscope Sensor Test:**

Run 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 the 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 28: Merlin Kit Gyro test**

**Magnetometer Test:**

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

```
/ # ssc_drva_test -sensor=mag -duration=5 -sample_rate=50
2 ssc_drva_test version 1.24k
2 ssc_drva_test -sensor=mag -duration=5 -sample_rate=50
diag: Diag_LSM_Init: invoked for pid: 927115 with init_count: 0
diag:successfully connected to socket 5
diag: Diag_LSM_Init: done for pid: 927115 with init_count: 1
2 event_cb attribute event for da_test
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. -rumifact=1
2 -time_to_first_event=2129123
2 -time_to_last_event=-176583
2 -sample_ts=1137481055373
2 -total_samples=245
2 -avg_delta=382431
2 -recvd_phy_config_sample_rate=50
2 -random_seed_used=3513670264
2 -num_request_sent=2
2 -first_sample_timestamp=1137387182431
2 received event: PASS
2 enter send_memory_log_req cookie: 2
2 exit send_memory_log_req
2 PASS
/ #
```

**Figure 29: Merlin Kit Magnetometer test**

**Pressure Sensor Test:**

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

```
/ # ssc_drva_test -sensor=pressure -duration=5 -sample_rate=50
3 ssc_drva_test version 1.24k
3 ssc_drva_test -sensor=pressure -duration=5 -sample_rate=50
diag: Diag_LSM_Init: invoked for pid: 929296 with init_count: 0
diag:successfully connected to socket 5
diag: Diag_LSM_Init: done for pid: 929296 with init_count: 1
3 event_cb attribute event for da_test
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. -rumifact=1
3 -time_to_first_event=144579
3 -time_to_last_event=-95857840
3 -sample_ts=1140136906143
3 -total_samples=1
3 -avg_delta=0
3 -recvd_phy_config_sample_rate=0
3 -random_seed_used=1874553349
3 -num_request_sent=2
3 -first_sample_timestamp=1140041047743
3 received event: PASS
3 enter send_memory_log_req cookie: 3
3 exit send_memory_log_req
3 PASS
/ #
```

**Figure 30: Merlin Kit Barometer test**

## 3.9. 5G Modem

MRD8550 Platform supports connecting the PCIe/USB based cellular connectivity modems. The default platform (Merlin 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. Default working state for the modem will be in USB Mode.

**Note**: Veriy whether the 5G module is USB mode and ollow the appropriate instructions below or network setup.

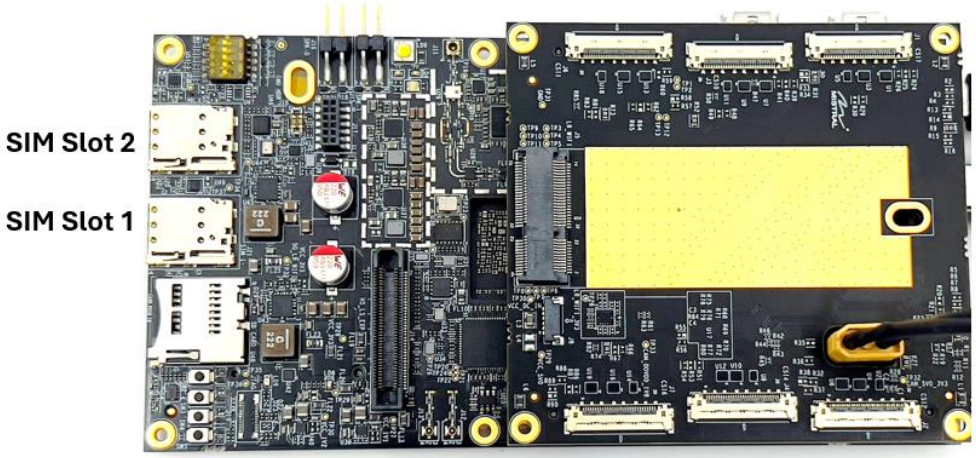


Figure 31:Modem SIM slots

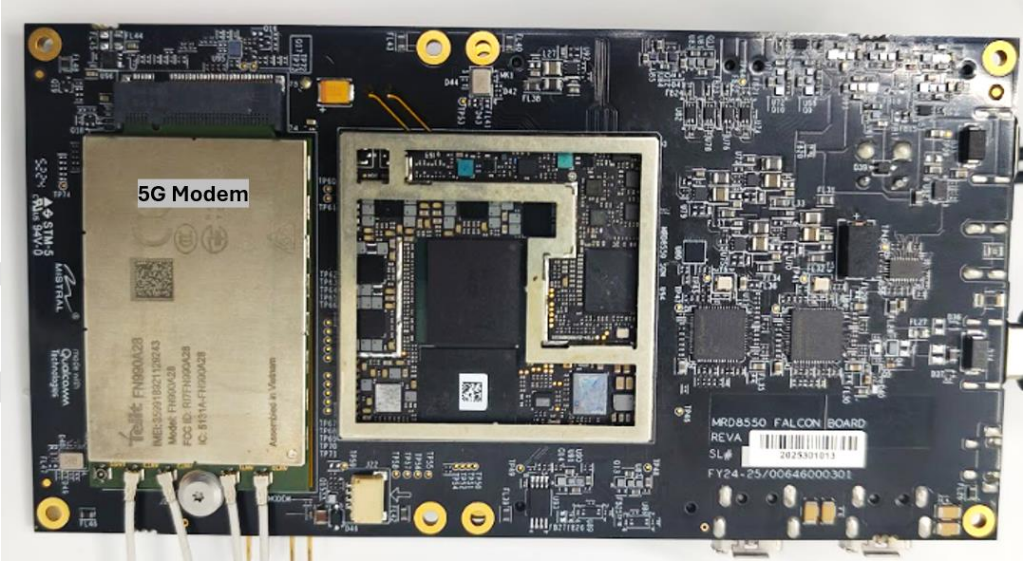


Figure 32: Merlin-Kit 5G Modem

MRD8550 Platform Setup:

- Connect a 5G module with antenna to the MRD8550 Platform and power up the setup.
- Connect a SIM card to SIM1 slot.
- Open an adb shell and execute the following commands to verify the 5G module connection.

Check the device availability with

```
# mbimcli -d /dev/cdc-wdm0 --query-device-caps
```

Check the SIM status with

```
# mbimcli -d /dev/cdc-wdm0 --query-subscriber-ready-status
```

It should return SIM status

Connect to you service provider

```
# mbimcli -d /dev/cdc-wdm0 --connect="apn=airtelgprs.com"
```

Wait for: "Network Started successfully"

Get the IP with the udhpc

```
# udhcpc -i wwan0
```

Perform ping test with google.co.in

```
# ping google.co.in

Pinging google.co.in [142.250.67.35] with 32 bytes of data:
Reply from 142.250.67.35: bytes=32 time=26ms TTL=115
Reply from 142.250.67.35: bytes=32 time=26ms TTL=115
Reply from 142.250.67.35: bytes=32 time=26ms TTL=115
Reply from 142.250.67.35: bytes=32 time=24ms TTL=115

Ping statistics for 142.250.67.35:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 24ms, Maximum = 26ms, Average = 25ms
```

**Figure 33: 5G Modem ping test**

CONFIDENTIAL

## 3.10. USB Port

**MRD8550 Platform Setup**:

- Open an adb shell and execute the following commands to verify USB Port.
- Connect a USB Drive to the USB type-C Port.

**Figure 34: USB Device Connection**

1. Check the device recognition [ /dev/sdX]

```
# dmesg | tail
# lsblk
```

2. Mount the connected device

```
# mount -o remount,rw /
# mkdir /mnt/usb
# mount /dev/sdX1 /mnt/usb
```

3. Move to device and create file in connected device

```
# cd /mnt/usb
# echo "USB Type-C 1 test" > testfile.txt
```

4. Verify the file copied

```
# cat testfile.txt
```

5. Unmount the connected device

```
# umount /dev/sdX
```

## 3.11. Speakers and DMICs

**MRD8550 Platform Setup:**

- Open an adb shell and execute the following commands to verify DMIC and speakers.
- Connect Speakers to the speaker port.

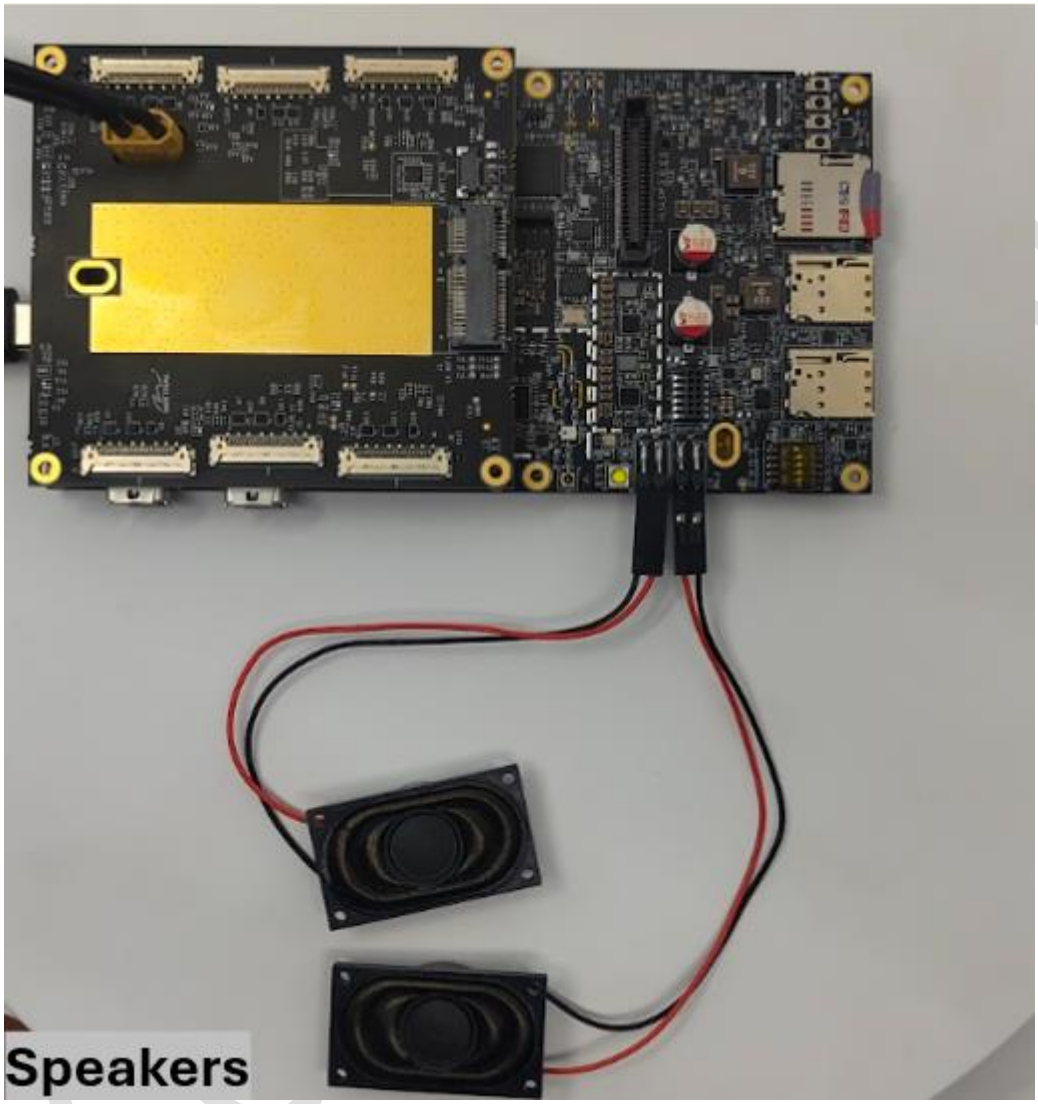


Figure 35: DMIC and Speakers

**Recording with DMIC**

1. Command to disable pulse audio

```
# systemctl stop pulseaudio
```

2. Routing commands to enable DMIC0/1

```
# qtitinymix 'TX DEC0 MUX' 'MSM_DMIC'
# qtitinymix 'TX DMIC MUX0' 'DMIC0'
```

```
# qtitinymix 'TX DMIC MUX0' 'DMIC1'
# qtitinymix 'TX_AIF1_CAP Mixer DEC0' 1
# qtitinymix 'TX_DEC0 Volume' 84
```

3. Command to capture audio

```
# agmcap /data/tx.wav -D 100 -d 101 -c 1 -r 48000 -b 16 -
i CODEC_DMA-LPAIF_RXTX-TX-3
```

4. Press Ctrl + C to stop recording and save

**Note:** Captured Audio can be verified by copying it to Host PC or using Speaker playback.

### Speaker Playback

1. Command to start pulse audio

```
# systemctl start pulseaudio
```

2. Routing commands to enable speaker

```
# qtitinymix "WSA RX0 MUX" "AIF1_PB"
# qtitinymix "WSA RX1 MUX" "AIF1_PB"
# qtitinymix "WSA_RX0 INP0" "RX0"
# qtitinymix "WSA_RX1 INP0" "RX1"
# qtitinymix "WSA_COMP1 Switch" 1
# qtitinymix "WSA_COMP2 Switch" 1
# qtitinymix "SpkrLeft COMP Switch" 1
# qtitinymix "SpkrLeft VISENSE Switch" 1
# qtitinymix "SpkrLeft SWR DAC_Port Switch" 1
# qtitinymix "SpkrRight COMP Switch" 1
# qtitinymix "SpkrRight VISENSE Switch" 1
# qtitinymix "SpkrRight SWR DAC_Port Switch" 1
# qtitinymix 'WSA_RX0 Digital Volume' 100
# qtitinymix 'WSA_RX1 Digital Volume' 100
```

3. Command to play audio:

```
# paplay -v /data/tx.wav
```

CONFIDENTIAL

## 3.12. GNSS

**MRD8550 Platform Setup:**

- Connect GNSS antenna to the UFL connector slot i.e. **J11**.
- Open an adb shell and execute the following commands to verify GNSS.

```
# location_client_api_testapp -a -i 1000 -t 30 -r 1 -s g -l 1
```

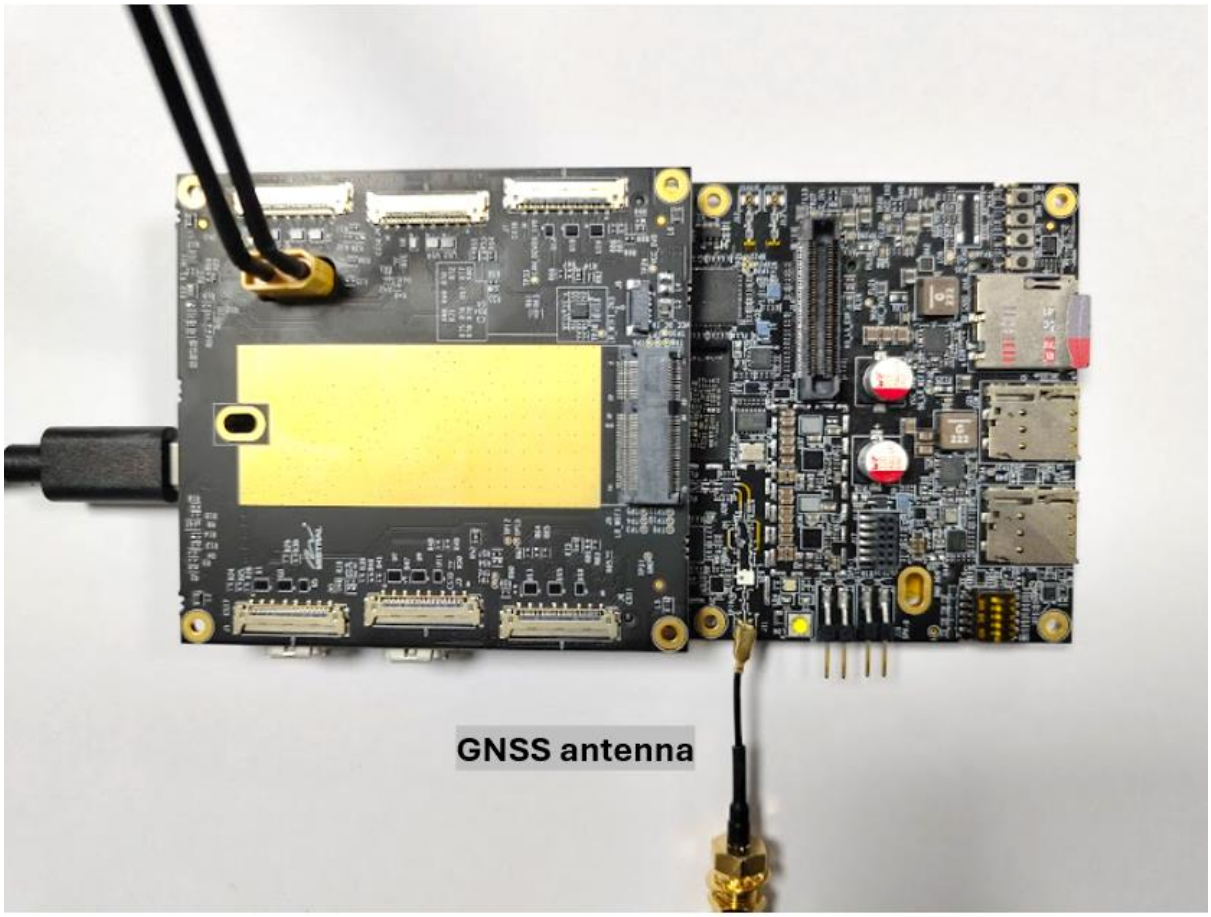


Figure 36: GNSS UFL Antenna Connector

**Note:** The GNSS module may fail to acquire a position fix, This is a known issue, and we are actively working on a resolution.

## 3.13. Type-C Redriver Display

**MRD8550 Platform Setup:**

- Connect the Type-C DP port to a display monitor via USB-C to HDMI converter.
- Wait for board to Boot and check the Weston display on connected the monitor.

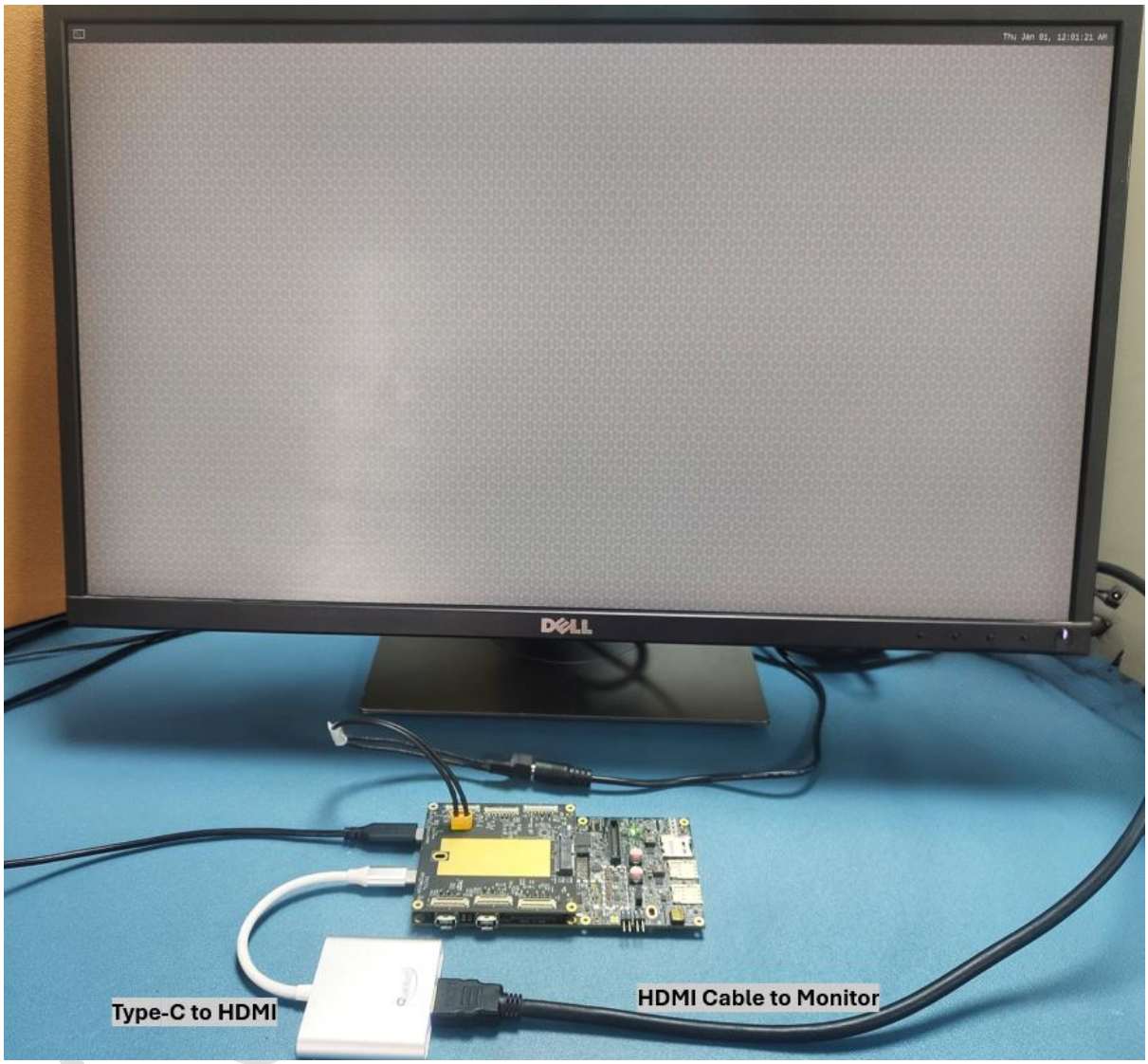


Figure 37: Merlin-Kit Type-C Display Port

## 3.14. HDMI-in Camera

**MRD8550 Platform Setup:**

- Connect the Type-C DP port to a display monitor via USB-C to HDMI converter
- Connect a HDMI camera to HDMI-Micro port
- Connect the debug port to Host PC via USB Micro-B Cable to access the shell.

Figure 38: Merlin-Kit Micro-HDMI Connector

Open a debug shell and execute the following command to preview the HDMI-In camera on Type-C Display.

```
# echo 487 > /sys/class/gpio/export
#echo out > /sys/class/gpio/gpio487/direction
#echo 1 > /sys/class/gpio/gpio487/value

#export XDG_RUNTIME_DIR=/run/user/root


# export WAYLAND_DISPLAY=wayland-1

#gst-launch-1.0 -e qtiqmmfsrc camera=0 name=camsrc ! video/x-raw,format=NV12,width=1920,height=1080,framerate=30/1 ! waylandsink fullscreen=true async=true sync=false
```

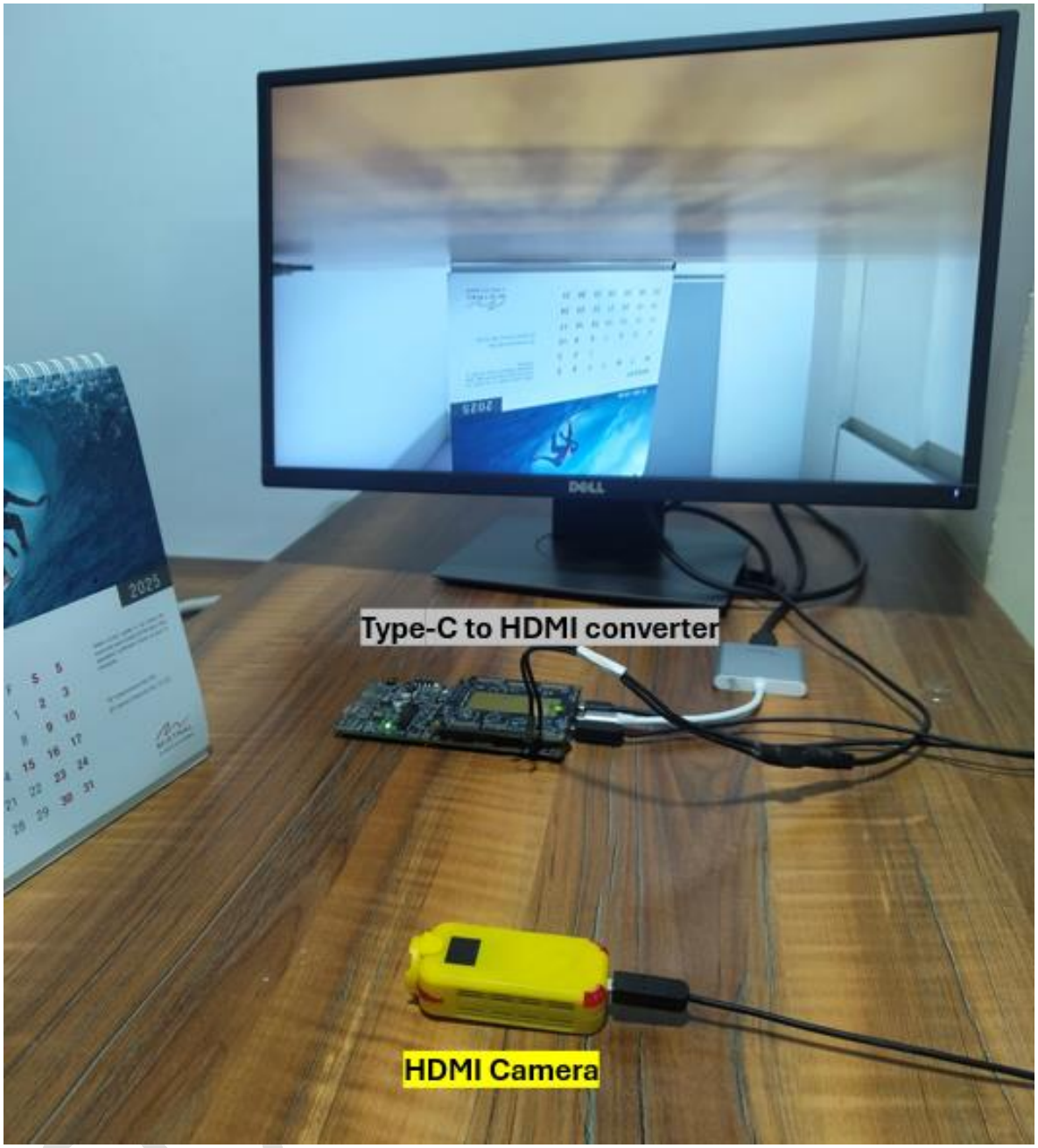


Figure 39: HDMI-IN Camera demo

## 3.15. MRD8550 Platform Camera Ports

**MRD8550 Platform Camera Ports:**

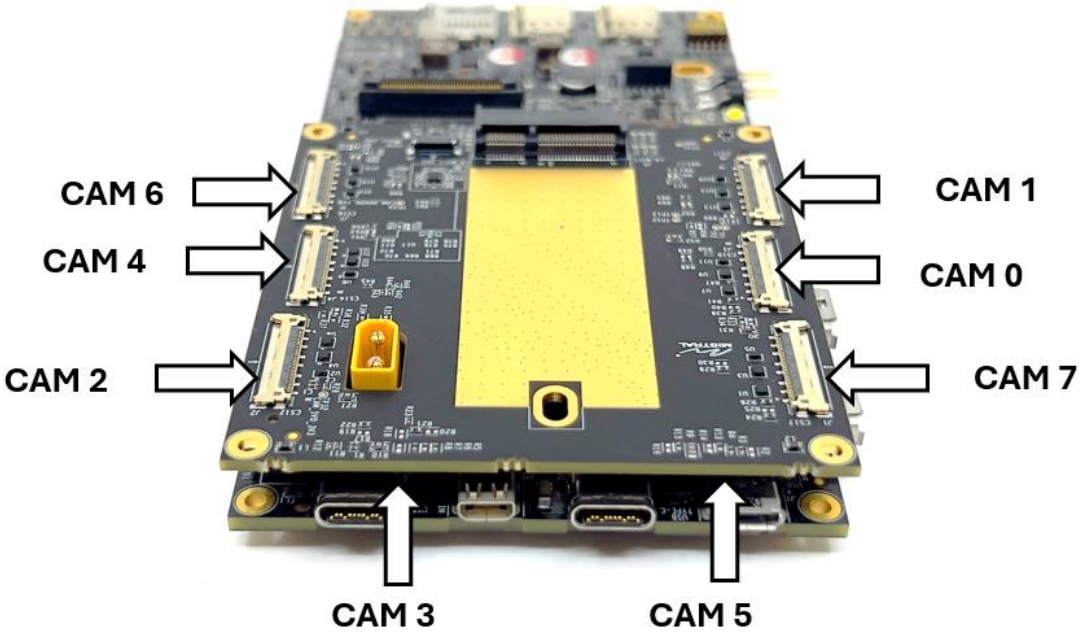


Figure 40: Merlin Kit Camera Ports

**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 186 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.

Presently, the camera module driver on the software side supports CSI0 to CSI7. Both IMX577 and OV9282 camera modules can be connected.

- To preview the video, connect the display port to any HDMI monitor via USB to HDMI converter as shown in section: 3.14
- Connect the debug port to Host PC via USB Micro-B Cable to access the shell.

**MRD8550 Platform Camera expansion board configuration**

The Camera expansion board enables the connection of various camera modules using an IPEX cable. Both IMX577 and OV9282 can be connected to any ports.

For the IMX577 to be connected, the gpio of the respective port should be pulled low, and the same gpio should be pulled high to connect to OV9282.

| CSI 0 | CSI 1 | CSI 2 | CSI 3 | CSI 4 | CSI 5 | CSI 6 | CSI 7 |
|---|---|---|---|---|---|---|---|
| GPIO_311 | GPIO_334 | GPIO_436 | GPIO_363 | GPIO_333 | GPIO_455 | GPIO_469 | GPIO_489 |

To access the gpio following command should be used.

```
# echo GPIO_NUM > /sys/class/gpio/export
# echo out > /sys/class/gpio/gpio_NUM/direction
```

To pull GPIO High (Select OV9282 Camera)

```
#echo 1 > /sys/class/gpio/gpio_NUM/value
```

To pull GPIO Low (Select IMX577 Camera)

```
#echo 0 > /sys/class/gpio/GPIO_NUM/value
```

**Note**: Replace the GPIO_NUM with respective gpio number.

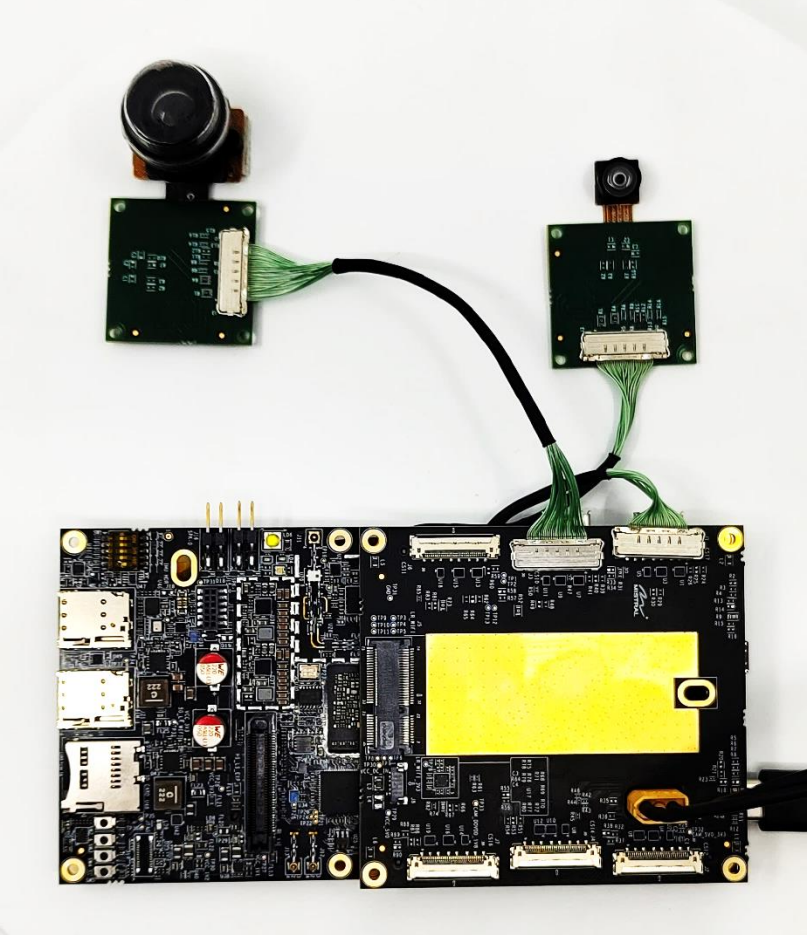

Figure 41: Camera Setup 1

1. CSI-7 port connected to IMX577 camera module.
2. CSI-0 port connected to OV9282 camera module.

**CSI-0 IMX577 Camera Preview:**

Open an adb shell and execute the following command to preview CSI-0 IMX577 camera on Type-c Display monitor.

```
# export XDG_RUNTIME_DIR=/run/user/root
# export WAYLAND_DISPLAY=wayland-1

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# 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 42: IMX Cam streamer**

**Figure 43: IMX Camera video streaming**

**CSI-1 OV9282 Camera Preview:**

Open an adb shell and execute the following command to preview CSI-1 OV9282 camera on Type-c Display monitor.

```
# export XDG_RUNTIME_DIR=/run/user/root
# export WAYLAND_DISPLAY=wayland-1
gst-launch-1.0 -e qtiqmmfsrc camera=0 name=camsrc ! video/x-raw,format=NV12,width=1280,height=720,framerate=30/1 ! waylandsink fullscreen=true async=true sync=false
```

```
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 44: OE camera video stream command**

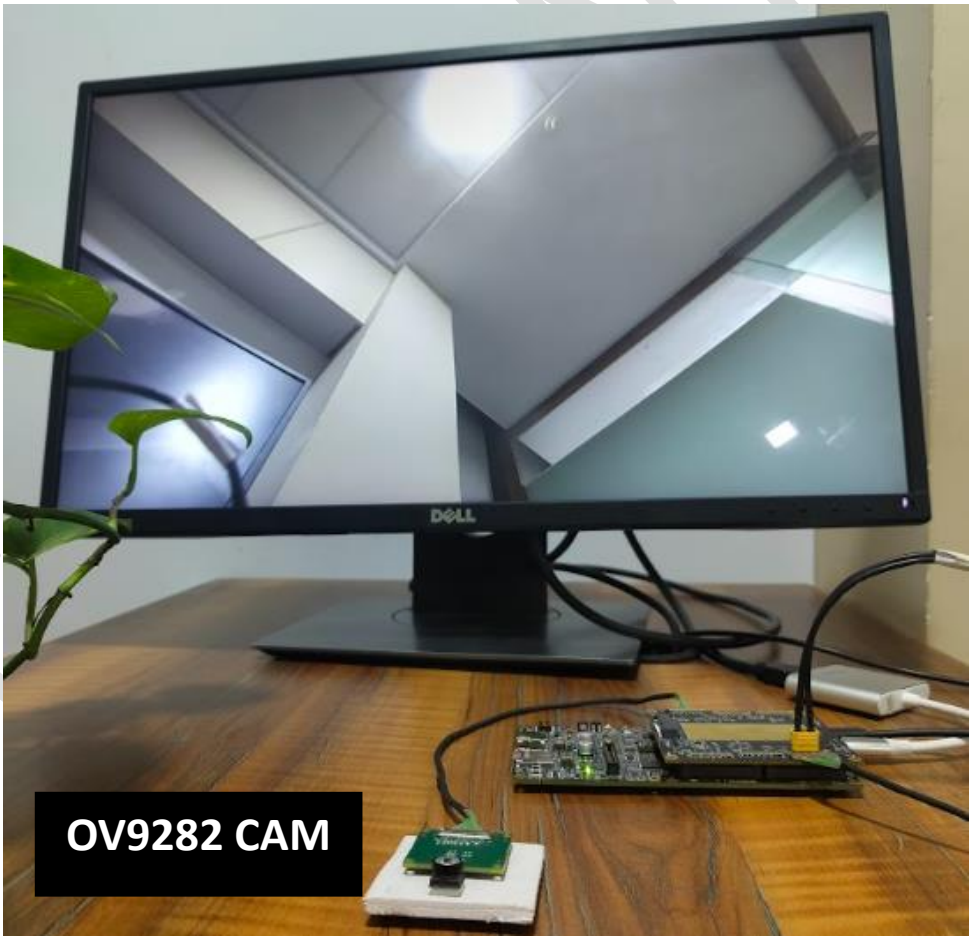


**Figure 45: OE Camera video streaming**

## 3.16. MRD8550 LR Wi-Fi (Station Mode)

**MRD8550 Platform Setup:**

Merlin-Kit Long Range Wi-Fi Setup:

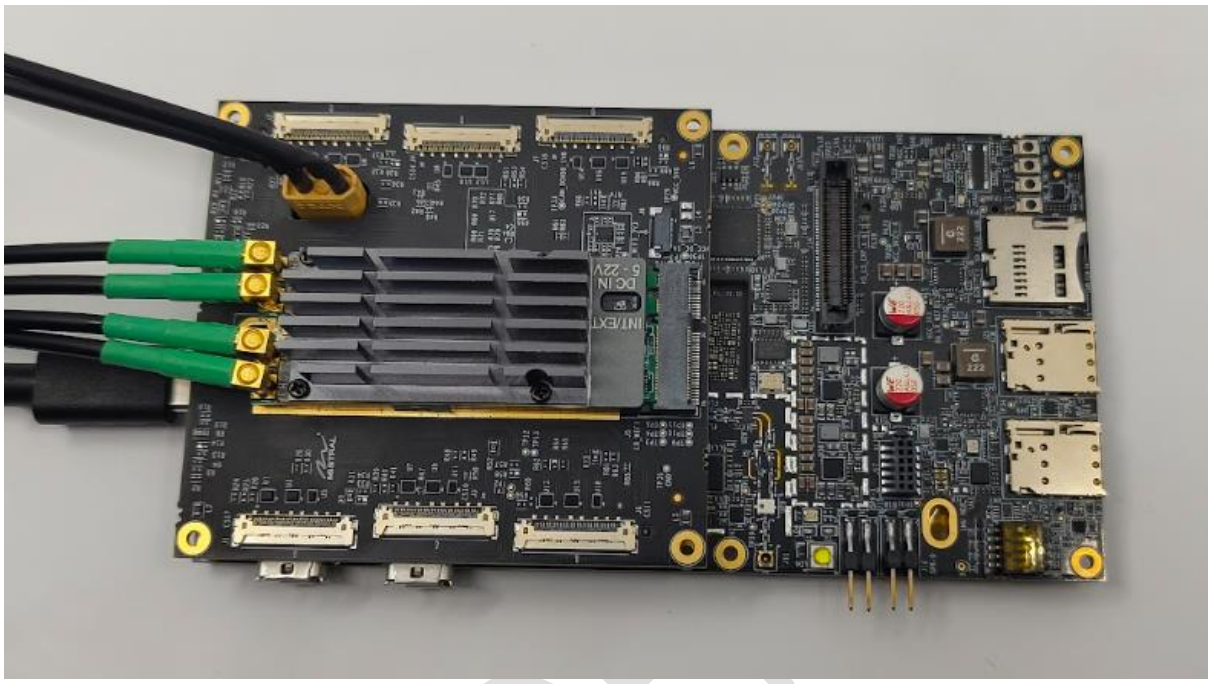

Figure 46: Merlin LR Wi-Fi Module

Open an adb shell and execute the following commands to verify the Long-Range Wi-Fi Station mode connection.

a. Enable the Interface

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

b. Run the following commands to connect to an AP network available

```
# add_network
```

c. Set SSID and PASSWORD of AP to be connected and enable the network

```
# set_network 1 ssid "User_ID"
# set_network 1 psk "User_Password"
# enable_network 1
```

d. Enable dhcpcd

```
# udhcpc pkill
# udhcpc -i ath1
```

e. Perform ping test

```
# ping google.co.in
```

```
# ping google.co.in

Pinging google.co.in [142.250.67.35] with 32 bytes of data:
Reply from 142.250.67.35: bytes=32 time=26ms TTL=115
Reply from 142.250.67.35: bytes=32 time=26ms TTL=115
Reply from 142.250.67.35: bytes=32 time=26ms TTL=115
Reply from 142.250.67.35: bytes=32 time=24ms TTL=115

Ping statistics for 142.250.67.35:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 24ms, Maximum = 26ms, Average = 25ms
```

Figure 47: LR WiFi ping test

## 3.17. MRD8550 LR Wi-Fi (Access point Mode)

**MRD8550 Platform Setup:**

Open an adb shell and execute the following commands to verify the Long-Range Wi-Fi Station mode connection.

a. Enable the Interface

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

b. Check the SSID and Password by giving below commands

```
# cat /data/misc/wifi/examples/hostapd_he20_ch36.conf |
grep ssid
```

```
# cat /data/misc/wifi/examples/hostapd_he20_ch36.conf |
grep wpa_passphrase
```

c. Connect the AP with your Host PC with the SSID and wpa_passphrase.

CONFIDENTIAL

## 3.18. EEPROM

**MRD8550 Platform Setup:**

Open an adb shell and execute the following commands to verify EEPROM.

a. Device detection

```
# ls /sys/class/i2c-dev/i2c-X/device/X-0050/
```

**Note:** Determine which I²C bus (X) has an EEPROM device detected

b. Dump the Hex values from EEPROM

```
# hexdump -C /sys/class/i2c-dev/i2c-X/device/X-0050/eeprom
```

c. Write command (Fill EEPROM with 0xFF)

```
# dd if=/dev/urandom of=/sys/class/i2c-dev/i2c-X/device/X-0050/eeprom bs=1 count=128 seek=0
```

```
~ # dd if=/dev/urandom of=/sys/class/i2c-dev/i2c-1/device/1-0050/eeprom bs=1 cou
nt=128 seek=0
128+0 records in
128+0 records out
128 bytes (128B) copied, 0.766559 seconds, 166B/s
~ # hexdump -C /sys/class/i2c-dev/i2c-1/device/1-0050/eeprom
00000000  c8 ab a5 93 53 84 ec cf  47 96 f5 44 85 d9 6f 69  |....S...G..D..oi|
00000010  0b ce ea 54 27 64 5f 24  2a 21 49 99 b9 61 af 0c  |...T'd_$*!I..a..|
00000020  23 75 e7 61 1d 85 be c7  bf 26 6e c7 6d 5b a0 19  |#u.a.....&n.m[..|
00000030  fe 4f 50 05 c7 57 33 b2  73 97 75 99 d8 69 e4 9f  |.OP..W3.s.u..i..|
00000040  d7 07 4f 4c e6 14 61 0a  74 10 92 9d ee aa d2 90  |..OL..a.t.......|
00000050  1c bd df bc 2c a3 3f f0  5d ed 9e 23 1b 0c df 7b  |....,.?.]..#...{|
00000060  ab b2 ee eb 5e 4d 02 85  68 d0 c0 31 31 52 90 7f  |....^M..h..11R..|
00000070  50 0c 50 61 6f 79 1e bc  ab 71 51 57 5c ea 06 6b  |P.Paoy...qQW\..k|
00000080
~ #
```

Figure 48: EEPROM hex dump command

CONFIDENTIAL

## 3.19. CAN

**MRD8550 Platform Setup:**

Open an adb shell and execute the following commands to verify CAN protocol.

a. Device detection:

```
# ifconfig -a | grep can0
```

b. Enabling device

```
# ip link set can0 down
# ip link set can0 up type can bitrate 1000000
```

c. Testing CAN device

```
# ip -d link show can0
```

d. Can send and dump commands

```
# cansend can0 001#1122334455667788 | candump can0
```

```
~ # ifconfig -a | grep can
can0: flags=128<NOARP>  mtu 72
~ # ip link set can0 down
~ # ip link set can0 up type can bitrate 1000000
~ # ip -d link show can0
14: can0: <NOARP,UP,LOWER_UP,ECHO> mtu 72 qdisc pfifo_fast state UP mode DEFAULT group default qlen 10
    link/can  promiscuity 0
    can <LOOPBACK,FD> state ERROR-ACTIVE (berr-counter tx 0 rx 0) restart-ms 0
          bitrate 1000000 sample-point 0.750
          tq 25 prop-seg 14 phase-seg1 15 phase-seg2 10 sjw 1
          mcp25xxfd: tseg1 2..256 tseg2 1..128 sjw 1..128 brp 1..256 brp-inc 1
          dbitrate 2000000 dsample-point 0.750
          dtq 25 dprop-seg 7 dphase-seg1 7 dphase-seg2 5 dsjw 1
          mcp25xxfd: dtseg1 1..32 dtseg2 1..16 dsjw 1..16 dbrp 1..256 dbrp-inc 1
          clock 40000000
~ # ip link set can0 down
~ # ip link set can0 type can bitrate 500000 dbitrate 2000000 fd on loopback on
~ # ip link set can0 up
~ # ip -d link show can0
14: can0: <NOARP,UP,LOWER_UP,ECHO> mtu 72 qdisc pfifo_fast state UP mode DEFAULT group default qlen 10
    link/can  promiscuity 0
    can <LOOPBACK,FD> state ERROR-ACTIVE (berr-counter tx 0 rx 0) restart-ms 0
          bitrate 500000 sample-point 0.875
          tq 25 prop-seg 34 phase-seg1 35 phase-seg2 10 sjw 1
          mcp25xxfd: tseg1 2..256 tseg2 1..128 sjw 1..128 brp 1..256 brp-inc 1
          dbitrate 2000000 dsample-point 0.750
          dtq 25 dprop-seg 7 dphase-seg1 7 dphase-seg2 5 dsjw 1
          mcp25xxfd: dtseg1 1..32 dtseg2 1..16 dsjw 1..16 dbrp 1..256 dbrp-inc 1
          clock 40000000
~ #
```

Figure 49: CAN Link show command

# 4. Additional Assistance

Please contact our support team for further assistance:

Phone India: +91-80-4562 1100

Email: info@mistralsolutions.com

Web: https://mistralsolutions.com/MRD8550-Merlin-Drone-Controller

**Note**: Information contained in this document is subject to change

CONFIDENTIAL