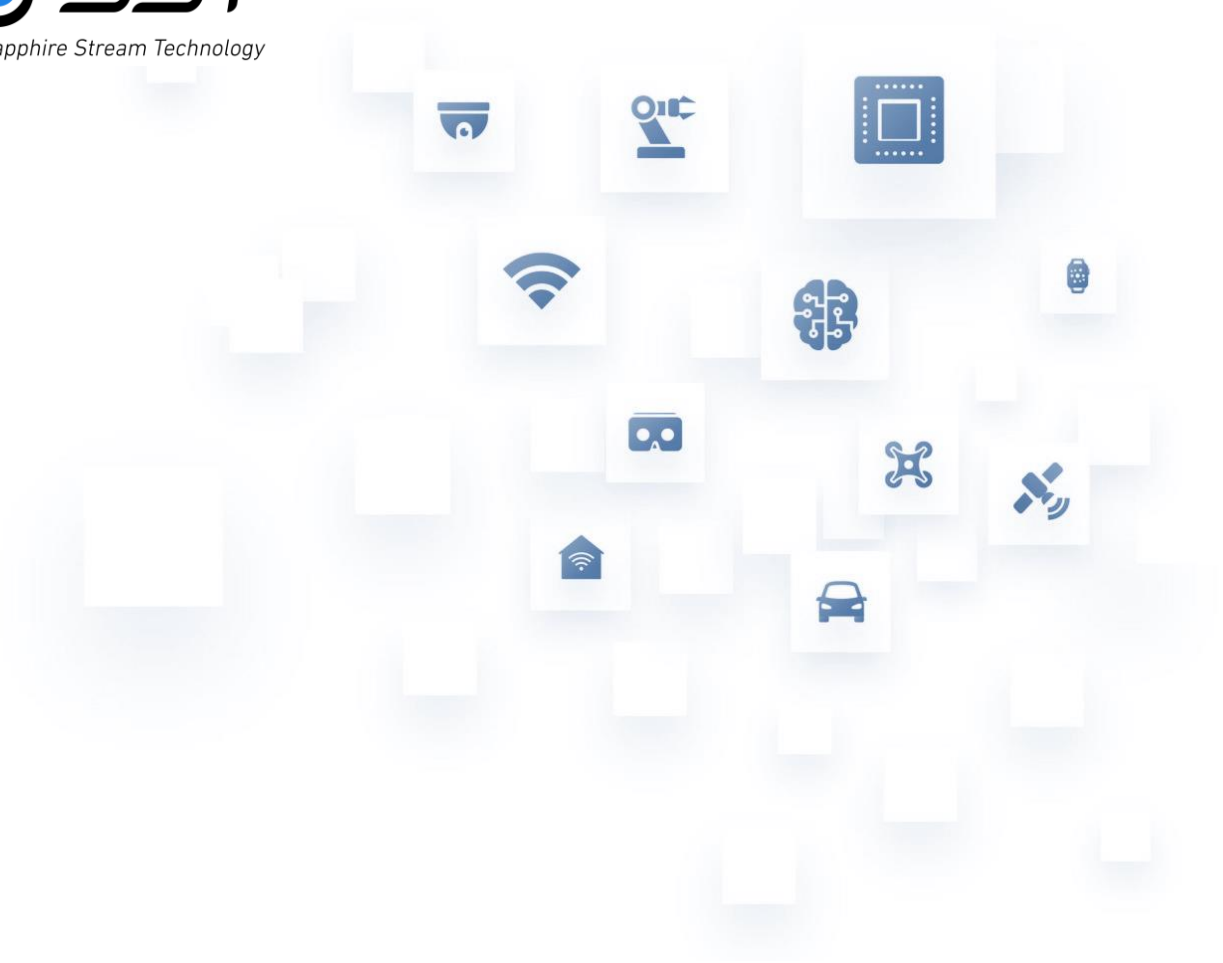
Sapphire Stream Technology



SST SQC6490 Development Kit

User Manual

Rev. V1.2.1
Oct 15, 2024

DN: sst-x3-2512.1

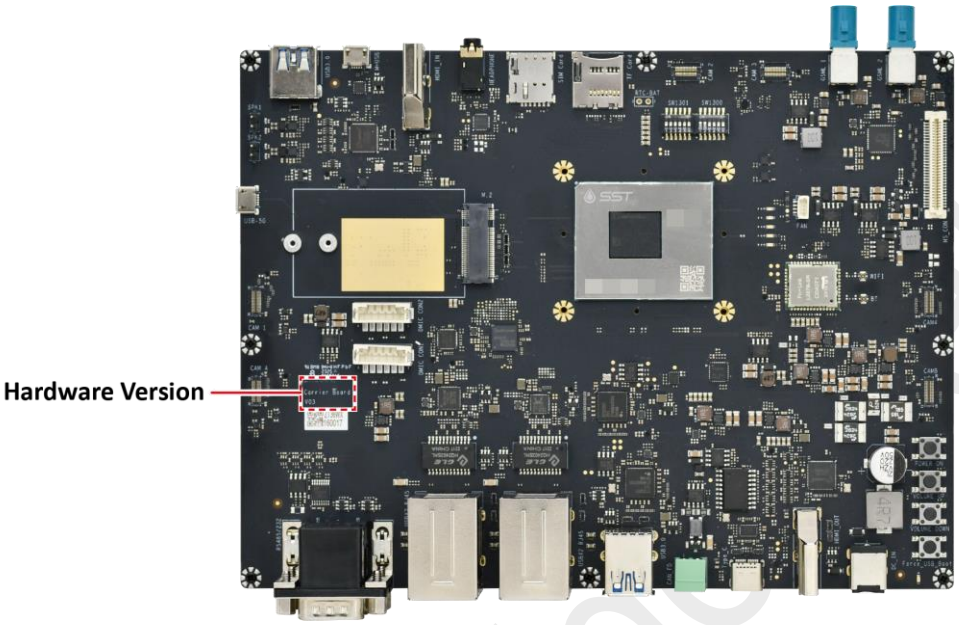
Revision History

Revision	Date	Author	Description
1.0	Oct 17, 2023	Xiaolong, Xie	Initial release.
1.1	Jan 16, 2024	Chenggang, Zhao	• Update boards relevant figures. • Update the entire document according to the latest software versions.
1.1.1	Mar 13, 2024	Chenggang, Zhao	Update Table 1-1.
1.2.2	Apr 12, 2024	Chenggang, Zhao	Add a note in 3.11.1.1. Hardware configuration.
1.2	Apr 26, 2024	Tongtong, Li	Add introduction to the function test methods on the LE baseline.
1.2.1	Oct 15, 2024	Tongtong, Li	• Remove the relevant information about Pin 2, Pin 3 and Pin 5 in Table 1-3. • Remove DK V03 relevant description for Pin 6 in Table 1-3. • Update the commands in 3.12.1.4.2. Camera preview and 3.12.1.4.3. Camera preview/recording/snapshot.

Table List

About This Document

- Illustrations in this documentation apply to **SQC6490 DK hardware V05 and later**. You can check the silk screen printed on the board to determine its hardware version.



- Depending on the model, some optional accessories, features, and software programs might not be available on your device.
- Depending on the version of operating systems and programs, some user interface instructions might not be applicable to your device.
- Documentation content is subject to change without notice. SST makes constant improvements to the documentation of the products, including this guidebook.
- Function declarations, function names, type declarations, attributes, and code samples appear in a different format, for example, `cp armcc armcpp`.
- Code variables appear in angle brackets, for example, `<number>`.
- Button, tool, and key names appear in bold font, for example, click **Save** or press **Enter**.
- Commands to be entered appear in a different font; on the host computer use $ as shell prompt, while on the target device use # as shell prompt, for example,

```
$ adb devices
# logcat
```

- Part of the code that does not contain instructions appear in a different format, for example,

```
SUBSYSTEM=="usb", ATTR{idVendor}=="18d1", MODE="0777", GROUP="adm"
```

- Folders and files are formatted in italic, for example, *sst_flash_flat.sh*.

Table of Contents

Chapter 1. Product Overview

SQC6490 development kit is a hardware and software development kit based on the SQC6490 smart module. It integrates a high-performance and power-efficient Qualcomm® 6nm QCS6490 SoC with strong computing power (12.5 TOPS). It has the ability of 4K@60FPS decode and 4K@30FPS encode. The SQC6490 DK provides a rich set of interfaces for customers to evaluate performance in different application scenarios, including 2 x 1000M RJ45 Ethernet ports, 7 x USB Type-A connectors, HDMI IN, HDMI OUT, 3 x MIPI camera connectors, and 2 x GMSL camera connectors. The SQC6490 SOM is an excellent platform for IoT products with high AI performance requirements, such as industrial handheld devices, industrial tablets, service robots, industrial robots, and edge computing.

The SQC6490 SOM is pin-to-pin compatible with SQC8550 SOM, sharing a universal IO board, which provides customers with assistance in prototype development and performance verification, reduces development difficulty, accelerates development cycles, and effectively saves the secondary development investment of products iteration.

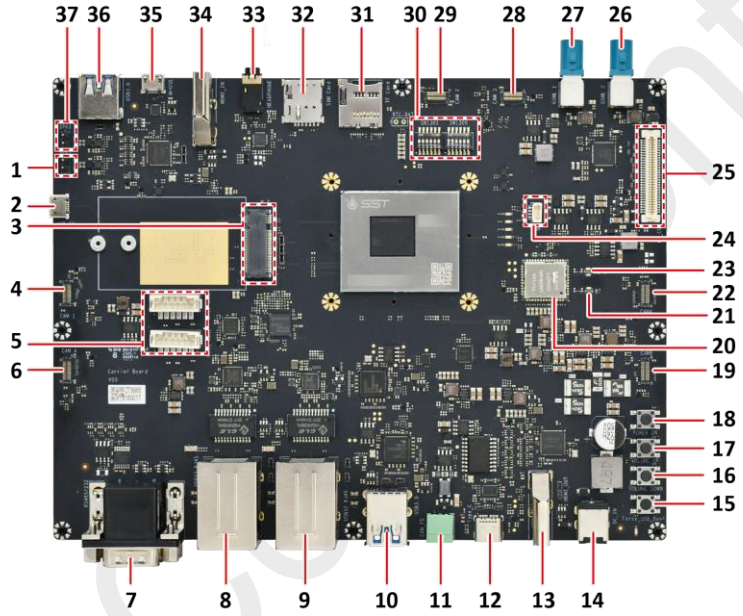


Figure 1-1. Board Overview

Table 1-1. Board Interface List

1. Speaker connector 2	20. Wi-Fi Dual-band 1x1 11a/b/g/n/ac+ Bluetooth 5.2 Combo Module
2. Micro USB connector for M.2 debug [a)]	21. BT antenna connector
3. M.2 connector B-KEY (5G module and SSD supported)	22. Camera module connector 4
4. Camera module connector 1	23. Wi-Fi antenna connector
5. DMIC connector x 2	24. Fan connector
6. Camera module connector A [a) b)]	25. High speed connector
7. DB9 connector x 2 (RS232 + RS485)	26. GMSL connector 2
8. 1 x RJ 45 + 2 x USB 3.0 Type-A connector	27. GMSL connector 1
9. 1 x RJ 45 + 2 x USB 3.0 Type-A connector	28. Camera module connector 3
10. 2 x USB 3.0 Type-A connector	29. NA
11. CAN connector	30. DIP switch x 2
12. USB 3.1 Type-C connector	31. NA
13. HDMI OUT connector	32. SIM card slot [a)]
14. DC in jack	33. Headphone connector
15. Force USB Boot button	34. HDMI IN connector
16. Volume down button	35. Micro USB connector for UART debug
17. Volume up button	36. USB 3.0 Type-A connector
18. Power on button	37. Speaker connector 1 [a) b)]
19. Camera module connector B [a) b)]	-

[a)] Temporarily unsupported on SQC6490.

[b)] These interfaces are left unpopulated on SQC6490 DK hardware V05.

1.1. Hardware features and specifications

Table 1-2. Hardware features and specifications

Category	Description
SOM on board	SQC6490 SOM
Display Interfaces	• 1 x HDMI OUT • 1 x DP V1.4 (Over Type-C)
Audio Interfaces	• 1 x Speaker OUT • 1 x HDMI IN with Audio (Over I2S interface) • 1 x HDMI OUT with Audio (Over I2S interface)
Camera Interfaces	• 3 x 4-lane D-PHY Camera • 2 x GMSL
General Interfaces on Mother Board	• 1 x USB 3.1 Type-C • 7 x USB 3.0 Type-A • 2 x Micro USB for debug (one for 5G/LTE module debug) • 1 x SDIO Wi-Fi 5 module • 2 x 1000M Ethernet Port • 1 x M.2 • 1 x HDMI OUT • 1 x HDMI IN • 2 x DMIC connector, support up to 6 DMICs • 2 x DB9 connectors (for RS232 and RS485 respectively) • 1 x CAN • 1 x HS connector • 1 x 3.5mm connector • 1 x Fan connector
Power Supply	12V DC in jack
Operating Environment	Operating Temperature: -35℃ ~ + 75℃
Dimensions	150mm x 200mm
OS Support	Android 13, Linux, LU

1.2. DIP switches

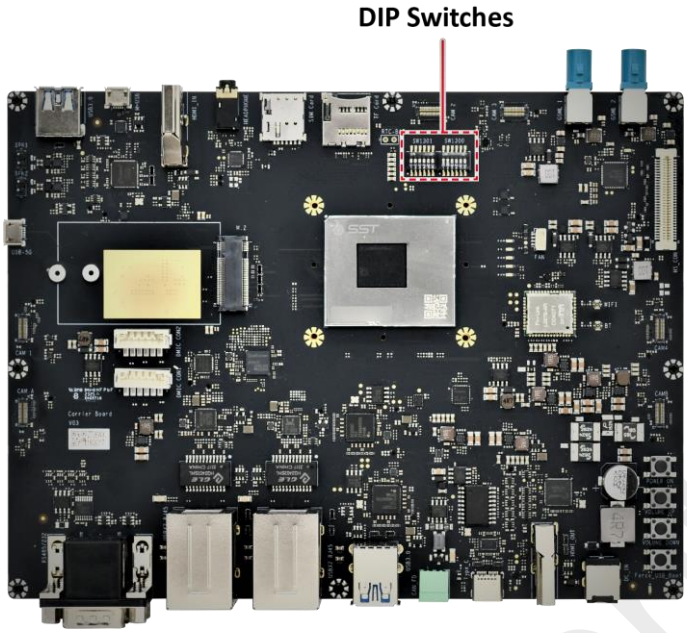


Figure 1-2. DIP Switches

Table 1-3. State description of SW1300

Pin #	DK HW Version	Name	Function
1	V03	SW_DSI0	Switch DSI0 to HDMI OUT or HS connector. • OFF: Switch DSI0 to the HS connector. • ON: Switch DSI0 to the HDMI OUT.
	V04 and later		
4	V03	SW_CSI3	Switch CSI3 to CAM1 or HDMI IN. • OFF: Switch CSI3 to HDMI IN. • ON: Switch CSI3 to CAM1.
	V04 and later		
6	V04	SW_I2S	Switch SPEAKER2_I2S speaker connector or M.2 connector. • OFF: Switch SPEAKER2_I2S to M.2 connector. • ON: Switch SPEAKER2_I2S to speaker connector.

➲ **NOTES:**

- To extend the display function via HS connectors, a customized adapter board is required. In case of any customization needs, please contact our FAE team for assistance.
- Set all pins of SW1301 to the "OFF" position.

Chapter 2. Flash Images

Refer to SQ*C6490 DK LA/LU/LE Software Release Notes*.

Chapter 3. Operation Procedures

This chapter introduces detailed operation procedures for various functions.

3.1. UART interfaces

3.1.1. UART debug port

➲ **NOTE:** This function has been verified in all software versions starting from the following:

- **LA3.0.ES.r000001**
- **LU1.0.FC.r001002**
- **SQC6490.LE1.0.ES.r000001**

Refer to the figure below to connect the device to PC via the **Micro USB connector for UART debug** (No.35 in Figure 1-1).

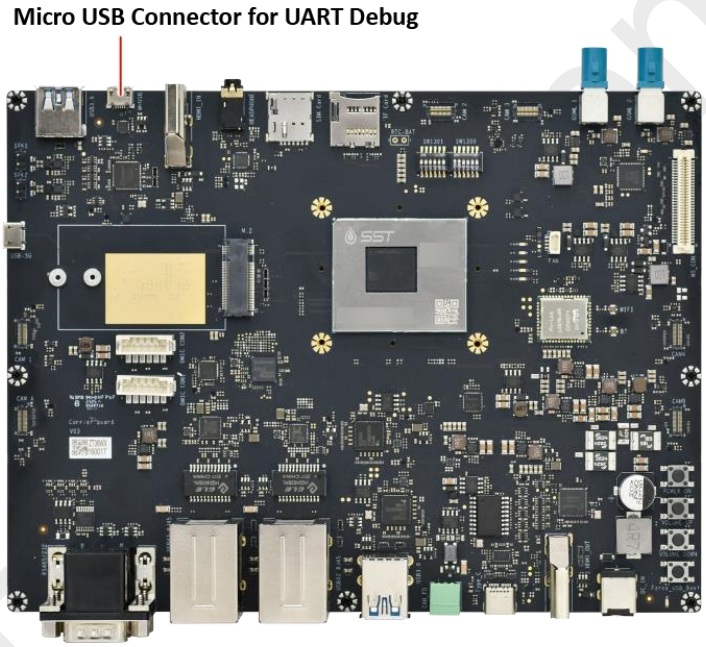


Figure 3-1. UART Debug Port

Step 1. Use a UART tool you prefer, e.g., **Minicom**.

A new serial port will be added to your PC after the device is connected. Select the new serial port, and configure it to `115200 8N1`. Refer to the figure below for the configuration of **Minicom 2.7**.

```
+-----------------------------------------------------------------------+
| A -    Serial Device      : /dev/ttyACM0                              |
| B - Lockfile Location     : /var/lock                                 |
| C -   Callin Program      :                                           |
| D -  Callout Program      :                                           |
| E -    Bps/Par/Bits       : 115200 8N1                                |
| F - Hardware Flow Control : No                                        |
| G - Software Flow Control : No                                        |
|                                                                       |
|    Change which setting?                                              |
+-----------------------------------------------------------------------+
```

Figure 3-2. Minicom 2.7 Configuration

Step 2. Press the **Power on button** (No.18 in Figure 1-1) to boot up the system and check the log output (example on LA).

```
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.1.0-00946.1-LAHAINA-1
S - IMAGE_VARIANT_STRING=SocKodiakLAA
S - OEM_IMAGE_VERSION_STRING=e0224778dd8a
S - Boot Interface: UFS
S - Secure Boot: Off
S - Boot Config @ 0x00786070 = 0x000000c1
S - JTAG ID @ 0x00786130 = 0x001980e1
S - OEM ID @ 0x00786138 = 0x00000000
S - Serial Number @ 0x00786134 = 0x7e81ad8b
S - OEM Config Row 0 @ 0x007841c0 = 0x0000000000000000
S - OEM Config Row 1 @ 0x007841c8 = 0x0000000000000000
S - Feature Config Row 0 @ 0x00784148 = 0x0000000000000000
S - Feature Config Row 1 @ 0x00784150 = 0x0000000000000000
S - Core 0 Frequency, 1516 MHz
S - PBL Patch Ver: 1
D -     6625 - pbl_apps_init_timestamp
D -    39236 - bootable_media_detect_timestamp
D -      946 - bl_elf_metadata_loading_timestamp
D -      704 - bl_hash_seg_auth_timestamp
D -     6800 - bl_elf_loadable_segment_loading_timestamp
D -     4553 - bl_elf_segs_hash_verify_timestamp
D -    17258 - bl_sec_hash_seg_auth_timestamp
D -      821 - bl_sec_segs_hash_verify_timestamp
D -       28 - pbl_populate_shared_data_and_exit_timestamp
... ...
... ...
```

3.1.2. DB9 connectors

3.1.2.1. Hardware configuration

Refer to the figure below to connect the board to PC via the **DB9 connectors** (No.7 in Figure 1-1).

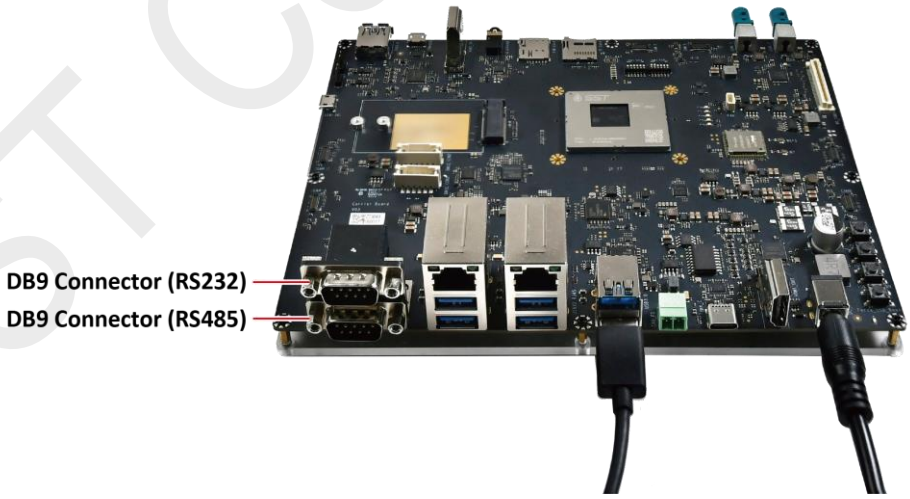


Figure 3-3. DB9 Connectors

3.1.2.2. Operation procedure on LA

➲ **NOTE:** This function has been verified in all software versions starting from **LA3.0.FC.r001002**.

Follow the steps below to test data transfer for RS232 and RS485 respectively.

- **RS232:**

  Step 1. Connect the **DB9 connector** (No.7 in Figure 1-1) on the board to the USB port of PC with a USB-to-RS232/RS485 converter and toggle the switch on the converter to "RS232".

  Step 2. Use a serial port tool on PC to connect the serial port of the board. Select the correct COM port and configure the baud rate to **115200**.

  Step 3. Enter the following commands to test UART by **Microcom**.

```
$ adb root
$ adb shell
lahaina:/ # echo 0 >  sys/devices/platform/soc/990000.qcom,qup_uart/loopback
lahaina:/ # microcom -s 115200 -X /dev/ttyHS1
```

  Step 4. Check the result of data transfer.

  - **Send mode:**

    Enter characters in the **Microcom** terminal and check if there is data in the serial port tool on PC.
  - **Receive mode:**

    Enter characters in the serial port tool on PC and check if the **Microcom** terminal on the device can receive data.

- **RS485:**

  Step 1. Connect the **DB9 connector** (No.7 in Figure 1-1) on the board to the USB port of PC with a USB-to-RS232/RS485 converter and toggle the switch on the converter to "RS485".

  Step 2. Use a serial port tool on PC to connect the serial port on Board. Select the correct COM port and configure the baud rate to **115200**.

  Step 3. Set RS485 to the send/receive mode with the following commands as required.

  - **Send mode:**

```
$ adb root
$ adb shell
lahaina:/ # cd /sys/class/gpio
lahaina:/ # echo 393 > export
lahaina:/ # cd gpio393/
lahaina:/ # echo out> direction
lahaina:/ # echo 0 > value
```

  - **Receive mode:**

```
$ adb root
$ adb shell
lahaina:/ # cd /sys/class/gpio
lahaina:/ # echo 393 > export
lahaina:/ # cd gpio393/
lahaina:/ # echo out> direction
lahaina:/ # echo 1 > value
```

  Step 4. Enter the following commands to test UART by **Microcom**.

```
$ adb root
$ adb shell
lahaina:/ # echo 0 >  sys/devices/platform/soc/a98000.qcom,qup_uart/loopback
lahaina:/ # microcom -s 115200 -X /dev/ttyHS2
```

Step 5. Check the result of data transfer.

- **Send mode:**

  Enter characters in the **Microcom** terminal and check if there is data in the serial port tool on PC.

- **Receive mode:**

  Enter characters in the serial port tool on PC and check if the **Microcom** terminal on the device can receive data.

3.1.2.3. Operation procedure on LU

➲ **NOTE:** This function has been verified in all software versions starting from **LU1.0.FC.r001002**.

- **RS232 – Send mode:**

Step 1. Connect the **DB9 connector** (No.7 in Figure 1-1) to the USB port on the board with a USB-to-RS232/RS485 converter and toggle the switch on the converter to "RS232".

Step 2. Open two terminals and enter the following commands to verify RS232 sending.

1) Terminal 1:

```
$ adb root
$ adb shell
/ # stty -F /dev/ttyHS1 speed 115200
/ # stty -F /dev/ttyUSB0 speed 115200
```

2) Terminal 2:

```
$ adb root
$ adb shell
/ # cat /dev/ttyUSB0
```

3) Terminal 1:

```
/ # echo rs232send > /dev/ttyHS1
```

Step 3. Observe RS232 sending via the output of at Terminal 2.

```
/ # cat /dev/ttyUSB0
rs232send
```

- **RS232 – Receive mode:**

Step 1. Connect the **DB9 connector** (No.7 in Figure 1-1) to the USB port on the board with a USB-to-RS232/RS485 converter and toggle the switch on the converter to "RS232".

Step 2. Open two terminals and enter the following commands to verify RS232 receiving.

1) Terminal 1:

```
$ adb root
$ adb shell
/ # stty -F /dev/ttyHS1 speed 115200
/ # stty -F /dev/ttyUSB0 speed 115200
/ # cat /dev/ttyHS1
```

2) Terminal 2:

```
$ adb root
$ adb shell
/ # echo rs232receive > /dev/ttyUSB0
```

Step 3. Observe RS232 receiving via the output of Terminal 1.

```
/ # cat /dev/ttyHS1
rs232receive
```

- **RS485 - Send mode:**

  Step 1. Connect the **DB9 connector** (No.7 in Figure 1-1) to the USB port on the board with a USB-to-RS232/RS485 converter and toggle the switch on the converter to “RS485”.

  Step 2. Open 2 terminals and enter the following commands to verify RS485 sending.

  1) Terminal 1:

```
$ adb root
$ adb shell
/ # stty -F /dev/ttyHS2 speed 115200
/ # stty -F /dev/ttyUSB0 speed 115200
/ # cd /sys/class/gpio
/ # echo 393 > export
/ # cd gpio393/
/ # echo out> direction
/ # echo 0 > value
/ # cat /dev/ttyUSB0
```

  2) Terminal 2:

```
$ adb root
$ adb shell
/ # echo rs485send > /dev/ttyHS2
```

  Step 3. Observe RS485 sending via the output of Terminal 1.

```
/ # cat /dev/ttyUSB0
rs485send
```

- **RS485 – Receive mode:**

  Step 1. Connect the **DB9 connector** (No.7 in Figure 1-1) to the USB port on the board with a USB-to-RS232/RS485 converter and toggle the switch on the converter to “RS485”.

  Step 2. Open two terminals and enter the following commands to verify RS485 receiving.

  1) Terminal 1:

```
$ adb root
$ adb shell
/ # stty -F /dev/ttyHS2 speed 115200
/ # stty -F /dev/ttyUSB0 speed 115200
/ # cd /sys/class/gpio
/ # echo 393 > export
/ # cd gpio393/
/ # echo out> direction
/ # echo 1 > value
/ # cat /dev/ttyHS2
```

  2) Terminal 2:

```
$ adb root
$ adb shell
/ # echo rs485receive > /dev/ttyUSB0
```

  Step 3. Observe RS485 receiving via the output of Terminal 1.

```
/ # cat /dev/ttyHS2
rs485receive
```

3.1.2.4. Operation procedure on LE

This function is currently unavailable in the latest SDK version SQC6490.LE1.0.ES.r000001.

3.2. Buttons

Refer to Figure 3-4 to locate the buttons (No.15, No.16, No.17 and No.18 in Figure 1-1).

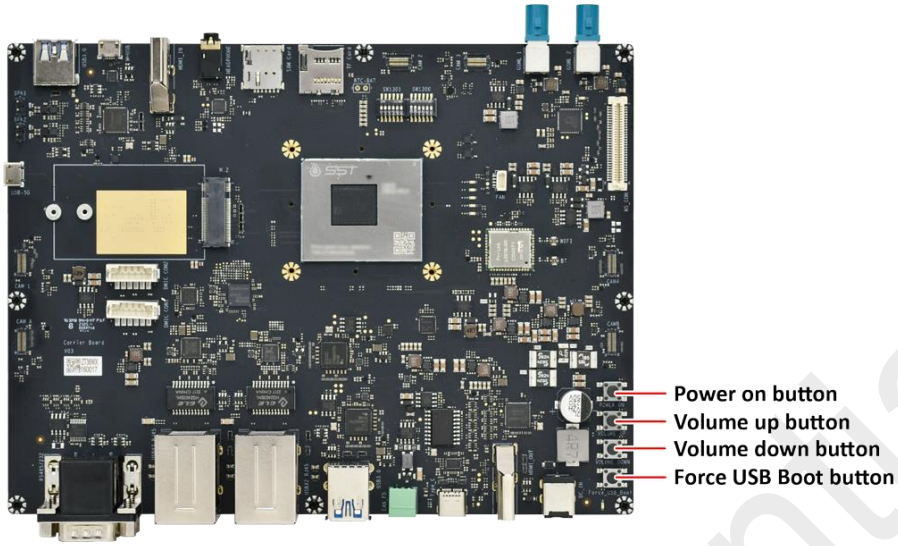


Figure 3-4. Buttons

Table 3-1. Button specifications

Button	Device Node
Power on	/dev/input/event0
Volume up	/dev/input/event1
Volume down	/dev/input/event0
Force USB boot	-

➲ **NOTE**: The **Force USB boot button** is used to force the system to enter emergency download mode.

3.2.1. Operation procedure on LA

➲ **NOTE:** This function has been verified in all software versions starting from **LA3.0.ES.r000001**.

Execute the following commands to test the functionality of the buttons:

```
$ adb root
$ adb shell
lahaina:/ # getevent -l
add device 1: /dev/input/event2
 name: "lahaina-yupikidp-snd-card Headset"
add device 2: /dev/input/event1
 name: "gpio-keys"
add device 3: /dev/input/event0
 name: "qpnp_pon"
/dev/input/event0: EV_KEY    KEY_POWER      DOWN         //press power on button
/dev/input/event0: EV_SYN    SYN_REPORT     00000000
/dev/input/event0: EV_KEY    KEY_POWER      UP           //release power on button
/dev/input/event0: EV_SYN    SYN_REPORT     00000000

/dev/input/event1: EV_KEY    KEY_VOLUMEUP    DOWN        //press volume up button
/dev/input/event1: EV_SYN    SYN_REPORT      00000000
/dev/input/event1: EV_KEY    KEY_VOLUMEUP    UP          //release volume up button
/dev/input/event1: EV_SYN    SYN_REPORT      00000000

/dev/input/event0: EV_KEY    KEY_VOLUMEDOWN   DOWN       //press volume down button
/dev/input/event0: EV_SYN    SYN_REPORT       00000000
/dev/input/event0: EV_KEY    KEY_VOLUMEDOWN   UP         //release volume down button
/dev/input/event0: EV_SYN    SYN_REPORT       00000000
```

3.2.2. Operation procedure on LU

➲ **NOTE:** This function has been verified in all software versions starting from **LU1.0.FC.r001002**.

Refer to the following steps to test the functionality of each button.

- **Power on and Volume down:**

  Execute the following commands and press the **Power on** button or **Volume down** button, then some characters will be output.

```
$ adb root
$ adb shell
# cat /dev/input/event0
▒▒!d<▒t▒▒!d<▒
```

- **Volume up:**

  Execute the following commands and press the **Volume up** button, then some characters will be output.

```
# cat /dev/input/event1
▒▒!d%s▒▒!d%▒▒!d▒▒s▒▒!d▒▒
```

3.2.3. Operation procedure on LE

➲ **NOTE:** This function has been verified in all software versions starting from **SQC6490.LE1.0.ES.r000001**.

Refer to the following steps to test the functionality of each button.

- **Power on:**

  Execute the following commands and press the **Power on** button, then some characters will be output.

```
$ adb root
$ adb shell
# cat /dev/input/event1
▒▒!d<▒t▒▒!d<▒
```

- **Volume up:**

  Execute the following commands and press the **Volume up** button, then some characters will be output.

```
# cat /dev/input/event0
▒▒!d%s▒▒!d%▒▒!d▒▒s▒▒!d▒▒
```

- **Volume down:**

  Execute the following commands and press the **volume down** button, then some characters will be output.

```
# cat /dev/input/event2
▒▒!d%s▒▒!d%▒▒!d▒▒s▒▒!d▒▒
```

3.3. LEDS

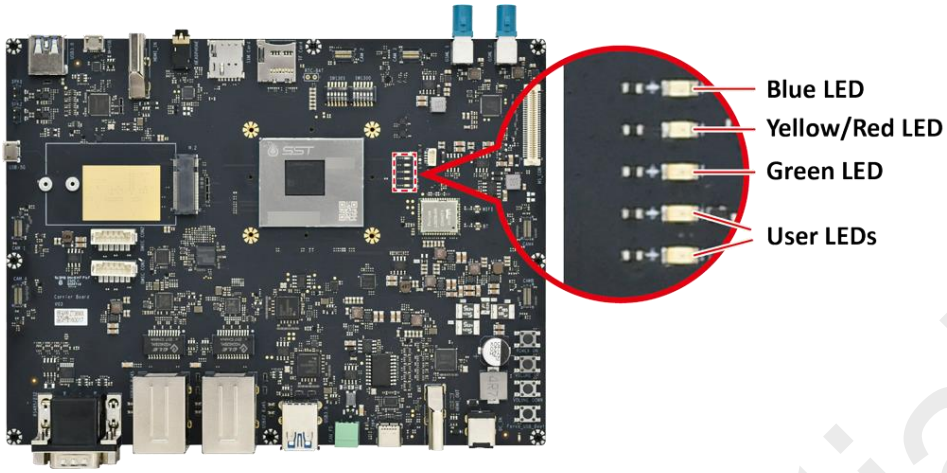


Figure 3-5. LEDs

➲ **NOTE:** This function has been verified in all software versions starting from the following:

- **LA3.0.ES.r000001**
- **LU1.0.FC.r001002**
- **SQC6490.LE1.0.ES.r000001**

Run the following commands to test LEDs.

```
$ adb root
$ adb shell
// Test RGB LEDs
# echo 255 > /sys/class/leds/blue/brightness       // LA3.0/LU1.0/LE1.0 turn on blue led
# echo 0 > /sys/class/leds/blue/brightness         // LA3.0/LU1.0/LE1.0 turn off blue led
# echo 255 > /sys/class/leds/yellow/brightness     // LA3.0 turn on yellow led
# echo 0 > /sys/class/leds/yellow/brightness       // LA3.0 turn off yellow led
# echo 255 > /sys/class/leds/red/brightness        // LU1.0/LE1.0 turn on red led
# echo 0 > /sys/class/leds/red/brightness          // LU1.0/LE1.0 turn off red led
# echo 255 > /sys/class/leds/green/brightness      // LA3.0/LU1.0/LE1.0 turn on green led
# echo 0 > /sys/class/leds/green/brightness        // LA3.0/LU1.0/LE1.0 turn off green led

// Test user LEDs
# echo 1 > /sys/class/leds/green_led1/brightness   // LA3.0/LU1.0 turn on green_led1 led
# echo 0 > /sys/class/leds/green_led1/brightness   // LA3.0/LU1.0 turn off green_led1 led
# echo 1 > /sys/class/leds/green_led2/brightness   // LA3.0/LU1.0 turn on green_led2 led
# echo 0 > /sys/class/leds/green_led2/brightness   // LA3.0/LU1.0 turn off green_led2 led
```

3.4. Fan

3.4.1. Hardware configuration

Refer to the figure below to connect a fan to the board via the **Fan connector** (No.24 in Figure 1-1).

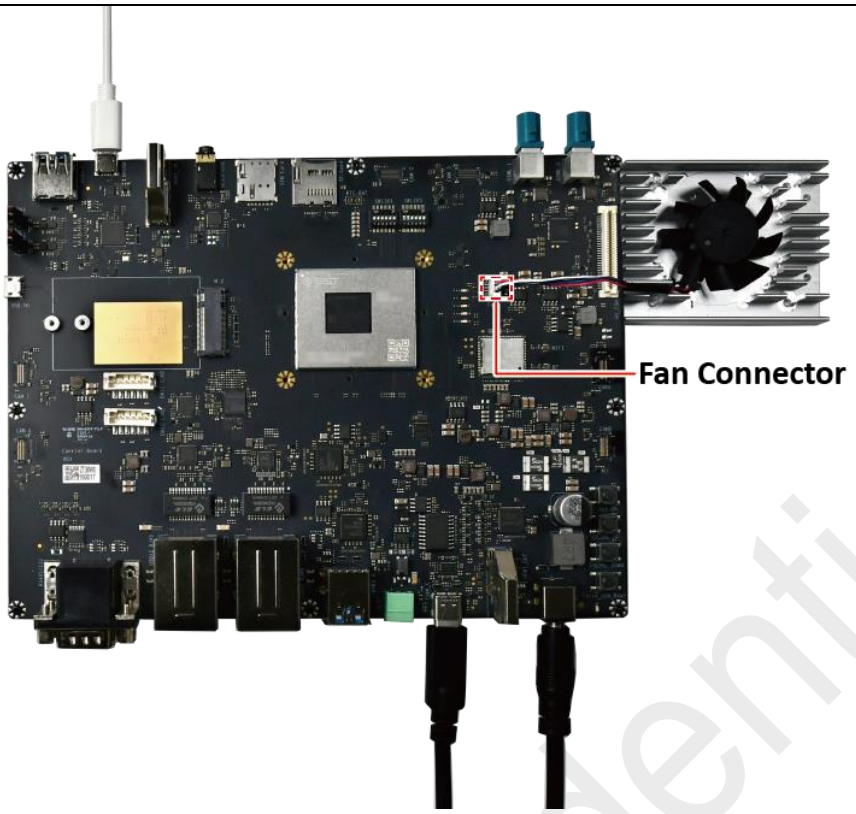


Figure 3-6. Fan Connector

Table 3-2. Pin function

No.	Function
1	PWM
2	NA
3	FAN_VCC
4	GND

3.4.2. Operation procedure on LA and LU

➲ **NOTES**:

- This function has been verified in all software versions starting from the following:
  - **LA3.0.ES.r000001**
  - **LU1.0.FC.r001002**
- Upon system boot-up, the fan will rotate by default.

Enter the following commands to control the fan:

```
$ adb root
$ adb shell
# echo 0 > /sys/class/hwmon/hwmon0/pwm1
# echo 255 > /sys/class/hwmon/hwmon0/pwm1
```

3.4.3. Operation procedure on LE

This function is currently unavailable in the latest SDK version **SQC6490.LE1.0.ES.r000001**.

3.5. Display interfaces

The device supports a DP over Type-C interface and a HDMI OUT interface.

3.5.1. DP over Type-C interface

3.5.1.1. Hardware configuration

Connect a monitor to the board with a Type-C-to-DP converter via the **USB 3.1 Type-C connector** (No. 12 in Figure 1-1).

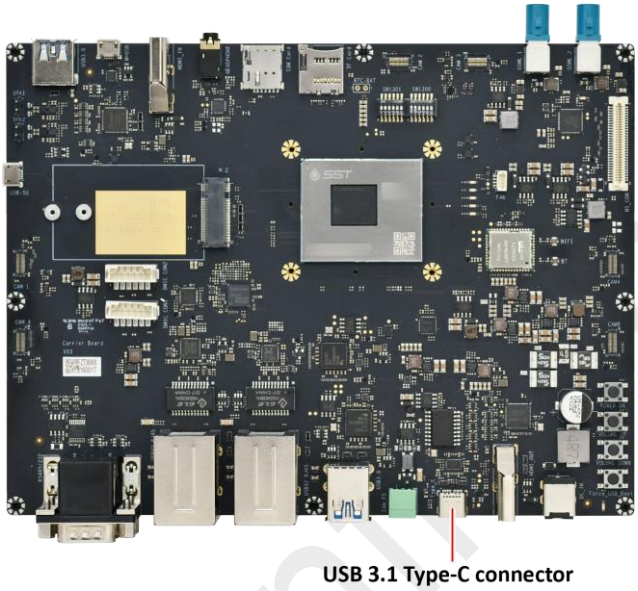


Figure 3-7. Locate USB 3.1 Type-C Connector

3.5.1.2. Operation procedure on LA

➲ **NOTE:** This function has been verified in all software versions starting from **LA3.0.FC.r001002**.

Step 1. Connect the power supply to the DK and press the **Power on button** (No. 18 in Figure 1-1) to boot up the system.

Step 2. The following screen (example on LA) will appear immediately on the monitor.

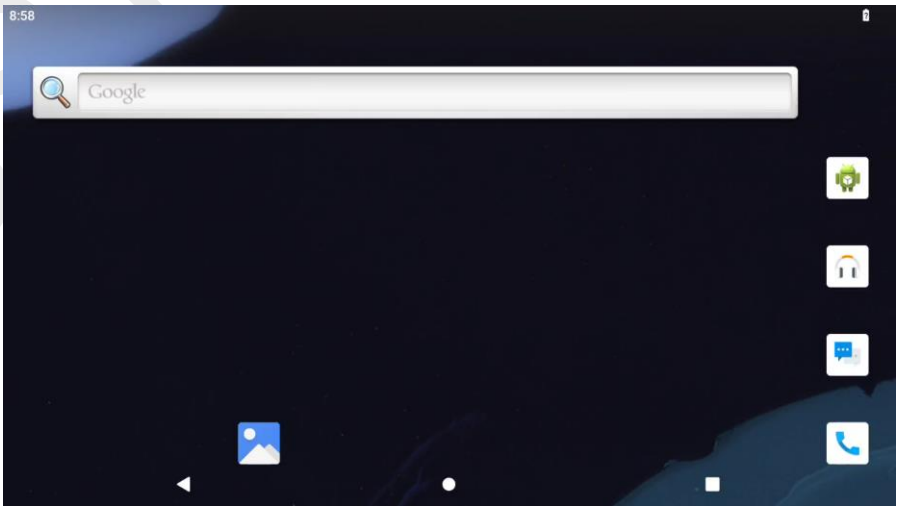


Figure 3-8. Monitor Screen (Example on LA)

3.5.1.3. Operation procedure on LU

➲ **NOTE:** This function is currently unavailable in the latest SDK version **LU1.0.FC.r001002**.

3.5.1.4. Operation procedure on LE

➲ **NOTE:** This function has been verified in all software versions starting from **SQC6490.LE1.0.ES.r000001**.

Step 1. Connect the power supply to the DK and press the **Power on button** (No. 18 in Figure 1-1) to boot up the system.

Step 2. The following screen (example on LE) will appear immediately on the monitor.

➲

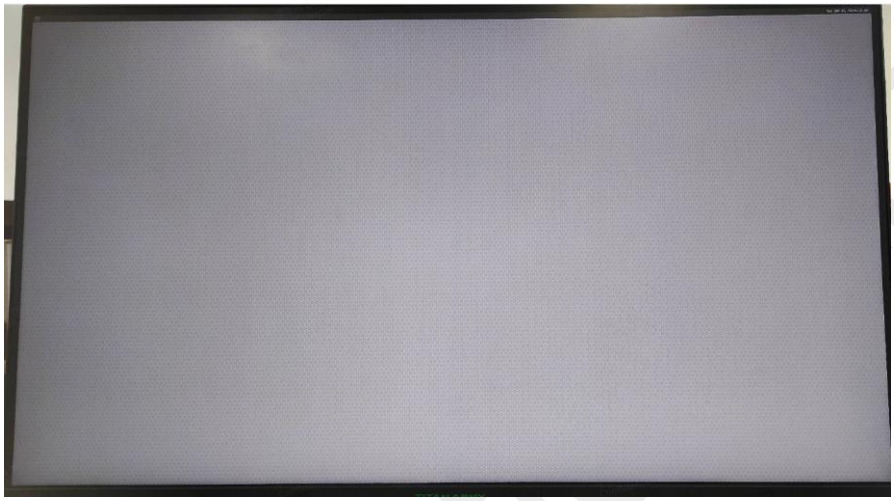


Figure 3-9. Monitor Screen (Example on LE)

3.5.2. HDMI OUT

3.5.2.1. Hardware configuration

Step 1. Set the DIP switch. Refer to Table 1-3.

- SW1300: Toggle 1 and 2 to ON.

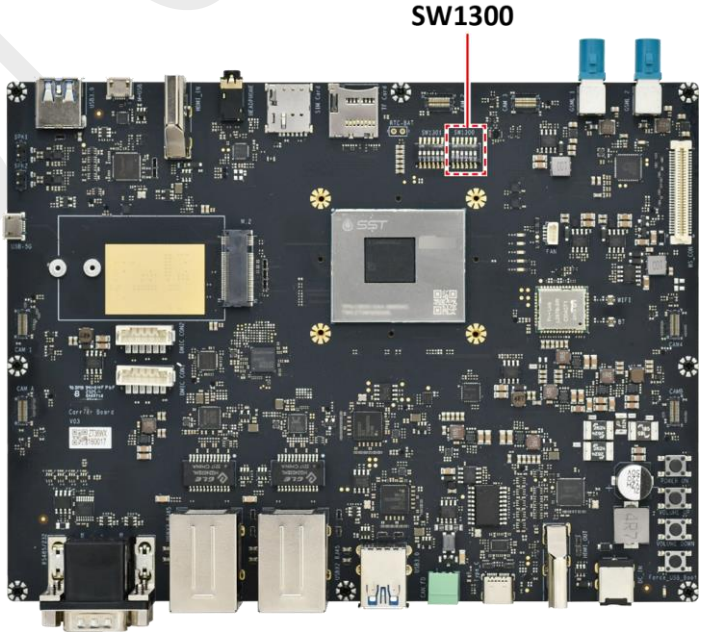


Figure 3-10. Set the DIP Switch

Step 2. Connect a monitor to the board via the **HDMI OUT connector** (No. 13 in Figure 1-1) with an HDMI cable.

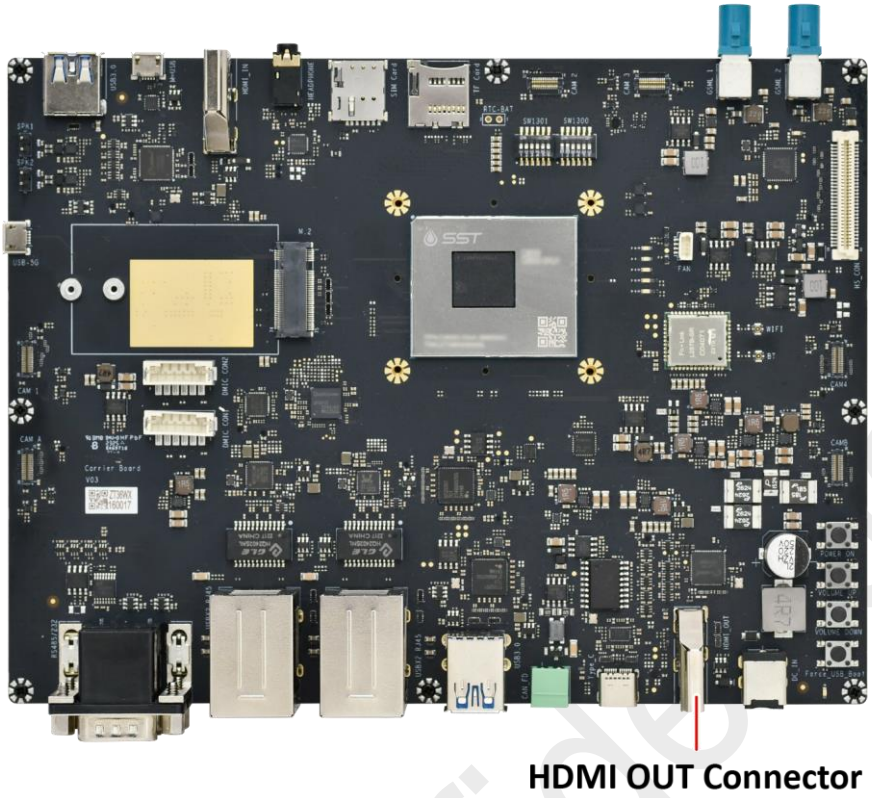


Figure 3-11. HDMI OUT Connector

3.5.2.2. Operation procedure on LA

➲ **NOTES**:

- This function has been verified in all software versions starting from **LA3.0.ES.r000001**.
- Supported resolution for HDMI in the current version: 1080P@60FPS.

Connect the power supply to the DK board and press the **Power on button** (No. 18 in Figure 1-1) to boot up the system. The following screen (example on LA) will appear immediately.

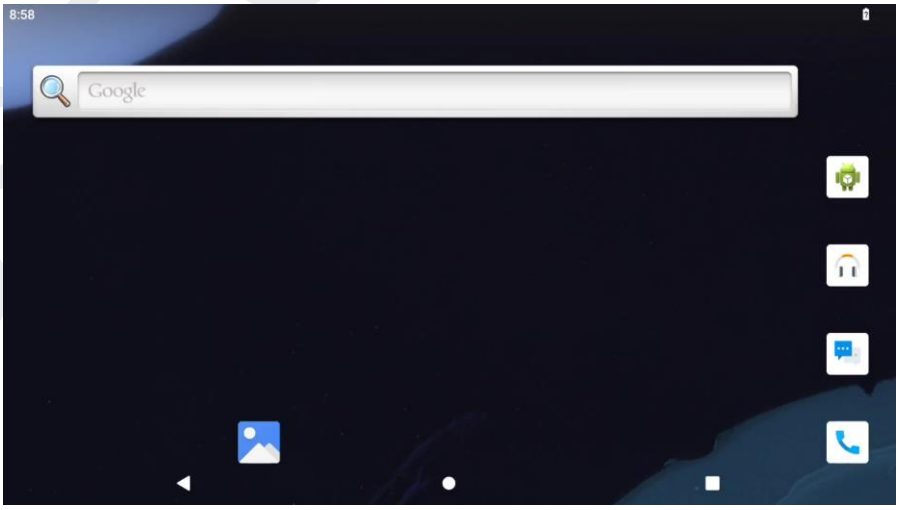


Figure 3-12. Display (Example on LA)

3.5.2.3. Operation procedure on LU and LE

➲ **NOTES**:

- This function has been verified in all software versions starting from the following:
  - **LU1.0.FC.r001002**
  - **SQC6490.LE1.0.ES.r000001**

Press the **Power on button** (No. 18 in Figure 1-1) to boot up the board. Then the Weston screen will appear on the monitor automatically.

Figure 3-13. Weston Screen

3.6. USB Type-C connector

3.6.1. Hardware configuration

Refer to the figure below to connect a USB flash disk, UVC, UAC, or USB Ethernet device to the board via **USB 3.1 Type C connector** (No.12 in Figure 1-1) with a Type-C USB OTG cable.

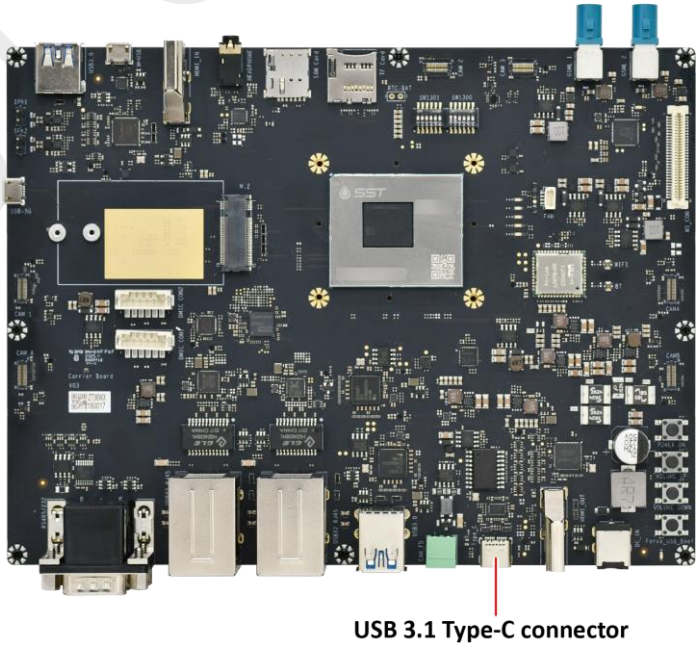


Figure 3-14. Locate USB 3.1 Type-C Connector

3.6.2. Operation procedure on LA

➲ **NOTE**: This function has been verified in all software versions starting from **LA3.0.ES.r000001**.

Step 1. The board shall detect an external hub once connected by a Type-C USB OTG cable. Input `dmesg | grep usb` for detailed information.

```
# dmesg | grep usb
[ 1541.985025] msm-dwc3 a600000.ssusb: could not transition HS PHY to L2
[ 1541.994981] msm-dwc3 a600000.ssusb: DWC3 in low power mode
[ 1541.995046] msm-dwc3 a600000.ssusb: Could not get usb psy
[ 1542.106137] msm-dwc3 8c00000.hsusb: DWC3 exited from low power mode
[ 1542.230006] msm-dwc3 8c00000.hsusb: DWC3 in low power mode
[ 1542.418317] ssusb redriver 6-001c: writing reg 0x01 failure
[ 1542.418326] ssusb redriver 6-001c: channel parameters update failure(-13).
[ 1542.462646] msm-dwc3 8c00000.hsusb: DWC3 exited from low power mode
[ 1542.856864] msm-usb-ssphy-qmp 88e8000.ssphy: USB DP QMP PHY: Update TYPEC CTRL(3)
[ 1542.861456] msm-dwc3 a600000.ssusb: DWC3 exited from low power mode
[ 1543.419252] usb usb9: We don't know the algorithms for LPM for this host,
disabling LPM.
[ 1543.652952] usb 8-1: new high-speed USB device number 2 using xhci-hcd
[ 1543.828410] usb 8-1: New USB device found, VID=2109, PID=2815
[ 1543.924696] usb 9-1: new SuperSpeed Gen 1 USB device number 2 using xhci-hcd
[ 1544.068498] usb 9-1: New USB device found, VID=2109, PID=0815
[ 1544.105971] msm-dwc3 8c00000.hsusb: DWC3 in low power mode
[ 1544.628957] usb 8-1.4: new full-speed USB device number 3 using xhci-hcd
[ 1544.766268] usb 8-1.4: not running at top speed; connect to a high speed hub
[ 1544.776265] usb 8-1.4: New USB device found, VID=25a4, PID=9311
```

Step 2. Connect a USB device to the board with a Type-C USB OTG cable. Input `dmesg` for detailed information of the USB storage.

```
# dmesg
[ 1541.985025] msm-dwc3 a600000.ssusb: could not transition HS PHY to L2
[ 1541.994981] msm-dwc3 a600000.ssusb: DWC3 in low power mode
[ 1541.995046] msm-dwc3 a600000.ssusb: Could not get usb psy
[ 1542.106137] msm-dwc3 8c00000.hsusb: DWC3 exited from low power mode
[ 1542.230006] msm-dwc3 8c00000.hsusb: DWC3 in low power mode
[ 1542.418317] ssusb redriver 6-001c: writing reg 0x01 failure
[ 1542.418326] ssusb redriver 6-001c: channel parameters update failure(-13).
[ 1542.462646] msm-dwc3 8c00000.hsusb: DWC3 exited from low power mode
[ 1542.856864] msm-usb-ssphy-qmp 88e8000.ssphy: USB DP QMP PHY: Update TYPEC CTRL(3)
[ 1542.861456] msm-dwc3 a600000.ssusb: DWC3 exited from low power mode
[ 1543.419252] usb usb9: We don't know the algorithms for LPM for this host,
disabling LPM.
[ 1543.652952] usb 8-1: new high-speed USB device number 2 using xhci-hcd
[ 1543.828410] usb 8-1: New USB device found, VID=2109, PID=2815
[ 1543.924696] usb 9-1: new SuperSpeed Gen 1 USB device number 2 using xhci-hcd
[ 1544.068498] usb 9-1: New USB device found, VID=2109, PID=0815
[ 1544.105971] msm-dwc3 8c00000.hsusb: DWC3 in low power mode
[ 1544.628957] usb 8-1.4: new full-speed USB device number 3 using xhci-hcd
[ 1544.766268] usb 8-1.4: not running at top speed; connect to a high speed hub
[ 1544.776265] usb 8-1.4: New USB device found, VID=25a4, PID=9311
[ 2244.341467] usb 8-1.1: new high-speed USB device number 4 using xhci-hcd
[ 2244.504065] usb 8-1.1: New USB device found, VID=3535, PID=6387
[ 2244.507378] usb-storage 8-1.1:1.0: USB Mass Storage device detected
[ 2244.512008] scsi host1: usb-storage 8-1.1:1.0
```

3.6.3. Operation procedure on LU and LE

➲ **NOTES**:

- This function has been verified in all software versions starting from the following:
  - **LU1.0.FC.r001002**
  - **SQC6490.LE1.0.ES.r000001**

3.6.3.1. USB storage over Type-C

Step 1. Connect a Micro USB cable to the board via the **UART debug port** (No.35 in Figure 1-1).

Step 2. Connect a USB storage device to the board with a Type-C USB OTG adapter via the **USB 3.1 Type-C connector** (No.12 in Figure 1-1).

Figure 3-15. Connect a USB Storage Device

Step 3. Enter `dmesg | grep usb` in UART for detailed information of the USB storage.

```
# dmesg | grep usb
[ 1254.520911] msm-usb-ssphy-qmp 88e8000.ssphy: USB DP QMP PHY: Update TYPEC CTRL(3)
[ 1254.524112] msm-dwc3 a600000.ssusb: DWC3 exited from low power mode
[ 1254.526459] ssusb-redriver 4-001c: writing reg 0x00 failure
[ 1254.526938] ssusb-redriver 4-001c: writing reg 0x01 failure
[ 1254.526957] ssusb-redriver 4-001c: channel parameters update failure(-107).
[ 1254.545657] usb usb9: We don't know the algorithms for LPM for this host,
disabling LPM.
[ 1255.133787] usb 9-1: new SuperSpeed Gen 1 USB device number 2 using xhci-hcd
[ 1255.172595] usb 9-1: New USB device found, VID=0951, PID=1666
[ 1255.174451] usb-storage 9-1:1.0: USB Mass Storage device detected
[ 1255.176901] scsi host1: usb-storage 9-1:1.0
```

Step 4. Execute the following command to view the device node.

```
# ls /dev/
```

Step 5. Mount the device node identified in the last step to */data/test*, then the files in the USB flash drive will be visible.

```
# mount -o rw,remount /
# mkdir -p /data/test
# mount /dev/sdg1 /data/test
# df -h
Filesystem     Size  Used Avail Use% Mounted on
/dev/root       99G  8.0G   91G   9% /
devtmpfs       2.4G     0  2.4G   0% /dev
tmpfs          3.6G     0  3.6G   0% /dev/shm
tmpfs          727M  4.3M  723M   1% /run
tmpfs          5.0M  4.0K  5.0M   1% /run/lock
tmpfs          3.6G     0  3.6G   0% /sys/fs/cgroup
tmpfs          3.6G  8.0K  3.6G   1% /tmp
/dev/sde9       59M   31M   28M  53% /dsp
/dev/sde4      220M   61M  160M  28% /firmware
tmpfs          727M     0  727M   0% /run/user/0
/dev/sdg1       29G  6.6G   23G  23% /data/test
```

3.6.3.2. USB camera over Type-C

Step 1. Connect a Micro USB cable to the board via the **UART debug port** (No.35 in Figure 1-1).

Step 2. Connect the board to PC with a USB Type-C cable via the **USB Type-C connector** (No.12 in Figure 1-1).

Step 3. Download the UVC test files.

Step 4. Enter the following commands to push the UVC test files to the board.

```
$ unzip uvctest.zip
$ cd uvctest
$ adb root
$ adb push 1080p_uvc /data
$ adb push 720p_uvc /data
$ adb push 480p_uvc /data
$ adb push yavta /data
$ adb shell
/ # chmod 777 /data/yavta
```

Step 5. Disconnect the USB Type-C cable.

Step 6. Connect a USB camera to the board with a Type-C USB OTG adapter via the **USB Type-C connector** (No.12 in Figure 1-1).

Figure 3-16. Connect USB Camera

Step 7. Enter `dmesg | grep usb` for detailed information.

```
# dmesg | grep usb
[ 2725.929529] usb 8-1: new high-speed USB device number 2 using xhci-hcd
[ 2726.670313] usb 8-1: New USB device found, VID=046d, PID=0892
[ 2726.686328] input: HD Pro Webcam C920 as
/devices/platform/soc/a600000.ssusb/a600000.dwc3/xhci-hcd.3.auto/usb8/8-1/8-
1:1.0/input/input2
```

Step 8. Enter the following commands in UART to capture images via the USB video device.

```
/ # cd /data
/data # ./yavta -f YUYV -s 1920x1080 -t 1/30 -c10 -F/data/testyuv1 /dev/video2
/data # ./yavta -f YUYV -s 1280x720 -t 1/30 -c10 -F/data/testyuv2 /dev/video2
/data # ./yavta -f YUYV -s 640x480 -t 1/30 -c10 -F/data/testyuv3 /dev/video2
```

➲ **NOTE**: Use the `/dev/video4` node on LE.

Step 9. Connect the device to the PC using a Type-C cable and enter the following commands to pull the captured image files and check the yuv file on the PC.

```
$ adb pull /data/testyuv1
$ adb pull /data/testyuv2
$ adb pull /data/testyuv3
```

3.6.3.3. USB audio over Type-C

Step 1. Connect a Micro USB cable to the board via the **UART debug port** (No.35 in Figure 1-1).

Step 2. Connect a USB audio device to the board with a Type-C to USB OTG adapter via the **USB Type-C connector** (No.12 in Figure 1-1).

Figure 3-17. Connect a USB Audio Device

Step 3. The board shall detect a USB audio device once connected by a Type-C USB OTG adapter. Enter `dmesg | grep usb` in UART for detailed information.

```
# dmesg | grep usb
[49394.971354] msm-usb-ssphy-qmp 88e8000.ssphy: USB DP QMP PHY: Update TYPEC CTRL(3)
[49394.983391] msm-dwc3 a600000.ssusb: DWC3 exited from low power mode
[49394.990622] ssusb-redriver 4-001c: writing reg 0x00 failure
[49395.002957] ssusb-redriver 4-001c: writing reg 0x01 failure
[49395.009534] ssusb-redriver 4-001c: channel parameters update failure(-107).
[49395.026988] usb usb9: We don't know the algorithms for LPM for this host,
disabling LPM.
[49395.821274] usb 8-1: new full-speed USB device number 2 using xhci-hcd
[49396.015725] usb 8-1: New USB device found, VID=1b3f, PID=2008
[49396.177689] input: GeneralPlus USB Audio Device as
/devices/platform/soc/a600000.ssusb/a600000.dwc3/xhci-hcd.3.auto/usb8/8-1/8-
1:1.3/0003:1B3F:2008.0001/input/input5
[49396.235548] hid-generic 0003:1B3F:2008.0001: input,hidraw0: USB HID v2.01 Device
[GeneralPlus USB Audio Device] on usb-xhci-hcd.3.auto-1/input3
```

Step 4. Enter `cat /proc/asound/cards` in UART to view information of the sound card.

```
# cat /proc/asound/cards
 0 [lahainayupikiot]: lahaina-yupikio - lahaina-yupikiot-snd-card
                      lahaina-yupikiot-snd-card
 1 [Device         ]: USB-Audio - USB Audio Device
                      GeneralPlus USB Audio Device at usb-xhci-hcd.3.auto-1, full speed
```

Step 5. Enter the following command in UART to record an audio file using the UAC device.

```
# arecord -Dhw:1,0 -d 10 -f cd -r 48000 -c 1 -t wav /data/test.wav
Recording WAVE '/data/test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Mono
```

➲ **NOTE:** The command to capture audio files using UAC devices on LE is "`# qtitinycap /data/test.wav -D 0 -d 0 -c 1`".

Step 6. Enter the following command in UART to play back the audio file via the speaker.

```
# gst-launch-1.0 filesrc location=/data/test.wav ! wavparse ! audioconvert !
pulsesink volume=0.2
Setting pipeline to PAUSED ...
Pipeline is PREROLLING ...
/mnt/vendor/persist/factory/audio/aw_cali.bin failed!
Redistribute latency...
Pipeline is PREROLLED ...
Setting pipeline to PLAYING ...
New clock: GstPulseSinkClock
Got EOS from element "pipeline0".
Execution ended after 0:00:09.900604059
Setting pipeline to NULL ...
Freeing pipeline ...
```

➲ **NOTE**: This step is currently unavailable in the latest SDK version SQC6490.LE1.0.ES.r000001.

3.6.3.4. USB Ethernet over Type-C

Step 1. Connect a Micro USB cable to the board via the **UART debug port** (No.35 in Figure 1-1).

Step 2. Insert an Ethernet-to-USB adapter to a Type-C to USB OTG adapter, and connect the latter to the board via the **USB Type-C connector** (No.12 in Figure 1-1).

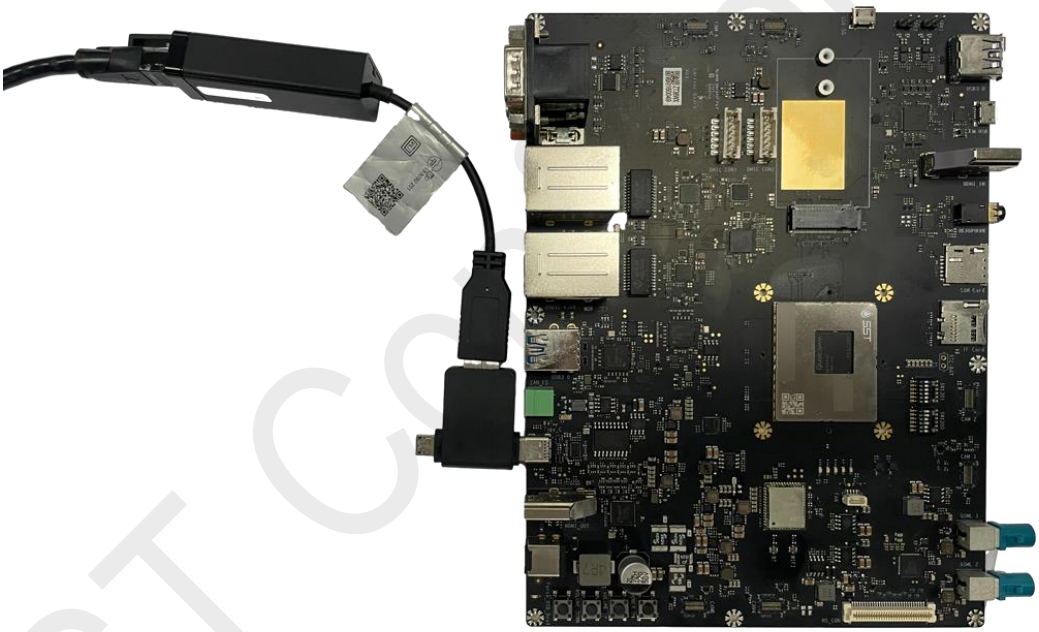


Figure 3-18. Connect Type-C to USB OTG Adapter

Step 3. Enter `ifconfig` to view Ethernet status.

```
/ # ifconfig eth2
      inet 10.9.5.58  netmask 255.255.255.0  broadcast 10.9.5.255
      inet6 fe80::5106:5755:2f87:fea  prefixlen 64  scopeid 0x20<link>
      ether 9c:eb:e8:54:33:71  txqueuelen 1000  (Ethernet)
      RX packets 26  bytes 2376 (2.3 KB)
      RX errors 0  dropped 0  overruns 0  frame 0
      TX packets 33  bytes 3112 (3.1 KB)
      TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Step 4. Perform a ping test via USB Ethernet.

```
# ping www.google.com
PING www.google.com.w.kunlungr.com (124.165.125.114): 56 data bytes
64 bytes from 124.165.125.114: icmp_seq=0 ttl=47 time=19.574 ms
64 bytes from 124.165.125.114: icmp_seq=1 ttl=47 time=18.912 ms
64 bytes from 124.165.125.114: icmp_seq=2 ttl=47 time=18.613 ms
64 bytes from 39.156.66.18: icmp_seq=3 ttl=49 time=25.247 ms
```

3.7. USB Type-A connector

3.7.1. Hardware configuration

Refer to the figure below to connect a USB flash drive, UVC, UAC, or a USB ethernet device to the board via either **USB 3.0 Type A connector** (No. 8, No. 9, No. 10 or No. 36 in Figure 1-1).

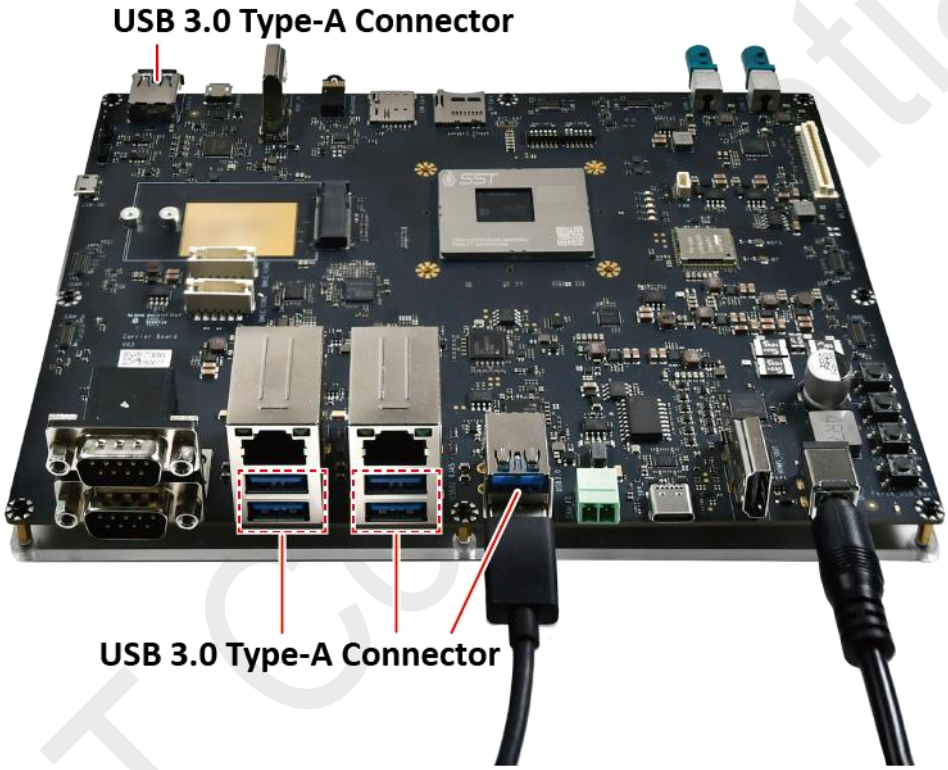


Figure 3-19. USB 3.0 Type-A Connectors

3.7.2. Operation procedure on LA

➲ **NOTE**: This function has been verified in all software versions starting from **LA3.0.FC.r001002**.

Step 1. When the mouse device is connected to the board via either **USB 3.0 Type A connector** (No. 8, No. 9, No. 10 or No. 36 in Figure 1-1)**,** a cursor will appear on the Android screen.

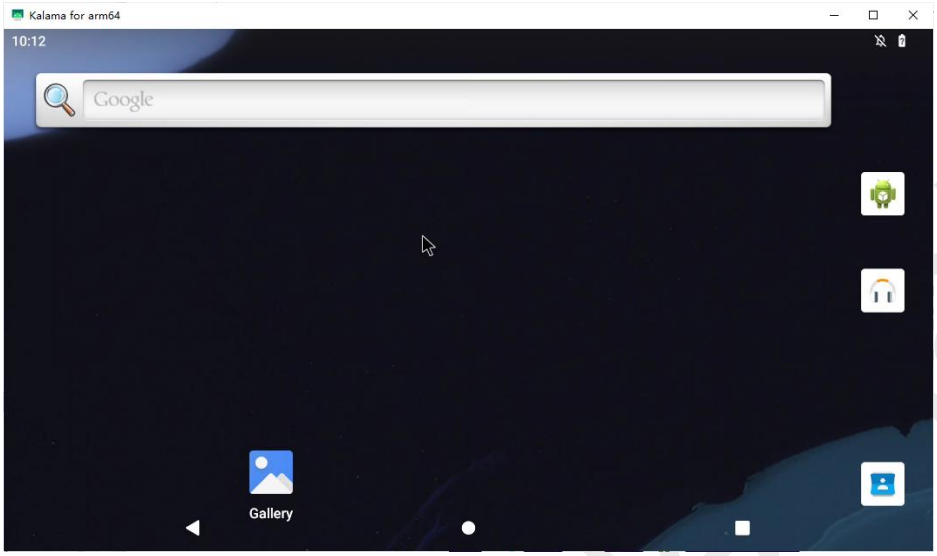


Figure 3-20. Android Interface

Step 2. Move the mouse and observe whether you can operate the Android system normally. Input `dmesg` for detailed information of the USB mouse.

```
$ adb root
$ adb shell
kalama:/ # dmesg | grep usb
[   10.248851] usb 3-2: New USB device found, idVendor=413c, idProduct=301a,
bcdDevice= 1.00
[   10.248859] usb 3-2: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[   10.248863] usb 3-2: Product: Dell MS116 USB Optical Mouse
[   10.248866] usb 3-2: Manufacturer: PixArt
```

3.7.3. Operation procedure on LU and LE

➲ **NOTES**:

- This function has been verified in all software versions starting from the following:
  - **LU1.0.FC.r001002**
  - **SQC6490.LE1.0.ES.r000001**
- USB 3.0 Type A connectors No.8 and No.9 (refer to Figure 1-1) are currently unavailable in SDK version SQC6490.LE1.0.ES.r000001.

3.7.3.1. USB storage over Type-A

Step 1. Insert a USB flash drive to the board via either **USB 3.0 Type A connector** (No. 8, No. 9, No. 10 or No. 36 in Figure 1-1), then you can obtain the device identifier of the newly inserted USB flash drive (i.e., "sdg1") through the system log of "dmesg".

```
$ adb root
$ adb shell
# dmesg
……..
[ 5261.663644] sd 0:0:0:0: [sda] Synchronizing SCSI cache
[ 5264.974760] usb 5-3: new SuperSpeed Gen 1 USB device number 3 using renesas xhci
[ 5265.015826] usb 5-3: New USB device found, VID=0951, PID=1666
[ 5265.017896] usb-storage 5-3:1.0: USB Mass Storage device detected
[ 5265.031209] scsi host1: usb-storage 5-3:1.0
[ 5266.060075] scsi 1:0:0:0: Direct-Access     Kingston DataTraveler 3.0     PQ: 0
ANSI: 6
[ 5266.064744] sd 1:0:0:0: Attached scsi generic sg9 type 0
[ 5266.084472] sd 1:0:0:0: [sdg] 60437492 512-byte logical blocks: (30.9 GB/28.8
GiB)
[ 5266.085648] sd 1:0:0:0: [sdg] Write Protect is off
[ 5266.085692] sd 1:0:0:0: [sdg] Mode Sense: 4f 00 00 00
[ 5266.086614] sd 1:0:0:0: [sdg] Write cache: disabled, read cache: enabled, doesn't
support DPO or FUA
[ 5266.095790]  sdg: sdg1
[ 5266.108038] sd 1:0:0:0: [sdg] Attached SCSI removable disk
[ 5266.124179] sd 0:0:0:0: [sda] Synchronizing SCSI cache
[ 5266.153761] (virq:irq_count)- 3:393538 25:179254 15:93216 19:11504 252:7353
43:5760 352:3575 22:3490 20:2604 351:2583
[ 5266.153809] (cpu:irq_count)- 0:117365 1:262907 2:165097 3:96298 4:3163 5:4200
6:62045 7:5434
[ 5266.153863] (ipi:irq_count)- 0:677479 1:533429 2:0 3:0 4:0 5:78912 6:0
```

Step 2. Execute the following command to view the device node.

```
# ls /dev/
```

Step 3. Mount the device node identified in the last step to */data/test*, then the files in the USB flash drive will be visible.

```
# mount -o rw,remount /
# mkdir -p /data/test
# mount /dev/sdg1 /data/test
# df -h
Filesystem     Size  Used Avail Use% Mounted on
/dev/root       99G  8.0G   91G   9% /
devtmpfs       2.4G     0  2.4G   0% /dev
tmpfs          3.6G     0  3.6G   0% /dev/shm
tmpfs          727M  4.3M  723M   1% /run
tmpfs          5.0M  4.0K  5.0M   1% /run/lock
tmpfs          3.6G     0  3.6G   0% /sys/fs/cgroup
tmpfs          3.6G  8.0K  3.6G   1% /tmp
/dev/sde9       59M   31M   28M  53% /dsp
/dev/sde4      220M   61M  160M  28% /firmware
tmpfs          727M     0  727M   0% /run/user/0
/dev/sdg1       29G  6.6G   23G  23% /data/test
```

3.7.3.2. USB camera over Type-A

Step 1. Connect a USB camera to the board via the **USB Type-A connector** (No. 8, No. 9, No. 10 or No. 36 in Figure 1-1).

Figure 3-21. Connect the USB Camera

Step 2. Connect the board to PC with a USB Type-C cable via the **USB Type-C connector** (No.12 in Figure 1-1)

Step 3. Download the UVC test files (uvctest.zip).

Step 4. Enter the following commands to push the UVC test files to the board.

```
$ unzip uvctest.zip
$ cd uvctest
$ adb root
$ adb push 1080p_uvc /data
$ adb push 720p_uvc /data
$ adb push 480p_uvc /data
$ adb push yavta /data
$ adb shell chmod 777 /data/yavta
```

Step 5. The board shall detect a USB video device once connected by a Type-A connector. Input `dmesg | grep usb` for detailed information.

```
$ adb shell
# dmesg | grep usb
[ 9233.726048] usb 4-3: New USB device found, VID=046d, PID=0892
[ 9233.728126] uvcvideo: Found UVC 1.00 device HD Pro Webcam C920 (046d:0892)
[ 9233.746095] input: HD Pro Webcam C920 as
/devices/platform/soc/1c08000.qcom,pcie/pci0001:00/0001:00:00.0/0001:01:00.0/0001:02
:02.0/0001:04:00.0/usb4/4-3/4-3:1.0/input/input3
[ 9234.921896] (virq:irq_count)- 3:674024 25:315167 15:175569 19:20199 252:11266
22:6129 43:5760 20:4654 352:3575 310:3015
[ 9234.921983] (cpu:irq_count)- 0:186955 1:468042 2:301118 3:161854 4:3163 5:5310
6:106098 7:5435
[ 9234.922044] (ipi:irq_count)- 0:1175740 1:927620 2:0 3:0 4:0 5:131786 6:0
```

Step 6. Enter the following commands to capture image via USB video device.

```
/ # cd /data
/data # ./yavta -f YUYV -s 1920x1080 -t 1/30 -c10 -F/data/testyuv1 /dev/video2
/data # ./yavta -f YUYV -s 1280x720 -t 1/30 -c10 -F/data/testyuv2 /dev/video2
/data # ./yavta -f YUYV -s 640x480 -t 1/30 -c10 -F/data/testyuv3 /dev/video2
```

➲ **NOTE**: Use the `/dev/video4` node on LE.

Step 7. Enter the following commands via the USB Type-C cable to pull the captured image files and check the yuv file on PC.

```
$ adb pull /data/testyuv1
$ adb pull /data/testyuv2
$ adb pull /data/testyuv3
```

3.7.3.3. USB audio over Type-A

Step 1. Connect a USB audio device to the board via the **USB Type-A connector** (No. 8, No. 9, No. 10 or No. 36 in Figure 1-1).

Figure 3-22. Connect the USB Audio Device

Step 2. Connect the board to PC with a USB Type-C cable via the **USB Type-C connector** (No.12 in Figure 1-1).

Step 3. The board shall detect an USB audio device once connected by Type-A connector. Enter `dmesg | grep usb` for detailed information.

```
$ adb root
$ adb shell
# dmesg | grep usb
[ 6809.173176] usb 4-3: new full-speed USB device number 2 using renesas xhci
[ 6809.385720] usb 4-3: New USB device found, VID=1b3f, PID=2008
[ 6809.616187] input: GeneralPlus USB Audio Device as
/devices/platform/soc/1c08000.qcom,pcie/pci0001:00/0001:00:00.0/0001:01:00.0/0001:02
:02.0/0001:04:00.0/usb4/4-3/4-3:1.3/0003:1B3F:2008.0001/input/input2
[ 6809.671771] hid-generic 0003:1B3F:2008.0001: input,hidraw0: USB HID v2.01 Device
[GeneralPlus USB Audio Device] on usb-0001:04:00.0-3/input3
[ 6810.089907] (virq:irq_count)- 3:504082 25:233167 15:124769 19:14894 252:8881
43:5760 22:4528 352:3575 20:3409 340:2773
[ 6810.089994] (cpu:irq_count)- 0:145650 1:343951 2:217480 3:123366 4:3163 5:4641
6:79274 7:5435
[ 6810.090055] (ipi:irq_count)- 0:875306 1:689928 2:0 3:0 4:0 5:99875 6:0
```

Step 4. Enter the command "`cat /proc/asound/cards`" to view information of the sound card.

```
# cat /proc/asound/cards
 0 [lahainayupikiot]: lahaina-yupikio - lahaina-yupikiot-snd-card
                      lahaina-yupikiot-snd-card
 1 [Device         ]: USB-Audio - USB Audio Device
                      GeneralPlus USB Audio Device at usb-0001:04:00.0-3, full speed
```

➲ **NOTE:** Enter the following command on LU and LE respectively to capture audio files using UAC devices.

- On LU:

```
# arecord -Dhw:1,0 -d 10 -f cd -r 48000 -c 1 -t wav /data/test.wav
Recording WAVE '/data/test.wav' : Signed 16 bit Little Endian, Rate 48000 Hz,
Mono
```

- On LE:

```
# qtitinycap /data/test.wav -D 0 -d 0 -c 1
```

3.7.3.4. USB Ethernet over Type-A

➲ **NOTE:** The device can support up to 7 Type A to USB network cards simultaneously.

Step 1. Connect a USB Ethernet adapter to the board via the **USB Type-A connector** (No. 8, No. 9, No. 10 or No. 36 in Figure 1-1).

Figure 3-23. Connect the USB Ethernet Adapter

Step 2. Connect the board to PC with a USB Type-C cable via the **USB Type-C connector** (No.12 in Figure 1-1).

Step 3. Enter the command "`ifconfig`"to view Ethernet status.

```
$ adb root
$ adb shell
/ # ifconfig eth2
      inet 10.9.5.58  netmask 255.255.255.0  broadcast 10.9.5.255
      inet6 fe80::5106:5755:2f87:fea  prefixlen 64  scopeid 0x20<link>
      ether 9c:eb:e8:54:33:71  txqueuelen 1000  (Ethernet)
      RX packets 26  bytes 2376 (2.3 KB)
      RX errors 0  dropped 0  overruns 0  frame 0
      TX packets 33  bytes 3112 (3.1 KB)
      TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Step 4. Perform a ping test via USB Ethernet.

```
# ping www.google.com
PING www.google.com.w.kunlungr.com (124.165.125.114): 56 data bytes
64 bytes from 124.165.125.114: icmp_seq=0 ttl=47 time=19.574 ms
64 bytes from 124.165.125.114: icmp_seq=1 ttl=47 time=18.912 ms
64 bytes from 124.165.125.114: icmp_seq=2 ttl=47 time=18.613 ms
```

➲ **NOTE**: The name of Ethernet on LE may vary, such as "enP1p4s0u2c2". To check the IP address on LE, you can run the "`ifconfig`" command.

3.8. Sensor

3.8.1. Operation procedure on LA

➲ **NOTE:** This function has been verified in all software versions starting from **LA3.0.ES.r000001**.

Step 1. Boot up your system.

Step 2. Open **QSensorTest app**.

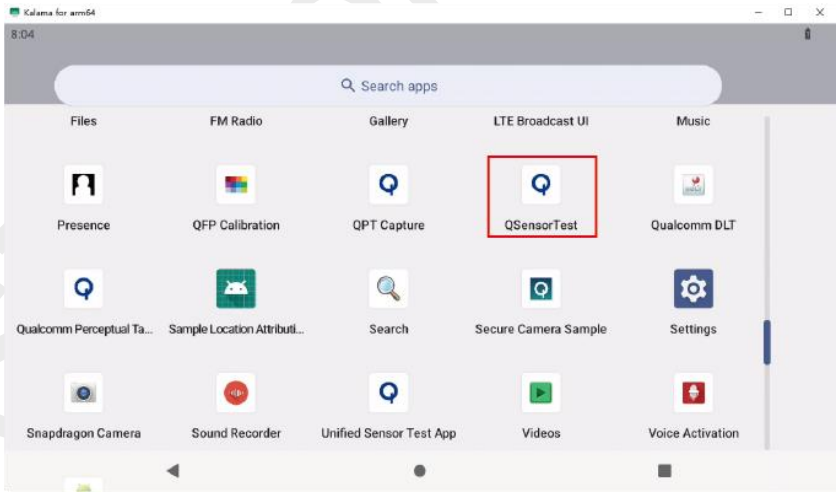


Figure 3-24. QSensorTest App

Step 3. Click either sensor on the screen and observe whether values of sensors (icm4x6xa, ak0991x) change.

➲ **NOTE: "**ICM-4x6xA" in the **QSensorTest app** corresponds to ICM-42607 on the DK, while "ak0991x" corresponds to AK09919C on the DK.

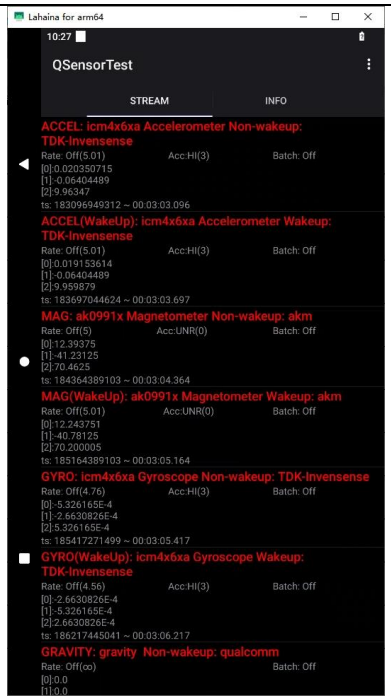


Figure 3-25. QSensorTest Interface

3.8.2. Operation procedure on LU

3.8.2.1. Accelerometer and Gyroscope sensor (ICM-42607)

Accelerometer and Gyroscope sensor (ICM-42607) are currently unavailable in the latest SDK version SQC6490 **LU1.0.r001002**.

3.8.2.2. Magnetometer sensor (AK09919C)

Execute the following commands to test the functionality of the sensor:

```
$ adb shell
sh-5.0# ssc_drva_test -sensor=mag -duration=10 -sample_rate=-1
3 ssc_drva_test version 1.13
3 ssc_drva_test -sensor=mag -duration=10 -sample_rate=-1
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=2181397
3 -time_to_last_event=-222324
3 -sample_ts=127778091221
3 -total_samples=989
3 -avg_delta=191805
3 -recvd_phy_config_sample_rate=100
3 -random_seed_used=3032008473
3 -num_request_sent=2
3 -first_sample_timestamp=127588257050
3 handle_event
3 received event: PASS
3 enter send_memory_log_req cookie: 3
3 exit send_memory_log_req
3 PASS
```

3.8.3. Operation procedure on LE

This function is currently unavailable in the latest SDK version **SQC6490.LE1.0.ES.r000001**.

3.9. Wi-Fi and Bluetooth connection

3.9.1. Connect to Wi-Fi network

3.9.1.1. Hardware configuration

Refer to the figure below to connect the Wi-Fi antenna to the board via the **Wi-Fi antenna connector** (No. 23 in Figure 1-1).

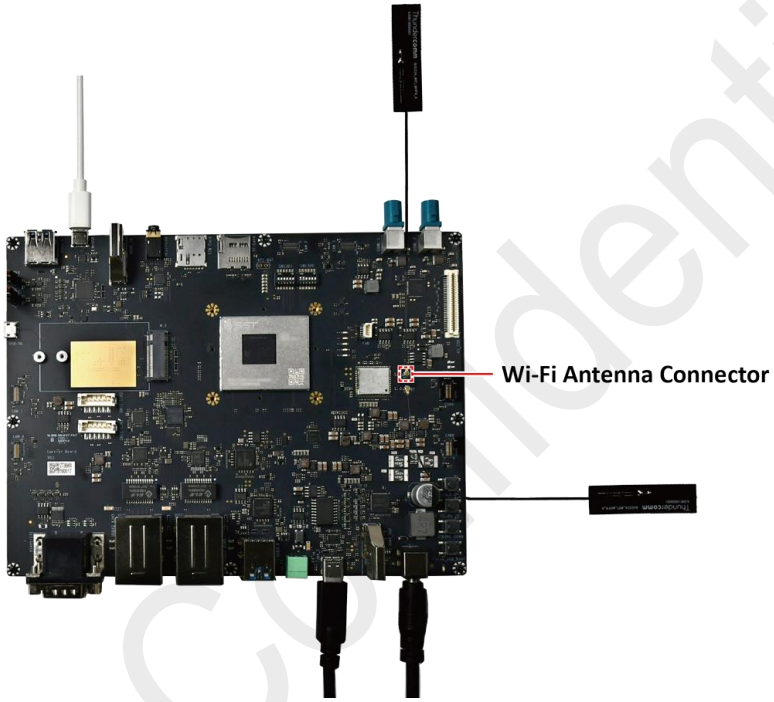


Figure 3-26. Wi-Fi Antenna Connector

3.9.1.2. Operation procedure on LA

3.9.1.2.1. Set up Wi-Fi STA mode

➲ **NOTE:** This function has been verified in all software versions starting from **LA3.0.FC.r001002**.

Step 1. Turn on your device.

Step 2. Swipe downwards shortly from the top edge of the screen to enter **Quick Settings** bar.

Step 3. Click the Internet icon to open the interface for Wi-Fi settings.

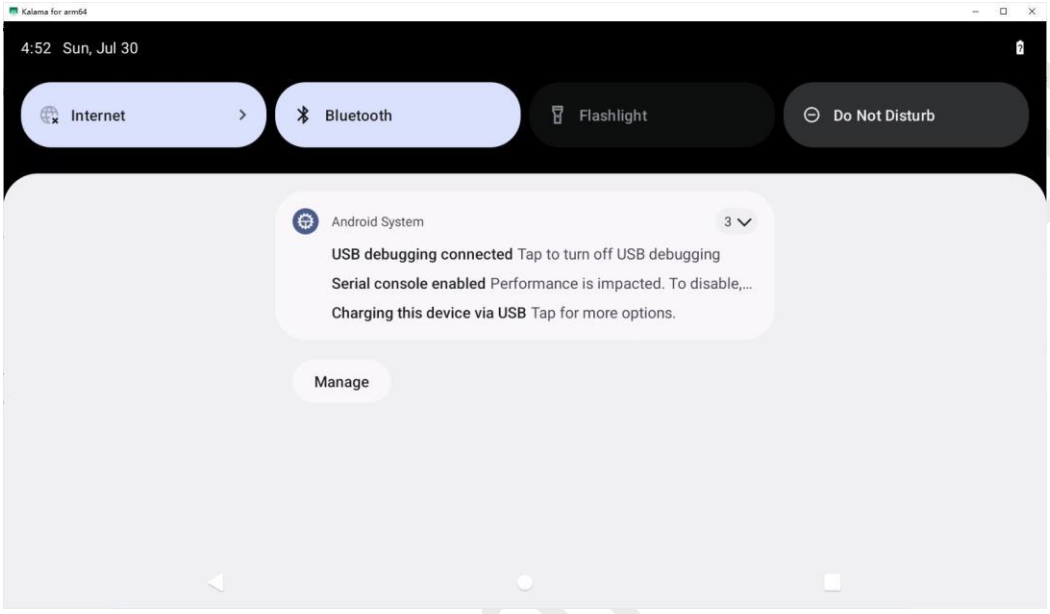


Figure 3-27. Wi-Fi Settings Interface

Step 4. Click the Wi-Fi switch  to turn on Wi-Fi connection function.

Figure 3-28. Turn on Wi-Fi Switch

Step 5. Click the network name you prefer to connect and enter the password.

➲ **NOTE:** Networks that require password have a lock icon marked.

Step 6. Confirm the connection with an IP obtained.

```
$ adb shell
kalama:/ # ifconfig wlan0
wlan0    Link encap:Ethernet  HWaddr fe:8f:f8:88:40:54  Driver wlan_sdio
         inet addr:192.168.1.107  Bcast:192.168.1.255  Mask:255.255.255.0
         inet6 addr: fe80::fc8f:f8ff:fe88:4054/64 Scope: Link
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:83 errors:0 dropped:0 overruns:0 frame:0
         TX packets:89 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:32086 TX bytes:22434
```

Step 7. Perform a ping test.

```
# ping www.google.com
PING www.google.com.w.kunlungr.com (182.118.13.176) 56(84) bytes of data.
64 bytes from hn.kd.ny.adsl (182.118.13.176): icmp_seq=1 ttl=49 time=16.2 ms
64 bytes from hn.kd.ny.adsl (182.118.13.176): icmp_seq=2 ttl=49 time=16.3 ms
```

3.9.1.2.2. Set up the Wi-Fi AP mode

This function is unavailable in the latest LA software version.

3.9.1.3. Operation procedure on LU

➲ **NOTE**: This function has been verified in all software versions starting from **LU1.0.FC.r001002**.

3.9.1.3.1. Set up the Wi-Fi STA mode

Step 1. Turn on your device.

➲ **NOTE:** The wlan0 interface is enabled by default upon device boot.

Step 2. Enter the following commands to enable Wi-Fi Station Mode and connect to AP network.

➲ **NOTE:** Modify "ssid" and "psk" of the Wi-Fi AP as required.

```
$ adb root
$ adb shell
# killall wpa_supplicant
# wpa_supplicant -Dnl80211 -iwlan0 -c /etc/wlan/wpa_supplicant.conf &
# sh-5.0# wpa_cli -iwlan0
wpa_cli v2.10-devel
Copyright (c) 2004-2019, Jouni Malinen <j@w1.fi> and contributors

This software may be distributed under the terms of the BSD license.
See README for more details.



Interactive mode

> add_network
1
> set_network 1 ssid "TP-LINK_5G_4295"
OK
> set_network 1 psk "12345678"
OK
> enable_network 1
wlan0: SME: Trying to authenticate with 58:41:20:b1:42:97 (SSID='TP-LINK_5G_4295'
freq=5180 MHz)
OK
> q
#
```

Step 3. Perform a ping test.

```
# ping www.google.com
PING www.google.com.w.kunlungr.com (111.62.92.186): 56 data bytes
64 bytes from 111.62.92.186: icmp_seq=0 ttl=52 time=201.575 ms
64 bytes from 111.62.92.186: icmp_seq=4 ttl=52 time=274.890 ms
64 bytes from 111.62.92.186: icmp_seq=5 ttl=52 time=91.587 ms
```

3.9.1.3.2. Set up the Wi-Fi AP mode

Step 1. Enter the following commands to enable WLAN.

```
$ adb root
$ adb shell
/ # killall wpa_supplicant
/ # killall dhcpcd
/ # killall dnsmasq
/ # hostapd -dddd -B /etc/wlan/hostapd.conf
/ # ifconfig wlan0 192.168.3.1 up
```

Step 2. Enter the following commands to enable the Wi-Fi AP Mode.

```
/ # dnsmasq -p 9000 -i wlan0 -l /data/dnsmasq.leases --no-daemon -- no-resolv --no-
poll --dhcp-range=192.168.3.100,192.168.3.200,1h &
dnsmasq: started, version 2.80 cachesize 150
dnsmasq: compile time options: IPv6 GNU-getopt DBus i18n IDN DHCP DHCPv6 no-Lua TFTP
conntrack ipset auth nettlehash DNSSEC loop-detect inotify dumpfile
dnsmasq: warning: no upstream servers configured
dnsmasq-dhcp: DHCP, IP range 192.168.3.100 -- 192.168.3.200, lease time 1h
dnsmasq: read /etc/hosts - 2 addresses
```

Step 3. Check the AP name and the password.

```
/ # cat /etc/wlan/hostapd.conf | grep ssid
ssid=QCS6490_SoftAP

/ # cat /etc/wlan/hostapd.conf | grep wpa_passphrase
wpa_passphrase=123456789
```

3.9.1.4. Operation procedure on LE

This function is currently unavailable in the latest SDK version SQC6490.LE1.0.ES.r000001.

3.9.2. Connect to a Bluetooth-enabled device

3.9.2.1. Hardware configuration

Connect a BT Antenna to the board via the **BT antenna connector** (No. 21 in Figure 1-1).

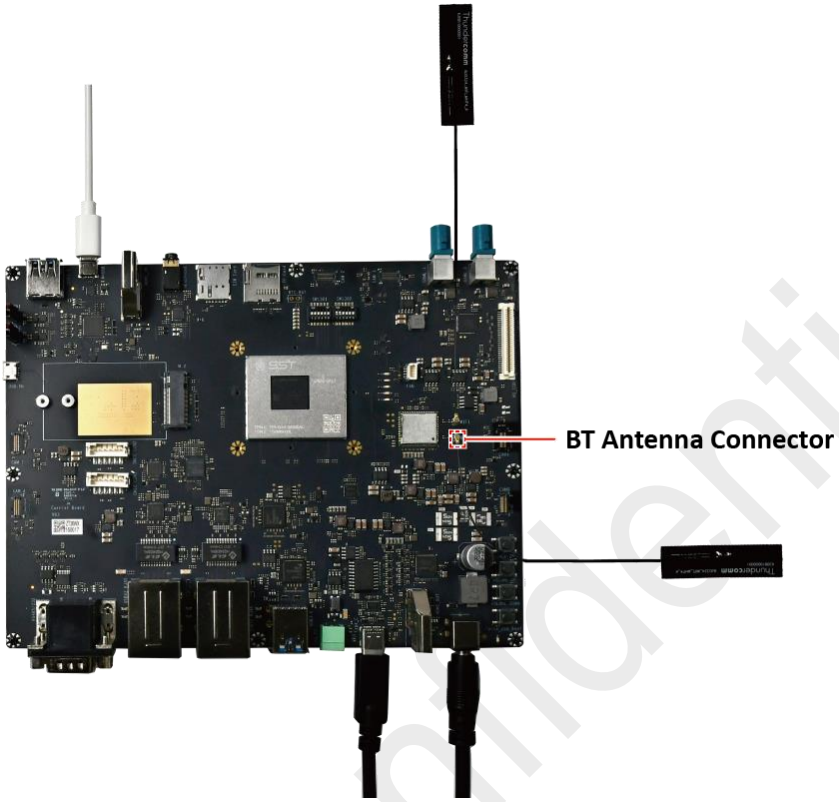


Figure 3-29. BT Antenna Connector

3.9.2.2. Operation procedure on LA

➲ **NOTE**: This function has been verified in all software versions starting from **LA3.0.FC.r001002**.

Step 1. Swipe downwards shortly from the top edge of the screen to enter **Quick Settings** bar.

Step 2. Click the Bluetooth icon to turn on Bluetooth connection function.

Step 3. Long press the **Bluetooth** icon to access the Bluetooth settings interface.

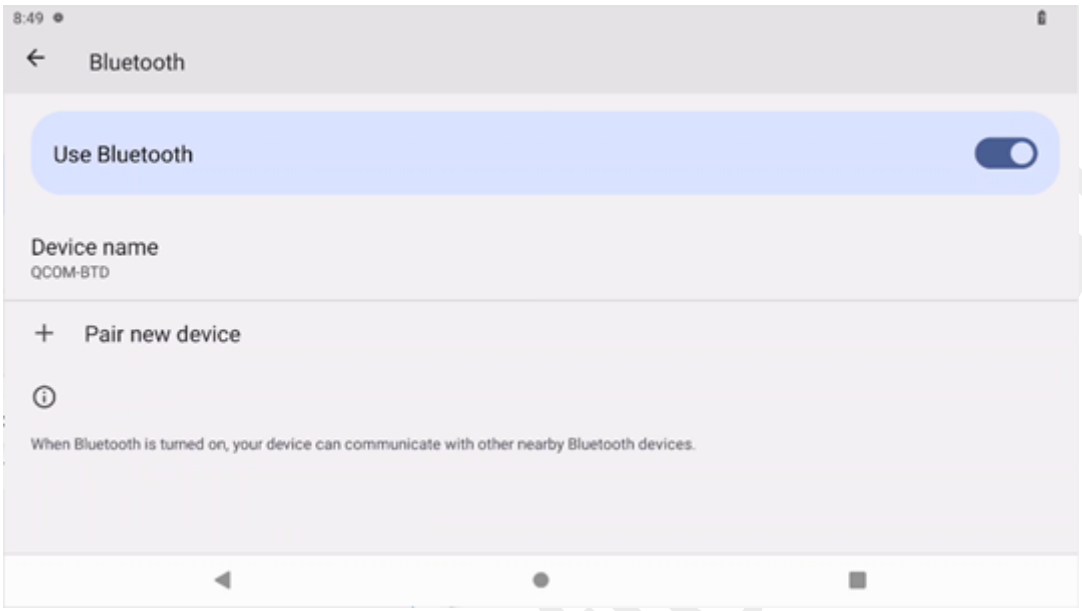


Figure 3-30. Bluetooth Settings

Step 4. Click on **Pair new device** to search the Bluetooth-enabled device around.

Figure 3-31. Pair New Device

Step 5. Click on the Bluetooth name to pair the Bluetooth-enabled device with your device.

3.9.2.3. Operation procedure on LU

➲ **NOTE**: This function has been verified in all software versions starting from **LU1.0.FC.r001002**.

Step 1. Before running **btapp**, run btproperty in the background (run only once at the beginning).

```
$ adb root
$ adb shell
/ # btproperty & btapp
```

Step 2. After running **btapp**, input gap_menu and press **Enter**.

```
gap_menu

****************** Menu ********************
        enable
        disable
        inquiry
        cancel_inquiry
        get_role_req<space><bt_address>    eg. get_role_req 00:11:22:33:44:55
        pair<space><bt_address><space><transport>        eg.  pair  00:11:22:33:44:55
0(auto)/1(BREDR)/2(BLE)
        unpair<space><bt_address>    eg. unpair 00:11:22:33:44:55
        inquiry_list
        bonded_list
        get_state
        get_bt_name
        get_bt_address
        set_bt_name<space><bt name>    eg. set_bt_name MDM_Fluoride
        set_scan_mode<space><scan mode value (range 0-2)>    eg. set_scan_mode 0 --0-
BT_SCAN_MODE_NONE,1-                                    BT_SCAN_MODE_CONNECTABLE,2-
BT_SCAN_MODE_CONNECTABLE_DISCOVERABLE
        set_afh<space><AFH_Host_Channel_Classification>                eg.   set_afh
1122334455667788 99f0
        send_hci_cmd<space><hci_cmd>      eg. send_hci_cmd 01,04,05,33,8b,9e,0a,00 -
For Inquiry
        read_clock<space><which_clock   range(0-1)><space><bt_address>            eg.
read_clock 0(local)/1(acl connection) 00:11:22:33:44:55
        main_menu
        switch_role_req<bt_address><space><new_role>            eg.   switch_role_req
00:11:22:33:44:55 0 or get_role_req 00:11:22:33:44:55 1
 ****************************************
```

Step 3. Input enable and press **Enter** to enable Bluetooth.

```
enable
wcnssfilter: no process found
btsnoop: no process found
qcbtdaemon: no process found
diag:successfully connected to socket 92
 current State = 0, new state = 1
 BT State is ON
```

Step 4. Input inquiry and press **Enter** to start inquiry.

```
inquiry
 Inquiry Started
Device Found details:
Found device Addr: 9c:fb:d5:d8:ac:fe
Found device Name: vivo X9
Device Type is: 1
Device Found details:
Found device Addr: d4:5e:ec:c3:77:ca
Found device Name: SST
Device Type is: 1
Device Found details:
Found device Addr: 28:64:b0:bf:9b:2b
Found device Name: HUAWEo)I P40
Device Type is: 1
 Inquiry Stopped automatically
```

Step 5. Input `inquiry_list` and press **Enter** to check the inquiry list.

```
inquiry_list

*************************** Inquiry List *********************************
1        vivo X9                                    9c:fb:d5:d8:ac:fe
2        SST                                  d4:5e:ec:c3:77:ca
3        HUAWEo)I P40                                28:64:b0:bf:9b:2b
4                                                   f8:c4:ae:f9:a2:c2
5                                                   d0:49:7c:b9:ab:b0
*************************** End of List **********************************
```

Step 6. Enter the following command to pair outgoing SSP.

```
pair 9c:fb:d5:d8:ac:fe 0
```

Step 7. In the following example (pair 9c:fb:d5:d8:ac:fe 0), enter `Yes` to accept or `No` to reject the outgoing pairing, and then press **Enter**.

```
************************************************
 BT pairing request::Device vivo X9::Pairing Code:: 283007
************************************************
 ** Please enter yes / no **
yes

************************************************
 Pairing state for vivo X9 is BONDED
************************************************
```

Step 8. Input `bonded_list` and press **Enter** to check the bonded list.

```
bonded_list

*************************** Bonded Device List ***************************
vivo X9                                    9c:fb:d5:d8:ac:fe
***************************  End of List *********************************
```

Step 9. To disconnect the bonded device, input `disable` and press **Enter**.

```
disable
 current State = 1, new state = 0
ConfigureAudio: audio_device is NULL
wcnssfilter: no process found
 BT State is OFF
```

Step 10. To exit from **btapp**, navigate to the main menu and input `exit`.

3.9.2.4. Operation procedure on LE

This function is currently unavailable in the latest SDK version SQC6490.LE1.0.ES.r000001.

3.10. Ethernet

3.10.1. Hardware configuration

Connect the board to a local network through the LAN Port of the router and **RJ 45 connector** (No.8 or No.9 in Figure 1-1) of the device with an Ethernet cable.

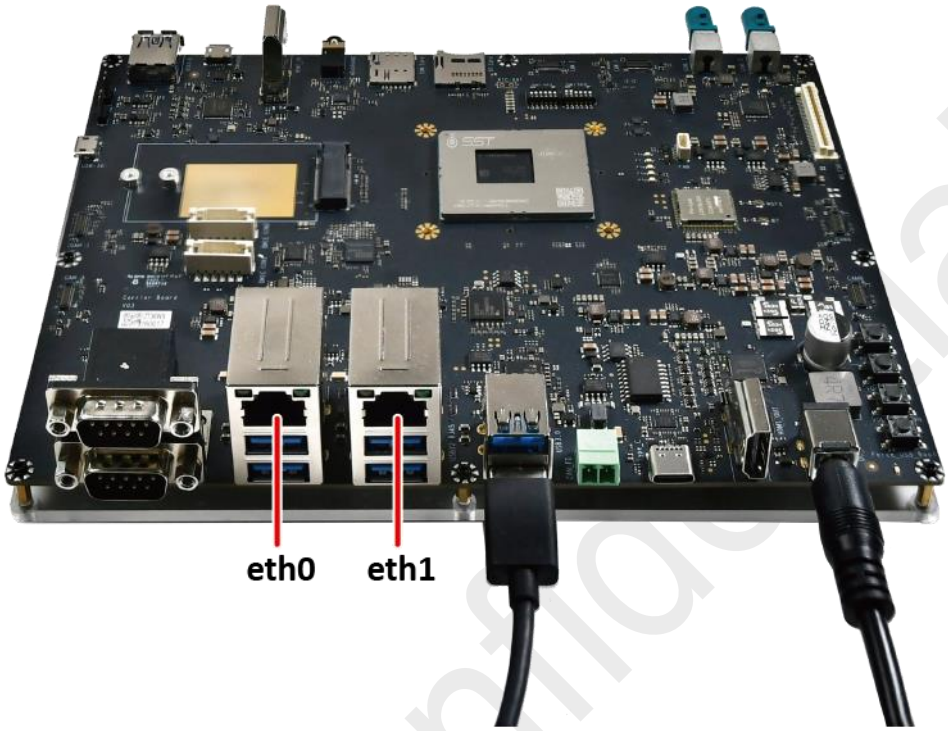


Figure 3-32. Ethernet Connector

3.10.2. Operation procedure on LA

➲ **NOTE**: This function has been verified in all software versions starting from **LA3.0.FC.r001002**.

Step 1. Confirm the connection setup with an IP obtained.

```
$ adb root
$ adb shell
kalama:/ # ifconfig eth0
kalama:/ # ifconfig eth1
```

```
eth0      Link encap:Ethernet  HWaddr ec:21:e5:10:4f:ea  Driver tc956x_pci-eth
          inet addr:192.168.8.189  Bcast:192.168.8.255  Mask:255.255.255.0
          inet6 addr: fe80::eb8e:c9e6:6ee9:c9fa/64 Scope: Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:48 errors:0 dropped:0 overruns:0 frame:0
          TX packets:46 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5978 TX bytes:4545
          Interrupt:151

eth1      Link encap:Ethernet  HWaddr ec:21:e5:11:4f:ea  Driver tc956x_pci-eth
          inet addr:192.168.8.144  Bcast:192.168.8.255  Mask:255.255.255.0
          inet6 addr: fe80::5fcd:3eef:2ed5:fa32/64 Scope: Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:58 errors:0 dropped:0 overruns:0 frame:0
          TX packets:60 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6837 TX bytes:6925
          Interrupt:152
```

Figure 3-33. Ethernet Info.

Step 2. Perform a ping test.

```
kalama:/ # ping www.google.com
PING nm.aicdn.com (183.131.200.72) 56(84) bytes of data.
64 bytes from 183.131.200.72: icmp_seq=1 ttl=49 time=30.8 ms
64 bytes from 183.131.200.72: icmp_seq=2 ttl=49 time=30.1 ms
64 bytes from 183.131.200.72: icmp_seq=3 ttl=49 time=30.8 ms
64 bytes from 183.131.200.72: icmp_seq=4 ttl=49 time=30.6 ms
^C
--- nm.aicdn.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3115ms
rtt min/avg/max/mdev = 30.183/30.620/30.834/0.315 ms
```

3.10.3. Operation procedure on LU

➲ **NOTE**: This function has been verified in all software versions starting from **LU1.0.FC.r001002**.

Step 1. Confirm the connection setup with an IP obtained.

```
$ adb root
$ adb shell
# ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 10.9.5.49  netmask 255.255.255.0  broadcast 10.9.5.255
      inet6 fe80::846f:3bd3:2588:65c2  prefixlen 64  scopeid 0x20<link>
      ether ec:21:e5:10:4f:ea  txqueuelen 1000  (Ethernet)
      RX packets 12  bytes 3552 (3.5 KB)
      RX errors 0  dropped 0  overruns 0  frame 0
      TX packets 18  bytes 1807 (1.8 KB)
      TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
      device interrupt 100

# ifconfig eth1
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 10.9.5.57  netmask 255.255.255.0  broadcast 10.9.5.255
      inet6 fe80::c607:9640:b46b:6164  prefixlen 64  scopeid 0x20<link>
      ether ec:21:e5:11:4f:ea  txqueuelen 1000  (Ethernet)
      RX packets 14  bytes 1642 (1.6 KB)
      RX errors 0  dropped 0  overruns 0  frame 0
      TX packets 17  bytes 1737 (1.7 KB)
      TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
      device interrupt 101
```

Step 2. Perform a ping test.

```
# ping www.google.com
PING www.google.com.w.kunlungr.com (124.165.125.114): 56 data bytes
64 bytes from 124.165.125.114: icmp_seq=0 ttl=47 time=19.574 ms
64 bytes from 124.165.125.114: icmp_seq=1 ttl=47 time=18.912 ms
64 bytes from 124.165.125.114: icmp_seq=2 ttl=47 time=18.613 ms
```

3.10.4. Operation procedure on LE

➲ **NOTE**: This function has been verified in all software versions starting from the **SQC6490.LE1.0.ES.r000001**.

Step 1. Confirm the connection setup with an IP obtained.

```
$ adb root
$ adb shell
# ifconfig enP1p5s0f0
enP1p5s0f0 Link encap:Ethernet  HWaddr EC:21:E5:10:4F:EA
        inet addr:10.9.5.37  Bcast:10.9.5.255  Mask:255.255.255.0
        inet6 addr: fe80::ee21:e5ff:fe10:4fea/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:76 errors:0 dropped:29 overruns:0 frame:0
        TX packets:55 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:10261 (10.0 KiB)  TX bytes:6613 (6.4 KiB)
        Interrupt:247

# ifconfig enP1p5s0f1
enP1p5s0f1 Link encap:Ethernet  HWaddr EC:21:E5:11:4F:EA
        inet addr:10.9.5.38  Bcast:10.9.5.255  Mask:255.255.255.0
        inet6 addr: fe80::ee21:e5ff:fe11:4fea/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:89 errors:0 dropped:46 overruns:0 frame:0
        TX packets:36 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:12043 (11.7 KiB)  TX bytes:5346 (5.2 KiB)
        Interrupt:248
```

Step 2. Perform a ping test.

```
# ping www.google.com
PING www.google.com (120.201.235.99): 56 data bytes
64 bytes from 120.201.235.99: seq=0 ttl=54 time=11.431 ms
64 bytes from 120.201.235.99: seq=1 ttl=54 time=11.998 ms
64 bytes from 120.201.235.99: seq=2 ttl=54 time=11.495 ms
```

3.11. Audio interfaces

3.11.1. DMICs

3.11.1.1. Hardware configuration

Connect DMICs to the board via the **DMIC connectors** (No.5 in Figure 1-1).

Figure 3-34. Connect DMICs

➲ **NOTE**: To realize this function, a customized digital microphone (see below) is required. This board is not sold separately, but you can apply for the relevant reference design document and design the board on your own per actual needs. For more information, please contact us at service@sapphirestreamtech.com.

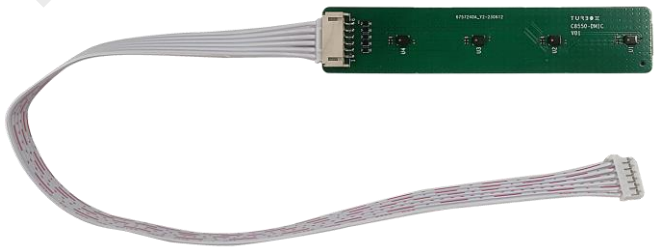


Figure 3-35. Digital Microphone

3.11.1.2. Operation procedure on LA

➲ **NOTE**: This function has been verified in all software versions starting from **LA3.0.FC.r001002**.

Step 1. Boot up your system.

Step 2. Click **Sounder Recorder** to open the recording interface.

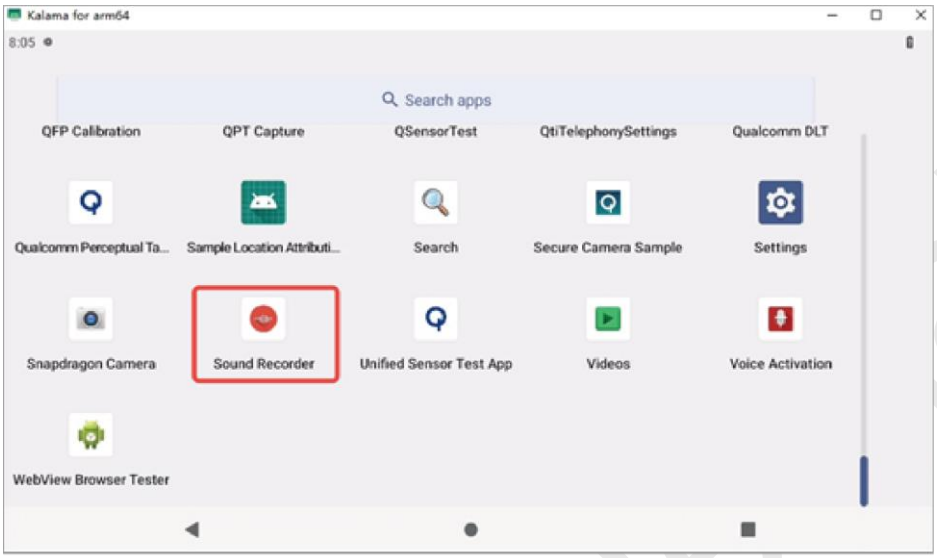


Figure 3-36. Sound Recorder

Step 3. Click to start recording.

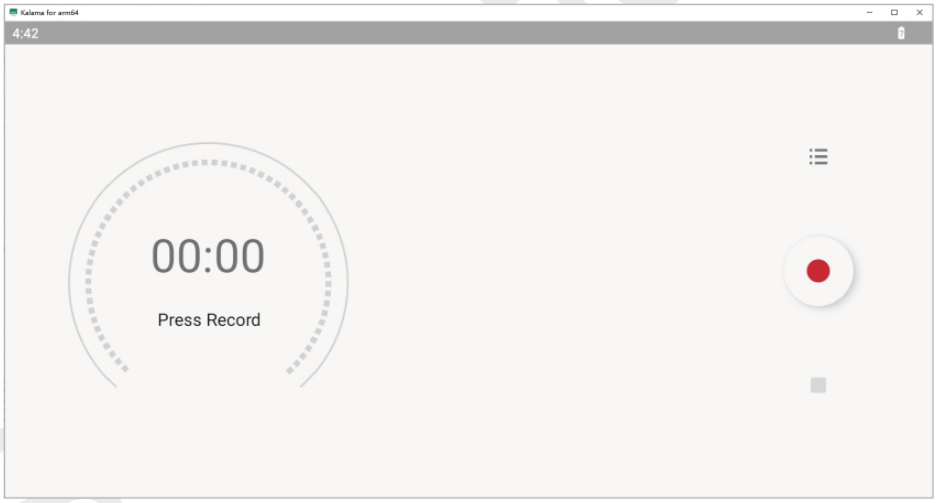


Figure 3-37. Start Recording

Step 4.Click ■ to stop recording.

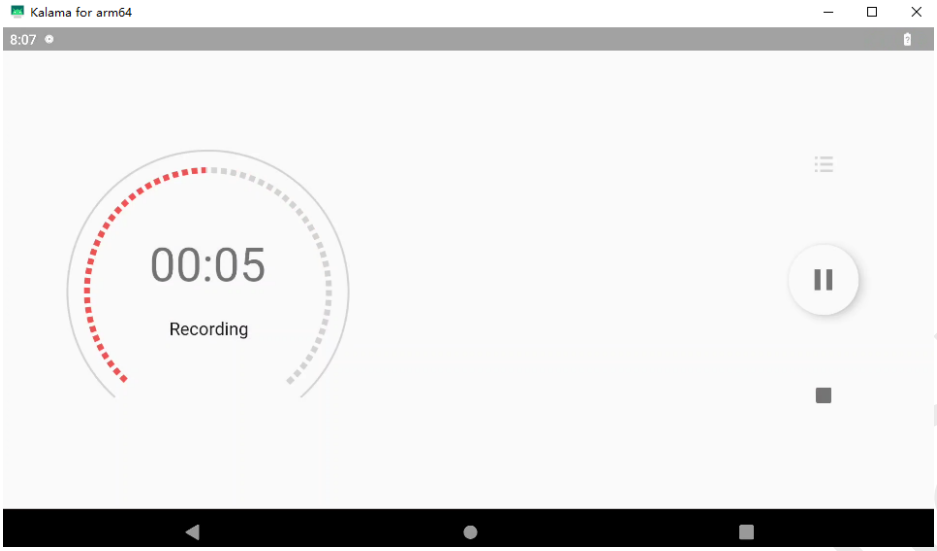


Figure 3-38.Sound Recording

Step 5.Save and play the recording file.

3.11.1.3.Operation procedure on LU

Enter the following commands to capture via DMICs.

- Capture under ***hal_rec_test_64bit*** via DMICs.

```
$ adb root
$ adb shell
/ # hal_rec_test_64bit -r 48000 -t 30 -c 2
```

Pull PCM clip from */data/audio/*.

```
$ adb pull /data/audio/rec1.wav
```

- Capture under ***parecord*** via DMICs.

```
$ adb root
$ adb shell
/ # parec -v --rate=48000 --format=s16le --channels=2 --file-format=wav /data/track.wav
```

Pull PCM clip from */data/*.

```
$ adb pull /data/track.wav
```

- Capture under ***gstreamer*** via DMICs.

➲ **NOTE: B**efore executing the following commands, make sure that you have not run `systemctl stop pulseaudio`; otherwise, you will need to execute `systemctl start pulseaudio` or restart your device.

```
$ adb root
$ adb shell
/ # gst-launch-1.0 -v pulsesrc ! audio/x-raw,format=S16LE,rate=48000,channels=2 !
audioconvert ! wavenc ! filesink location=/data/track.wav
```

Pull PCM clip from */data/*.

```
$ adb pull /data/track.wav
```

- Capture under ***tinyalsa*** via DMICs.

  Step 1. Execute the commands below to capture via each DMIC respectively.

  - Capture via DMIC0/DMIC1.

```
$ adb root
$ adb shell
# tinymix set "MultiMedia1 Mixer TX_CDC_DMA_TX_3" "1"
# tinymix set "TX_CDC_DMA_TX_3 Channels" "One"
# tinymix set "TX_AIF1_CAP Mixer DEC0" "1"
# tinymix set "TX DMIC MUX0" "DMIC0"
# tinycap /data/dmic0.wav -D 0 -d 0
```

    Disable the audio path.

```
# tinymix set "MultiMedia1 Mixer TX_CDC_DMA_TX_3" "0"
# tinymix set "TX_CDC_DMA_TX_3 Channels" "0"
# tinymix set "TX_AIF1_CAP Mixer DEC0" "0"
# tinymix set "TX DMIC MUX0" "0"
```

  - Capture via DMIC2/DMIC3.

```
$ adb root
$ adb shell
# tinymix set "MultiMedia1 Mixer TX_CDC_DMA_TX_3" "1"
# tinymix set "TX_CDC_DMA_TX_3 Channels" "One"
# tinymix set "TX_AIF1_CAP Mixer DEC2" "1"
# tinymix set "TX DMIC MUX2" "DMIC2"
# tinycap /data/dmic2.wav -D 0 -d 0
```

    Disable the audio path.

```
# tinymix set "MultiMedia1 Mixer TX_CDC_DMA_TX_3" "0"
# tinymix set "TX_CDC_DMA_TX_3 Channels" "0"
# tinymix set "TX_AIF1_CAP Mixer DEC2" "0"
# tinymix set "TX DMIC MUX2" "0"
```

  - Capture via DMIC4/DMIC5.

```
$ adb root
$ adb shell
# tinymix set "MultiMedia1 Mixer TX_CDC_DMA_TX_3" "1"
# tinymix set "TX_CDC_DMA_TX_3 Channels" "One"
# tinymix set "TX_AIF1_CAP Mixer DEC4" "1"
# tinymix set "TX DMIC MUX4" "DMIC4"
# tinycap /data/dmic4.wav -D 0 -d 0
```

    Disable the audio path.

```
# tinymix set "MultiMedia1 Mixer TX_CDC_DMA_TX_3" "0"
# tinymix set "TX_CDC_DMA_TX_3 Channels" "0"
# tinymix set "TX_AIF1_CAP Mixer DEC4" "0"
# tinymix set "TX DMIC MUX4" "0"
```

  Step 2. Pull PCM clip from /data/.

```
$ adb pull /data/dmicx.wav
```

3.11.1.4. Operation procedure on LE

This function is currently unavailable in the latest SDK version SQC6490.LE1.0.ES.r000001.

3.11.2. Speaker

3.11.2.1. Hardware configuration

Refer to the figure below to connect a speaker to the board via the **Speaker connector** (No.1 in Figure 1-1).

➲ **NOTE:** Two speaker connectors are available on SQC6490, but one of the them has no I2S data line connected in its hardware. As a result, only one speaker connector (SPK2) is functional.

Figure 3-39. Connect Speaker

3.11.2.2. Operation procedure on LA

➲ **NOTE**: This function has been verified in all software versions starting from **LA3.0.FC.r001002**.

Step 1. Push a wav file to your device.

```
$ adb root
$ adb push Mojito.wav /storage/emulated/0/Music
```

Step 2. Click on “Files”.

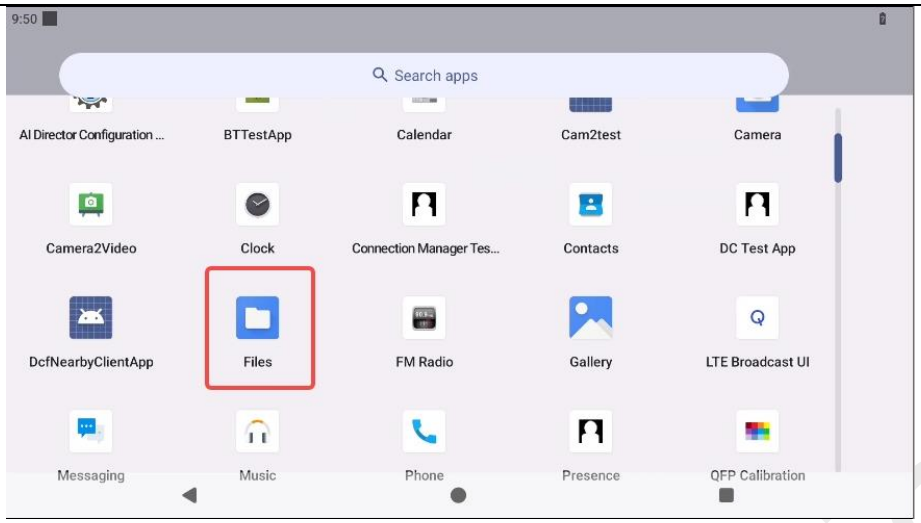


Figure 3-40. Open "Files"

Step 3. Enter the "Music" interface and play the *Mojito.wav* file, then voice will come out from the speaker.

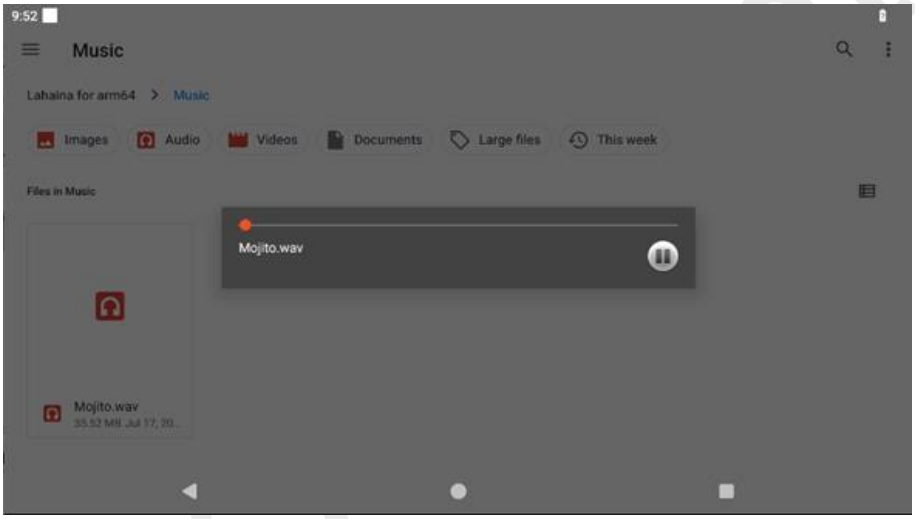


Figure 3-41. Play Wav File

3.11.2.3. Operation procedure on LU

Enter the following commands to play back the audio file via the speaker.

- Playback under ***hal_play_test_64bit*** via the speaker:

```
$ adb root
$ adb shell
/ # hal_play_test_64bit -f /data/audio/rec1.wav -v 1.0
```

- Playback under paplay via the speaker:

```
$ adb root
$ adb shell
/ # paplay -v /data/track.wav
```

- Playback under gstreamer via the speaker:

```
$ adb root
$ adb shell
/ # gst-launch-1.0 filesrc location=/data/track.wav ! wavparse ! audioconvert !
pulsesink volume=0.2
```

- Playback under tinyalsa via the speaker:

```
$ adb root
$ adb shell
/ # tinymix set "SEC_MI2S_RX Audio Mixer MultiMedia1" 1
/ # tinyplay /data/dmic0.wav
/ # tinyplay /data/dmic2.wav
/ # tinyplay /data/dmic4.wav
```

Disable the audio path.

```
/ # tinymix set "SEC_MI2S_RX Audio Mixer MultiMedia1" 0
```

3.11.2.4. Operation procedure on LE

This function is currently unavailable in the latest SDK version SQC6490.LE1.0.ES.r000001.

3.11.3. Headphone

3.11.3.1. Hardware configuration

Refer to the figure below to locate the Headphone connector (No.33 in Figure 1-1).

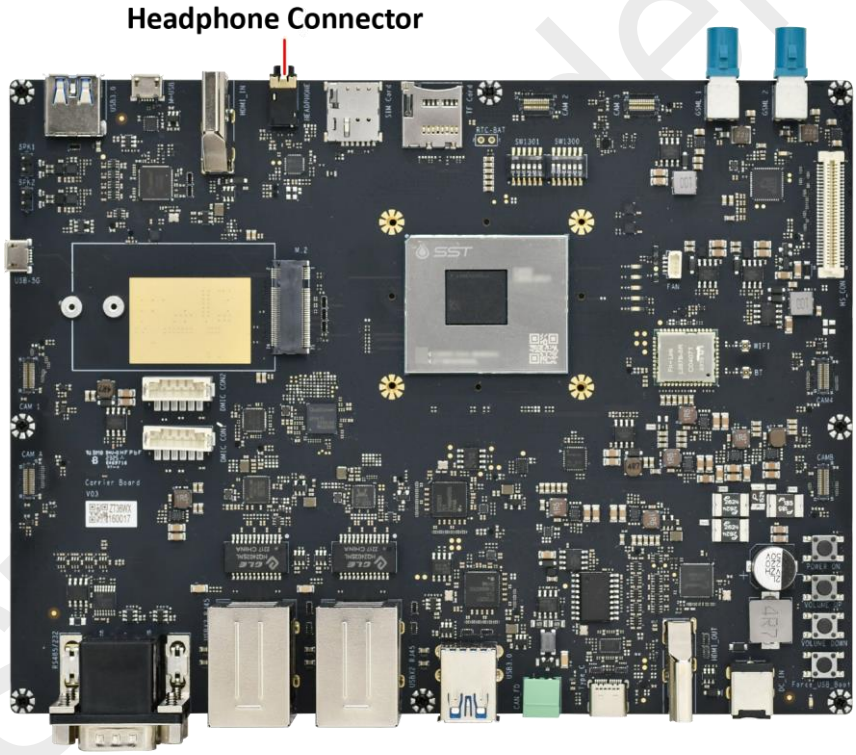


Figure 3-42. Locate the Headphone Connector

3.11.3.2. Operation procedure on LA

➲ **NOTE**: This function has been verified in all software versions starting from **LA3.0.FC.r001002**.

Step 1. Push a wav file to your device.

```
$ adb root
$ adb push Mojito.wav /storage/emulated/0/Music
```

Step 2. Play the *Mojito.wav* file, and the voice will come out from the headphone.

```
$ adb shell
lahaina:/ # tinymix "PRI_MI2S_RX Audio Mixer MultiMedia1" 1
lahaina:/ # tinyplay /data/Mojito.wav
```

Step 3. Record with the headphone.

```
$ adb shell
lahaina:/ # tinymix "MultiMedia1 Mixer PRI_MI2S_TX" 1
lahaina:/ # tinycap /data/my.wav
```

Step 4. Play the recording to verify whether the microphone on the headphone works.

```
$ adb shell
lahaina:/ # tinyplay /data/my.wav
```

3.11.3.3. Operation procedure on LU

This function is currently unavailable in the latest SDK version SQC6490 LU1.0.r001002.

3.11.3.4. Operation procedure on LE

This function is currently unavailable in the latest SDK version SQC6490.LE1.0.ES.r000001.

3.12. Camera

3.12.1. Camera module connectors

3.12.1.1. Hardware configuration

Step 1. Set the DIP switch. Refer to Table 1-3.

- SW1300: Toggle 1, 2 and 4 to ON.

Step 2. Refer to the figure below to connect IMX577 camera modules to the board via the **Camera module connectors** (No.4, No.22, and No.28 in Figure 1-1).

Step 3. Refer to the figure below to connect GMSL camera modules to the board via the **Camera module connectors** (No.26 and No.27 in Figure 1-1).

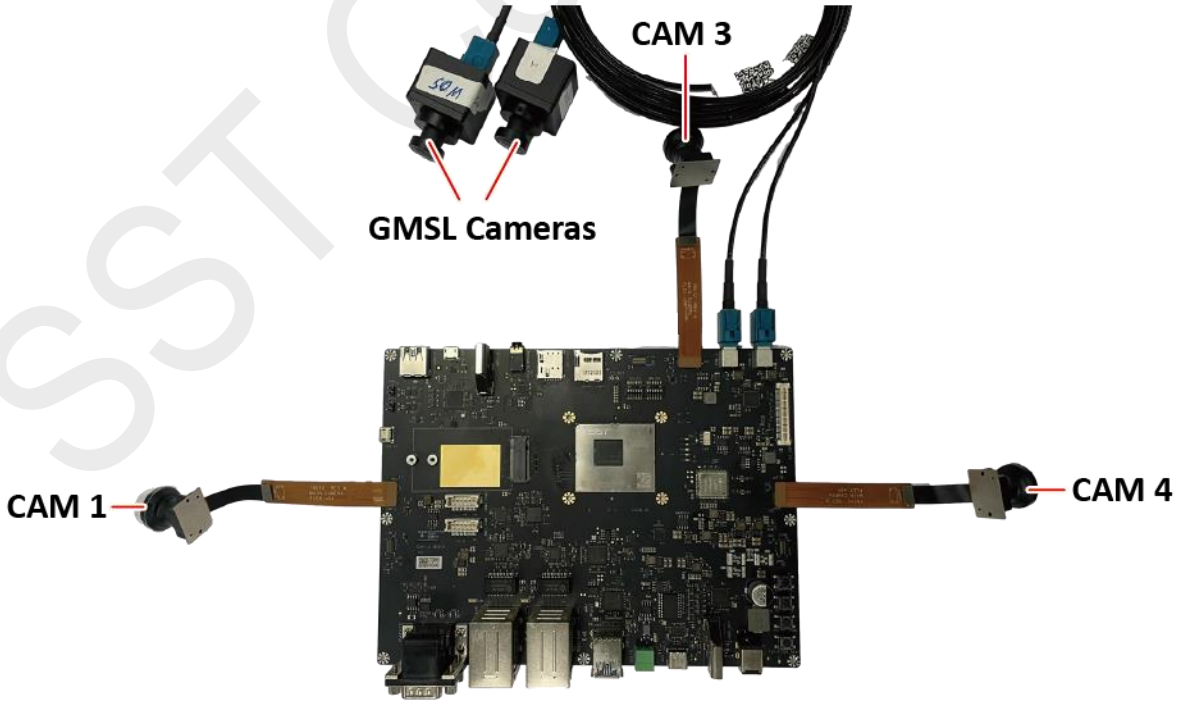


Figure 3-43. Camera Module Connection

3.12.1.2. Operation procedure on LA

➲ **NOTES:**

- This function has been verified in all software versions starting from **LA3.0.FC.r001002**.
- The video capture by GMSL camera is currently unavailable in the latest SDK version **LA3.0.FC.r001002**.

Step 1. Boot up your system and open the **Snapdragon Camera** app to take pictures or videos .

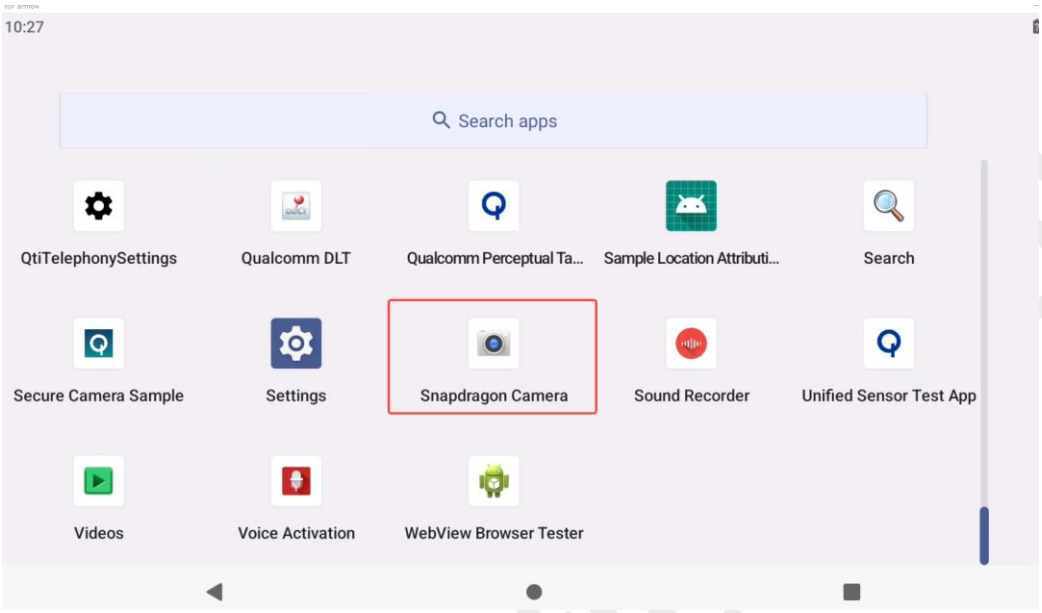


Figure 3-44. Home Page of Snapdragon Camera

➲ **NOTE:** Default rear camera (IMX577): CAM1.

To switch the camera, you can follow the steps below.

1) Click on the settings icon located at the top right corner of the preview screen.

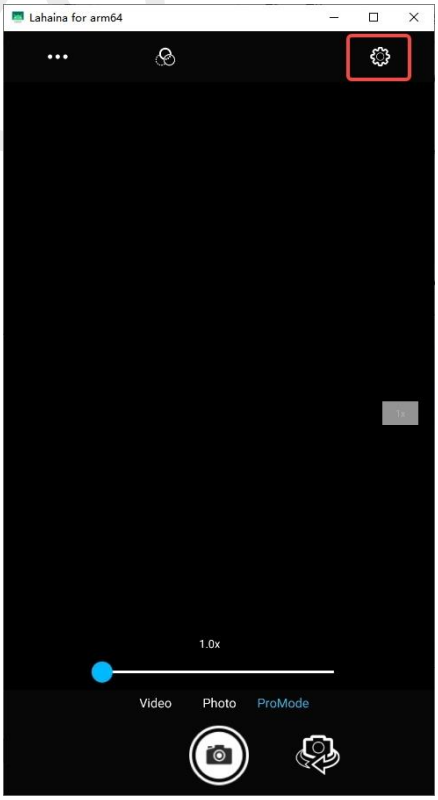


Figure 3-45. Click on the Settings Icon

2) Scroll down to the bottom of the screen and repeatedly click on "Version Info" until the interface returns to the preview screen. Then click on the settings icon again.

Figure 3-46. Return to Preview Screen

3) Scroll down and locate the "Switch camera by id" option, then click on it to switch the camera.

Step 2. After taking a picture or a video, you can view it in the Gallery.

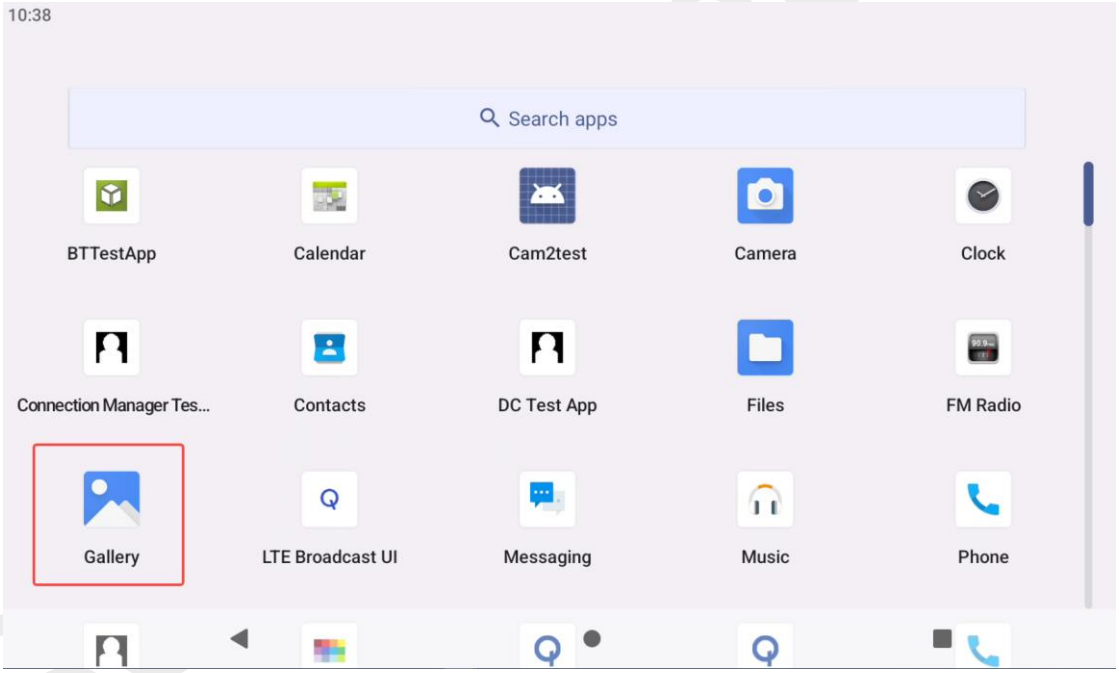


Figure 3-47. View the Picture in Gallery

3.12.1.3. Operation procedure on LU

➲ **NOTE**: This function has been verified in all software versions starting from **LU1.0.FC.r001002**.

3.12.1.3.1. Video recording

Step 1. Boot up your system.

Step 2. Enter the following commands to record video file via IMX577 camera module.

```
$ adb root
$ adb shell
# mkdir -p /data/output/
# gst-launch-1.0 -e qtiqmmfsrc camera=1 name=qmmf ! video/x-
raw,format=NV12,width=3840,height=2160,framerate=30/1 ! qtic2venc ! h264parse !
mp4mux ! queue ! filesink location=/data/output/cam1_mux.mp4
```

➲ **NOTE:** To use the GMSL camera for video recording, set the camera parameter to 0 or 2; the maximum resolution is 1920 x 1080.

Step 3. Pull the capture files from the */data/output/* folder to PC, and check the capture files.

```
$ adb pull /data/output/cam1_mux.mp4
```

3.12.1.3.2. Camera preview

Step 1. Connect an HDMI cable to the board via the **HDMI OUT connector** (see also 3.6.2. HDMI OUT).

Step 2. Enter the following commands and camera preview can be displayed on Wayland.

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

➲ **NOTE:** To use the GMSL camera for video recording, set the camera parameter to 0 or 2; the maximum resolution is 1920 x 1080.

Step 3. Check the camera preview on the monitor via the **HDMI OUT connector**.

3.12.1.3.3. Camera preview/recording/snapshot

Step 1. Connect an HDMI cable to the board via the **HDMI OUT connector** (see also 3.6.2. HDMI OUT).

Step 2. Enter the following commands to test the live snapshot.

```
$ adb root
$ adb shell
# export GST_PLUGIN_PATH=/usr/lib/gstreamer-1.0 &
# export XDG_RUNTIME_DIR=/run/user/root &
# export XDG_RUNTIME_DIR=/run/user/root && gst-pipeline-app -e qtiqmmfsrc camera=1
name=camsrc ! video/x-raw,format=NV12,width=3840,height=2160,framerate=30/1 !
waylandsink fullscreen=true async=true sync=false camsrc.image_1 !
"image/jpeg,width=3840,height=2160,framerate=30/1" ! multifilesink
location=/data/output/frame%d.jpg sync=true async=false
```

➲ **NOTE:** To use the GMSL camera for video recording, set the camera parameter to 0 or 2; the maximum resolution is 1920 x 1080.

Step 3. Upon entering the above commands, a menu will be displayed.

```
sh-5.0# gst-pipeline-app -e qtiqmmfsrc camera=1 name=camsrc ! video/x-raw,format
=1920,height=1080,framerate=30/1" ! multifilesink location=/data/output/camera0%
p4mux ! queue ! filesink location=/data/output/camera_0.mp4
gbm_create_device(192): Info: backend name is: msm_drm


################################### MENU #####################################

 ============================== Pipeline Controls ==============================
   (0) NULL                    : Set the pipeline into NULL state
   (1) READY                   : Set the pipeline into READY state
   (2) PAUSED                  : Set the pipeline into PAUSED state
   (3) PLAYING                 : Set the pipeline into PLAYING state
 ================================== Other ====================================
   (p) Plugin Mode             : Choose a plugin which to control
   (q) Quit                    : Exit the application


Choose an option:
```

Figure 3-48. Menu Display

Step 4. Follow the sub-steps below.

1) Select the “PLAYING” option index: **3**.
2) Select the “Plugin Mode” option index: **p**.
3) Select the “camsrc” option index: **13**.
4) Select “capture-image” option index: **35**.
5) Enter the “GstImageCaptureMode” value for arg0: **0**.
6) Enter the “guint” value for arg1: **1**.
7) Enter “**b**” to return to the menu: **b**.
8) Enter “**q**” to exit: **q**.

Step 5. Check the camera preview on the monitor via HDMI OUT.

Step 6. Pull the capture files from the */data/output/* folder to PC, and then check them.

```
$ adb pull /data/output/frame0.jpg
```

3.12.1.3.4. IMX577 (x 3) camera recording concurrency

Step 1. Open three terminals and enter the following commands to record the video file via the IMX577 camera module.

- **Terminal 1:**

```
$ adb root
$ adb shell
# mkdir -p /data/output
# export GST_PLUGIN_PATH=/usr/lib/gstreamer-1.0 &
# export XDG_RUNTIME_DIR=/run/user/root &
# gst-launch-1.0 -e qtiqmmfsrc camera=1 name=qmmf ! video/x-raw,format=NV12,width=192
0,height=1080,framerate=30/1 ! qtic2venc ! h264parse ! mp4mux ! queue ! filesink locat
ion=/data/output/cam1_mux.mp4
```

- **Terminal 2:**

```
$ adb shell
# gst-launch-1.0 -e qtiqmmfsrc camera=3 name=qmmf ! video/x-raw,format=NV12,width=192
0,height=1080,framerate=30/1 ! qtic2venc ! h264parse ! mp4mux ! queue ! filesink locat
ion=/data/output/cam3_mux.mp4
```

- **Terminal 3:**

```
$ adb shell
# gst-launch-1.0 -e qtiqmmfsrc camera=4 name=qmmf ! video/x-raw,format=NV12,width=192
0,height=1080,framerate=30/1 ! qtic2venc ! h264parse ! mp4mux ! queue ! filesink locat
ion=/data/output/cam4_mux.mp4
```

➲ **NOTE:** To use the GMSL camera for video recording, set the camera parameter to 0 or 2; the maximum resolution is 1920 x 1080.

Step 2. Pull the capture files from the */data/output/* folder to PC, and then check them.

```
$ adb pull /data/output/cam1_mux.mp4
$ adb pull /data/output/cam3_mux.mp4
$ adb pull /data/output/cam4_mux.mp4
```

3.12.1.4. Operation procedure on LE

This function has been verified in all software versions starting from **SQC6490.LE1.0.ES.r000002**.

3.12.1.4.1. Video recording

Step 1. Boot up your system.

Step 2. Enter the following commands to record video file via IMX577 camera module.

```
$ adb root
$ adb shell
# mount -o rw,remount /
# mkdir -p /data/output
# echo multiCameraLogicalXMLFile=kodiak_dc.xml >>
/var/cache/camera/camxoverridesettings.txt
# gst-launch-1.0 -e qtiqmmfsrc camera=0 name=camsrc video_0::type=preview ! video/x-
raw\(memory:GBM\),format=NV12,width=3840,height=2160,framerate=30/1,compression=ubwc
,interlace-mode=progressive,colorimetry=bt601 ! queue ! v4l2h264enc capture-io-
mode=5 output-io-mode=5 ! queue ! h264parse ! mp4mux ! queue ! filesink
location="/data/output/mux0.mp4"
```

Step 3. Pull the capture files from the /data/output/ folder to PC, and check the capture files.

```
$ adb pull /data/output/mux0.mp4
```

3.12.1.4.2. Camera preview

Step 1. Connect an HDMI cable to the board via the **HDMI OUT connector** (see also 3.6.2. HDMI OUT).

Step 2. Enter the following commands and camera preview can be displayed on Wayland.

```
$ adb root
$ adb shell
# mount -o rw,remount /
# echo multiCameraLogicalXMLFile=kodiak_dc.xml >>
/var/cache/camera/camxoverridesettings.txt
# export XDG_RUNTIME_DIR=/dev/socket/weston
# export WAYLAND_DISPLAY=wayland-1
# gst-launch-1.0 -e qtiqmmfsrc camera=0 name=camsrc ! video/x-
raw\(memory:GBM\),format=NV12,width=3840,height=2160,framerate=30/1,compression=ubwc
! queue ! tee name=split split. ! queue ! qtivcomposer name=mixer
sink_1::dimensions="<1920,1080>" sink_1::alpha=0.5 ! queue ! waylandsink
fullscreen=true
```

Step 3. Check the camera preview on the monitor via the **HDMI OUT connector**.

3.12.1.4.3. Camera preview/recording/snapshot

Step 1. Connect an HDMI cable to the board via the **HDMI OUT connector** (see also 3.6.2. HDMI OUT).

Step 2. Enter the following commands to test live snapshot.

```
$ adb root
$ adb shell
# mount -o rw,remount /
# mkdir -p /data/output
# echo multiCameraLogicalXMLFile=kodiak_dc.xml >>
/var/cache/camera/camxoverridesettings.txt
# export XDG_RUNTIME_DIR=/dev/socket/weston
# export WAYLAND_DISPLAY=wayland-1
# gst-launch-1.0 -e qtiqmmfsrc camera=0 name=camsrc ! video/x-
raw\(memory:GBM\),format=NV12,width=3840,height=2160,framerate=30/1,compression=ubwc
! queue ! tee name=split split. ! queue ! qtivcomposer name=mixer
sink_1::dimensions="<1920,1080>" sink_1::alpha=0.5 ! queue ! waylandsink
fullscreen=true camsrc. ! image/jpeg,width=3840,height=2160,framerate=30/1 !
multifilesink location=/data/output/frame_0%d.jpg sync=true async=false max-files=5
```

Step 3. Pull the capture files from the */data/output/* folder to PC, and then check them.

```
$ adb pull /data/output/
```

3.12.1.4.4. IMX577 (x 2) camera recording concurrency

Step 3. Open three terminals and enter the following commands to record the video file via the IMX577 camera module.

- **Terminal 1:**

```
$ adb root
$ adb shell
# mount -o rw,remount /
# mkdir -p /data/output
# echo multiCameraLogicalXMLFile=kodiak_dc.xml >>
/var/cache/camera/camxoverridesettings.txt
# gst-launch-1.0 -e qtiqmmfsrc camera=0 name=camsrc video_0::type=preview !
video/x-
raw\(memory:GBM\),format=NV12,width=3840,height=2160,framerate=30/1,compression=ub
wc,interlace-mode=progressive,colorimetry=bt601 ! queue ! v4l2h264enc capture-io-
mode=5 output-io-mode=5 ! queue ! h264parse ! mp4mux ! queue ! filesink
location="/data/output/mux0.mp4"
```

- **Terminal 2:**

```
$ adb shell
# gst-launch-1.0 -e qtiqmmfsrc camera=1 name=camsrc video_0::type=preview !
video/x-
raw\(memory:GBM\),format=NV12,width=3840,height=2160,framerate=30/1,compression=ub
wc,interlace-mode=progressive,colorimetry=bt601 ! queue ! v4l2h264enc capture-io-
mode=5 output-io-mode=5 ! queue ! h264parse ! mp4mux ! queue ! filesink
location="/data/output/mux1.mp4"
```

Step 4. Pull the capture files from the */data/output/* folder to PC, and then check them.

```
$ adb pull /data/output/mux0.mp4
$ adb pull /data/output/mux1.mp4
```

3.12.2. HDMI IN

3.12.2.1. Hardware configuration

Step 1. Power off the device by disconnecting the power cable.

Step 2. Disconnect all cameras from the board.

Step 3. Connect the **HDMI IN connector** (No. 34 in Figure 1-1) of the board to the HDMI OUT port of PC with an HDMI cable.

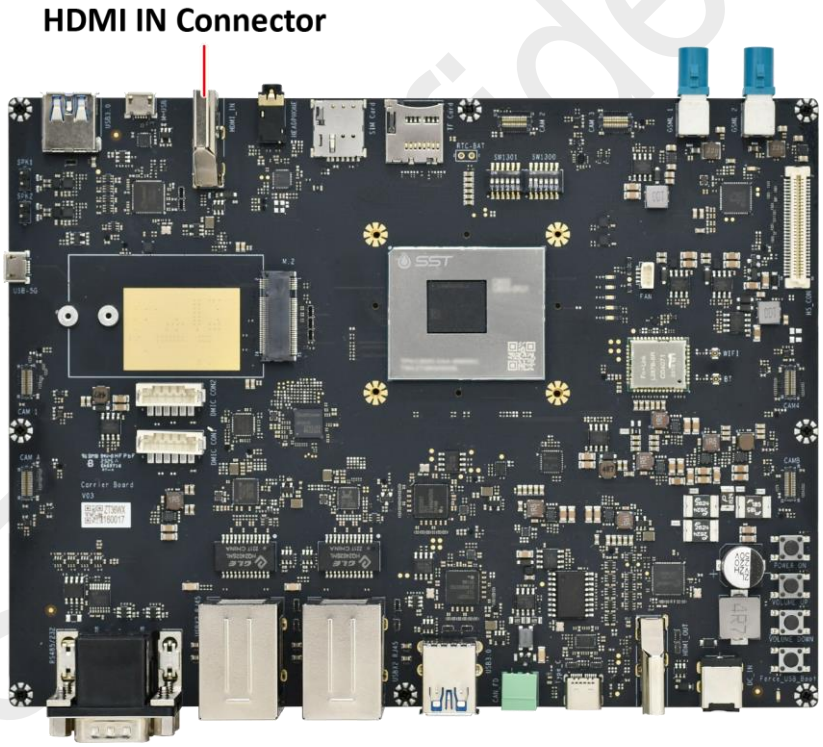


Figure 3-49. HDMI IN Connector

Step 4. Connect a display to the board via the **HDMI OUT connector** (No. 13 in Figure 1-1) with an HDMI cable.

Step 5. Set the DIP switch. Refer to Table 1-3 for details.

- SW1300: Toggle 1,2 to ON and 4 to OFF.

Step 6. Connect the power supply to the board and press the **Power on button** (No. 18 in Figure 1-1) to boot up the system.

3.12.2.2. Operation procedure on LA

➲ **NOTE:** This function has been verified in all software versions starting from **LA3.0.FC.r001002**.

Step 1. Install the **CameraRaw** app.

```
$ adb install -t CameraRaw.apk
```

Step 2. Disable dm-verity checking on userdebug builds.

```
$ adb root
$ adb disable-verity
$ adb reboot
```

Step 3. Modify GMSL related files.

```
$ adb root
$ adb shell mount -o rw,remount /vendor
$ adb shell
kalama:/ # mkdir /vendor/lib64/camera/bak
kalama:/ # mv /vendor/lib64/camera/com.qti.sensormodule.max9296a_ar0231_gmsl0.bin
/vendor/lib64/camera/bak
kalama:/ # mv /vendor/lib64/camera/com.qti.sensormodule.max9296a_ar0231_gmsl1.bin
/vendor/lib64/camera/bak
kalama:/ # sync
```

Step 4. Reboot the device.

```
kalama:/ # exit
$ adb reboot
```

Step 5. Open the **CameraRaw** app to preview the HDMI IN image.

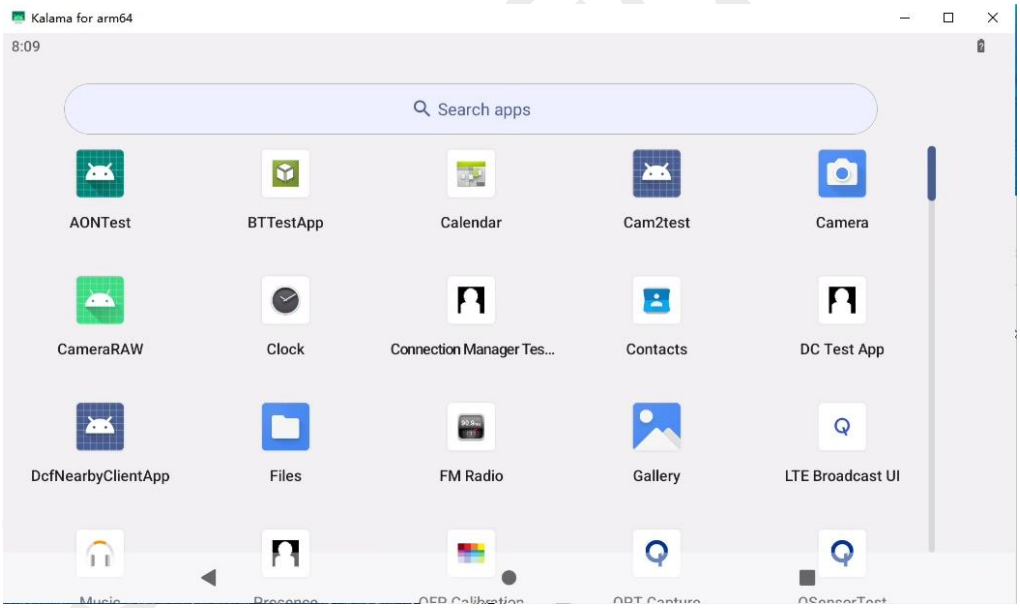


Figure 3-50. CameraRaw

Refer to the following HDMI IN display:

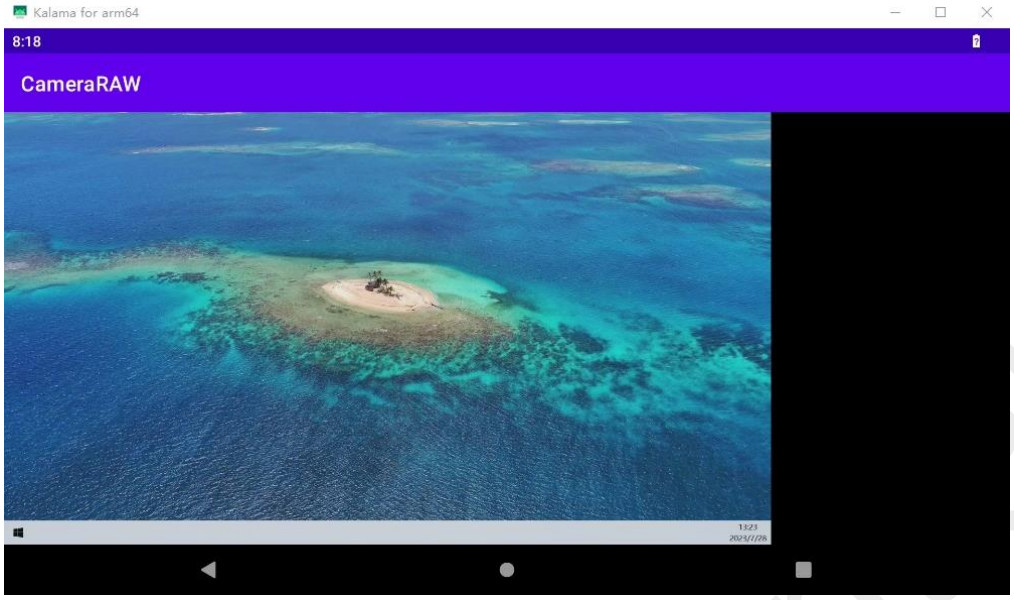


Figure 3-51. HDMI IN Display Preview

➲ **NOTE**: After testing HDMI IN, recover the files in the bak folder.

3.12.2.3. Operation procedure on LU

This function is currently unavailable in the latest SDK version SQC6490 LU1.0.r001002.

3.12.2.4. Operation procedure on LE

This function is currently unavailable in the latest SDK version SQC6490.LE1.0.ES.r000001.

3.13. CAN

3.13.1. Hardware configuration

Refer to the figure below to connect a CAN device to the board via the **CAN connector** (No.11 in Figure 1-1).

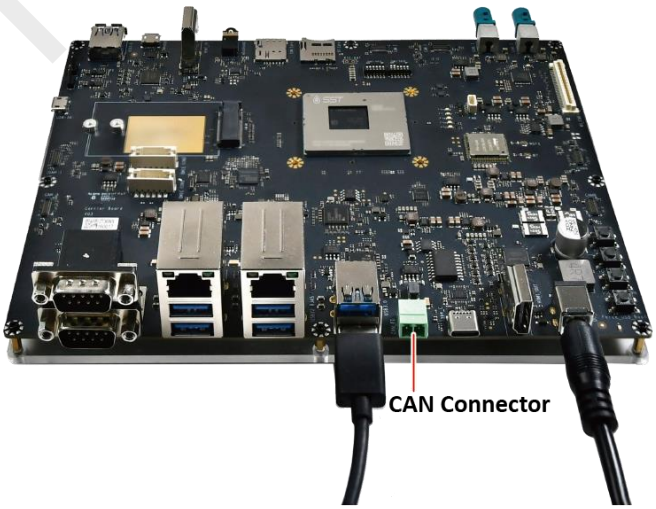


Figure 3-52. Connect CAN Transceiver to SQC6490 DK Board

➲ **NOTES**:

- To connect another board to the CAN connector of the SQC6490 DK board, please use a Dupont cable (female to female) for the connection.
- Make sure that the CAN connectors connected between the two boards are one-to-one corresponding. Specifically, the CAN connector near the USB 3.0 Type-A connector of one board should be connected to the CAN connector near the USB 3.0 Type-A connector of the other board, while the CAN connector near the USB 3.1 Type-C connector of one board should be connected to the CAN connector near the USB 3.1 Type-C connector of the other board.
- Obtain the test tools: cansend and candump.

3.13.2. Operation procedure on LA

➲ **NOTE**: This function has been verified in all software versions starting from **LA3.0.ES.r000001**.

Step 1. Search for the CAN devices of two boards.

```
$ adb root
$ adb remount
$ adb push candump /data/
$ adb push cansend /data/
$ adb shell
lahaina:/ # chmod a+x /data/candump
lahaina:/ # chmod a+x /data/cansend
lahaina:/ # ifconfig -a | grep can
can0 Link encap:UNSPEC Driver mcp251x
```

➲ **NOTE:** For software version LA3.0.FC.r001002, manual pushing of the candump and cansend files is no longer required. You may proceed with the verification process by skipping the related steps.

Step 2. Set **CAN0** for each device separately.

```
lahaina:/ # ip link set can0 down
lahaina:/ # ip link set can0 type can bitrate 100000 loopback off
lahaina:/ # ifconfig can0 up
```

Step 3. Send data on one of the devices.

```
lahaina:/ # /data/cansend can0 123#1122334455
```

Step 4. Receive data on the other device.

```
lahaina:/ # /data/candump can0
can0 123 [5] 11 22 33 44 55
```

3.13.3. Operation procedure on LU

This function is currently unavailable in the latest SDK version SQC6490 LU1.0.r001002.

3.13.4. Operation procedure on LE

This function is currently unavailable in the latest SDK version SQC6490.LE1.0.ES.r000001.

3.14. M.2

3.14.1. Hardware configuration

Refer to the figure below to connect NVME SSD (BM-KEY 2242 size) to the board via the **M.2 connector** (No.3 in Figure 1-1).

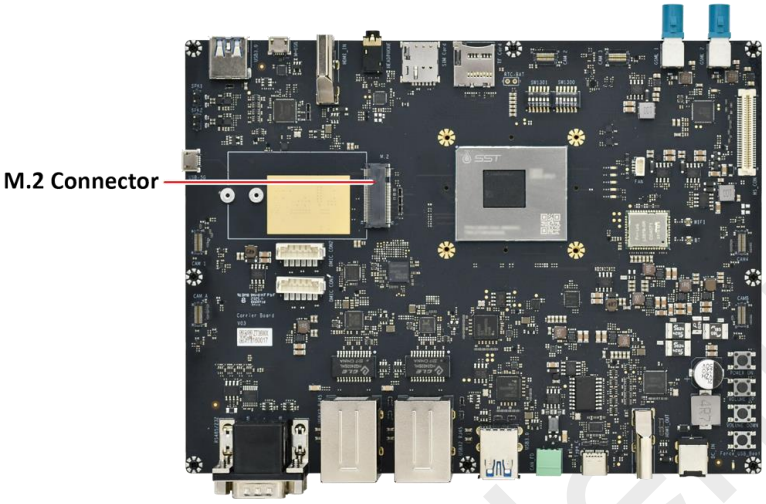


Figure 3-53. M.2 connector

3.14.2. Operation procedure on LA

➲ **NOTE**: This function has been verified in all software versions starting from **LA3.0.FC.r001002**.

Step 1. Install the test tool **tiotest** and push the tool to the device.

```
$ adb root
$ adb push tiotest /data/
$ adb shell chmod 755 /data/tiotest
```

Step 2. Run the following commands to test read and write speed of NVME SSD.

```
$ adb shell
lahaina:/ # cd /data/
lahaina:/ # mkdir nvme
lahaina:/ # mount /dev/block/nvme0n1 /data/nvme
lahaina:/ # ./tiotest -t 8 -d /data/nvme -f 1024 -k 1 -k 3 –L
Tiotest results for 8 concurrent io threads:
,---------------------------------------------------------------------------.
| Item | Time | Rate | IOPS | Usr CPU | Sys CPU |
+----------------------+----------+---------------+---------+----------+---------+
| Write 8192 MBs | 13.5 s | 607.903 MB/s | | 23.3 % | 380.5 % |
| Read 8192 MBs | 5.3 s | 1548.640 MB/s | | 46.3 % | 290.5 % |
+----------------------+----------+---------------+---------+----------+---------+
```

3.14.3. Operation procedure on LU and LE

➲ **NOTE**: This function has been verified in all software versions starting from the following:

- **LU1.0.FC.r001002**
- **SQC6490.LE1.0.ES.r000001**

Run the following commands to test the function.

```
$ adb root
$ adb shell
# mkdir -p /data/test
# mkfs.ext4 /dev/nvme0n1
/dev/nvme0n1 contains a ext4 file system
       last mounted on /data/nvme on Tue Nov 21 06:12:23 2023
Proceed anyway? (y,N) y
Discarding device blocks: done
Creating filesystem with 31258710 4k blocks and 7815168 inodes
Filesystem UUID: 7cf000de-16ac-4317-ae1b-4d0dd8c6c473
Superblock backups stored on blocks:
       32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
       4096000, 7962624, 11239424, 20480000, 23887872

Allocating group tables: done
Writing inode tables: done
Creating journal (131072 blocks): done
Writing superblocks and filesystem accounting information: done
# mount -t ext4 /dev/nvme0n1 /data/test
# df –h
Filesystem     Size  Used Avail Use% Mounted on
/dev/root       99G  7.7G   91G   8% /
devtmpfs       2.4G     0  2.4G   0% /dev
tmpfs          3.6G     0  3.6G   0% /dev/shm
tmpfs          727M  4.2M  723M   1% /run
tmpfs          5.0M  4.0K  5.0M   1% /run/lock
tmpfs          3.6G     0  3.6G   0% /sys/fs/cgroup
tmpfs          3.6G  8.0K  3.6G   1% /tmp
/dev/sde4      220M   61M  160M  28% /firmware
/dev/sde9       59M   31M   28M  53% /dsp
/dev/nvme0n1   117G   24K  111G   1% /data/test
```

➲ **NOTE:** On LE, replace the "`# df –h`" command with "`# df`".

Appendix 1. Notices

SST may have patents or pending patent programs covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries to service@sapphirestreamtech.com.

SST PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

Changes are made periodically to the information herein; these changes will be incorporated in new editions of the publication. To provide better service, SST reserves the right to improve and/or modify the products and software programs described in the manuals, and the content of the manual, at any time without additional notice.

The software interface and function and hardware configuration described in the manuals included with your development board or system on module might not match exactly the actual configuration of that you have purchased. For the configuration of the product, refer to the related contract (if any) or product packing list, or consult the distributor for the product sales. SST may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

The products described in this document are not intended for use in implantation or other life support applications where malfunction may result in injury or death to persons. The information contained in this document does not affect or change SST product specifications or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of SST or third parties. All information contained in this document was obtained in specific environments and is presented as an illustration. The result obtained in other operating environments may vary.

The information of this document should not be as any invitation for offer or any advice to the visitors. Please consult the professional comments from the sales consultant prior to do any actions of investment or purchase.

SST may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any references in this publication to non-SST websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this SST product, and use of those websites is at your own risk. SST shall not be responsible for the content of the third party.

Any performance data contained herein was determined in a controlled environment. Therefore, the result obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

This document is copyrighted by SST and the property right of the date mentioned in this document, including but not limited trademarks, patents, copyrights, trade name etc. are not covered by any open-source license. SST may update this document at any time without notice.

Anyone doesn’t have the right to amend, reprint, republication, reproduce, transmit, distribute or any other way to use this document in business or public purpose without the prior written consent by SST.

E-mail messages sent to SST via the Internet are not guaranteed to be completely secure. SST shall not be liable for any loss incurred by the surfer when transmitting any information over the Internet or for any loss incurred by SST when sending any information over the Internet at your request.

SST has all rights under other relevant exemptions provided by laws and regulations, and SST's failure to claim

or delay in claiming such rights shall not be deemed to be a waiver of such rights by SST.

SST reserves the right of final interpretation of this document.

Appendix 2. Trademarks

SST is the trademark of Sapphire Stream Technology Co., Ltd. or its associate companies. Intel, Intel SpeedStep, Optane, and Thunderbolt are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Microsoft, Windows, Direct3D, BitLocker, and Cortana are trademarks of the Microsoft group of companies. Mini DisplayPort (mDP), DisplayPort, and VESA are trademarks of the Video Electronics Standards Association. The terms HDMI and HDMI High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing LLC in the United States and other countries. Wi-Fi, Wi-Fi Alliance, WiGig, and Miracast are registered trademarks of Wi-Fi Alliance. USB-C is a registered trademark of USB Implementers Forum. All other trademarks are the property of their respective owners.