Edge AI Applied Computing

SINTRONES

IBOX-604-G2

User Manual

Version 1.0

NVIDIA JETSON

SINTRONES

Revision History

Version	Date	Description of Changes
1.0	2026-06-24	Initial release.

Copyright

Disclaimer

SINTRONES® Technology Corp. shall not be liable for any incidental or consequential damages resulting from the performance or use of this product.

SINTRONES® Technology Corp. makes no representation or warranty regarding the content of this manual. Information in this manual has been carefully checked for accuracy; however, no guarantee is given to the correctness of the contents. For continuing product improvement, SINTRONES® Technology Corp. reserves the right to revise the manual or make changes to the specifications of this product at any time without notice and obligation to any person or entity regarding such change. The information contained in this manual is provided for general use by customers.

Declaration of Conformity

The CE symbol on your product indicates that it complies with the European Union (EU) directives. A Certificate of Compliance is available by contacting Technical Support. This product has passed the CE test for environmental specifications when shielded cables are used for external wiring. We recommend the use of shielded cables.

This product has been tested and found to comply with the limits for a Class A device, according to Part 15 of the FCC Rules. These limits are designed to protect reasonably against harmful interference in a residential installation. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used by the manufacturer’s instructions, may cause harmful interference to radio communications.

Trademarks

SINTRONES® is a registered trademark of SINTRONES® Technology Corp.

The terms HDMI, HDMI High-Definition Multimedia Interface, HDMI Trade dress and the HDMI Logos are trademarks or registered trademarks of HDMI Licensing Administrator, Inc.

All the trademarks, registrations, and brands mentioned herein are used for identification purposes only and may be trademarks and registered trademarks of their respective owners.

Safety Information

Read the following precautions before setting up a SINTRONES® Product.

Electrical Safety

- Disconnect the power cable from the electrical outlet to prevent shock hazards before relocating the system.
- When adding or removing devices to or from the system, ensure that the power cables for the devices are unplugged before the signal cables are connected. Disrupt all power cables from the existing system before adding a device.
- Seek professional assistance before using an adapter or extension cord. These devices could interrupt the grounding circuit.
- Before connecting or removing signal cables from the motherboard, ensure all power cables are unplugged.
- Ensure your power supply is set to the correct voltage in your area. If you are unsure of the voltage of your current electrical outlets, contact your local power company.
- If the power supply is broken, do not fix it by yourself. Contact a qualified service technician or your retailer.

Operation Safety

- Before installing the motherboard and adding devices, carefully read all the manuals in the package.
- Before using the product, ensure all cables are correctly connected and the power cables are not damaged. If you detect any damage, contact your dealer immediately.
- Keep paper clips, screws, and staples away from connectors, slots, sockets, and circuitry to avoid short circuits.
- Avoid dust, humidity, and temperature extremes. Please do not place the product in any area that may become wet.
- Place the product on a stable surface.
- Contact a qualified service technician or retailer if you encounter technical problems with the product.

Environmental Safety

- Use this product in environments with ambient temperatures between -25˚C and 50ºC.
- Do not leave this product in an environment where the storage temperature may be below -40˚C or above 80˚C. To prevent damage, the product must be used in a controlled environment.

CAUTION:
Incorrectly replacing the battery may damage this computer. Replace only with the same or equivalent recommended by SINTRONES® Technology Corp. Dispose of the used battery according to the manufacturer's instructions.

Technical Support

Please call or e-mail our customer service when you cannot fix the problems.

- TEL: +886-2-8228-0101
- FAX: +886-2-8228-0100
- E-mail: sales@sintrones.com

SINTRONES

Contents

1. Introduction

Designed for maximum space efficiency, the IBOX-604-G2 can be seamlessly integrated into lithium battery systems or installed within Internal Combustion Engine Vehicles (ICEVs), while maintaining full operating capability. Additionally, the IBOX-604-G2 supports a wide input voltage range from DC 10V to 60V, making it ideal for 3S-14S lithium battery systems. This edge AI computer also features smart power management for ignition detection and safe shutdown, ensuring data integrity and extending system longevity.

Integrated with multiple carefully selected I/O interfaces, the IBOX-604-G2 is engineered for automation scenarios demanding high-performance AI, real-time image processing, and scalable connectivity. Powered by NVIDIA® Jetson Orin™ modules and equipped with two FAKRA Z connectors, this lightweight edge AI computer delivers up to 157 AI TOPS and supports automotive-grade GMSL-2 cameras for high-bandwidth video transmission and real-time data analytics.

Moreover, this compact AI system includes an M.2 3042/3052 Key B slot for WWAN or GNSS module, M.2 2230 Key E slot for WLAN or GNSS module, dual Ethernet ports (2.5GbE and GbE), a COM port (interface type varies by optional SKU), a single port supporting UART and CAN FD, and a built-in 3-axis accelerometer, ensuring a variety of connectivity across networks and sensors.

This chapter introduces SINTRONES® IBOX-604-G2 and gives an overview of its product details.

Topics in this chapter include:

1.1. Product Information

Table 1-1 Specifications

System

Module

- NVIDIA® Jetson Orin™ NX 16GB (1024 CUDA cores + 8-core ARM Cortex-A78AE CPU + 16 GB LPDDR5)
- NVIDIA® Jetson Orin™ NX 8GB (1024 CUDA cores + 6-core ARM Cortex-A78AE CPU + 8 GB LPDDR5)
- NVIDIA® Jetson Orin™ Nano 8GB (1024 CUDA cores + 6-core ARM Cortex-A78AE CPU + 8GB LPDDR5)

Sensor

- Built-in 3-axis accelerometer

Security

- Platform Security Controller (PSC), Security Engine (SE)

Watchdog

- Auto reset for unresponsive system

Interface

Video

- 1 x HDMI® Type-A output

Audio

- 1 x HD audio output via HDMI®

Ethernet

- 1 x 2.5GbE (Intel® I226-IT) via RJ-45
- 1 x GbE (NVIDIA® SoM) via RJ-45

Camera

- 2 x GMSL-2 via FAKRA

CAN

- 1 x CAN FD

USB

- 2 x USB 3.2

DIO (Optional)

- 1 x DI , 1 x DO , I²C (Customizable SKU)

GPIO

- 1 x UART (DC 3.3V, connector shared with CAN FD)

COM

- 1 x RS-232/422/485

Mgmt. Port

- 1 x USB Type-C for system recovery (device only)

Internal Expansion

M.2

- 1 x M.2 3042/3052 Key B for WWAN w/ dual SIM support
- 1 x M.2 2230 Key E for Wi-Fi/BT

Storage

Type

- 1 x M.2 2242 Key M for NVMe SSD

Power

Power Input

- DC 10-60V (nominal power input DC 11V/22V/58V) via 3-pin terminal block

Power Protection

- OCP, OVP, surge protection, and reversed polarity protection

Power Management

- Ignition detection, Smart Power Management

RTC Battery

- High-capacity coin cell battery for RTC

Software

Operating System

- NVIDIA® JetPack 6.2 or above (Jetson Linux and NVIDIA® development tools included)

Environmental

Operating Temp.

- -25°C ~ 50°C (-13°F ~ 122°F) with 0.6 m/s airflow

*Operating temperature varies by accessories installed.

**SoM performance may vary with installation method at high operating temperatures.

Storage Temp.

- -40°C ~ 80°C (-40°F ~ 176°F)

Relative Humidity

- 10% RH ~ 90% RH (non-condensing)

Vibration

- IEC60068-2-64, random, 2.5G@5~500Hz, 1hr/axis
- MIL-STD-810H, Method 514.8, Procedure I, Category 4

Shock

- MIL-STD-810H, Method 516.8, Procedure I, Trucks and semi-trailers = 15G (11ms)

Certification / Compliance

- CE, FCC Class A, UKCA, IEC 62368-1

Mechanical

Construction

- Aluminum alloy

Antenna

- 3 x SMA connector mounting hole

Mounting

- Wall mounting

Net Weight

- < 339 g (0.75 lb)

Dimensions (L x W x H)

- 120 x 115 x 32 mm (4.73 x 4.53 x 1.26 in.)

Table 1-2 Ordering Information

Model Number	IBOX-604-G2-ONX16	NVIDIA® Jetson Orin™ NX 16GB / 2 x GMSL-2 / 1 x CAN FD / 1 x UART
	IBOX-604-G2-ONX8	NVIDIA® Jetson Orin™ NX 8GB / 2 x GMSL-2 / 1 x CAN FD / 1 x UART
	IBOX-604-G2-ON8	NVIDIA® Jetson Orin™ Nano 8GB / 2 x GMSL-2 / 1 x CAN FD / 1 x UART
Description	NVIDIA® Jetson Orin™ NX/Nano SoM w/ 2 x GMSL-2 / 1 x CAN FD / 1 x UART / 10-60 VDC Lightweight AI Drone Computer	

Table 1-3 Optional Accessories

Storage	M.2 2242 Key M NVMe SSD, -40°C ~ 85°C
Wi-Fi	M.2 2230 Key A-E Wi-Fi module, -40°C ~ 85°C
WWAN	M.2 3042/3052 Key B WWAN module, -40°C ~ 85°C
GNSS	M.2 2242 Key B GNSS module, -40°C ~ 85°C M.2 2230 Key A-E GNSS module, -40°C ~ 85°C
Power Adapter	AC/DC 100-240V/12V 60W C14 DC plug power adapter

Note:

All items listed in the Optional Accessories table are sold separately.

Table 1-4 Supported Camera List

Brand	Camera Part Number	Max. No. of Camera Connections
oToBrite	oToCAM264ISP Series (2MP, Sony IMX390)	2
	oToCAM260ISP Series (5.36MP, Sony IMX490)	-
	oToCAM223 Series (2.95MP, Sony ISX031)	2
e-con Systems	STURDeCAM25 (2MP, Omsemi AR0234CS)	2
	STURDeCAM31 (3MP, Sony ISX031)	2
	STURDeCAM81 (8MP, Omsemi AR0821)	2
StereoLabs	ZED X Stereo (Dual 2MP)	1
	ZED X One 4K (8.2MP)	1
	ZED X One S/GS (2.3MP)	2

*BSP versions may vary depending on supported GMSL cameras.

**All brand and product names mentioned are the property of their respective owners.

1.2. Product Photos

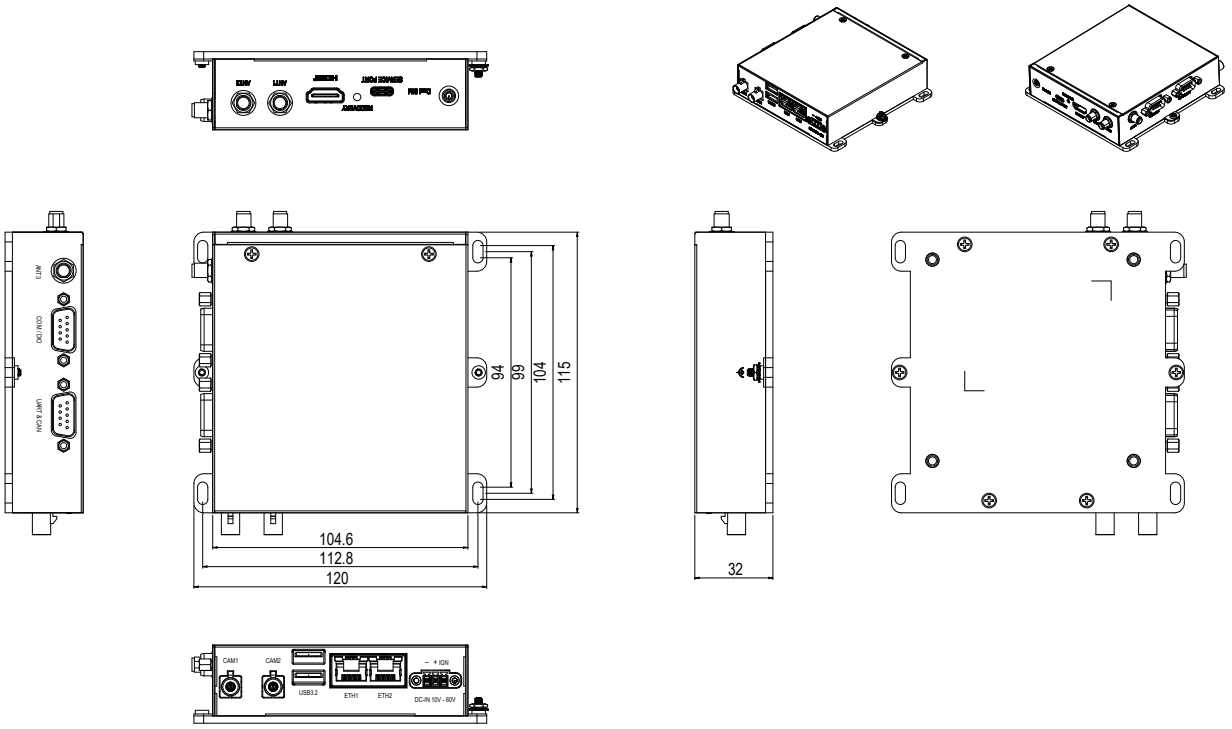
Figure 1-1 Front View of IBOX-604-G2

Figure 1-2 Left View of IBOX-604-G2

Figure 1-3 Rear View of IBOX-604-G2

1.3. Mechanical Drawings

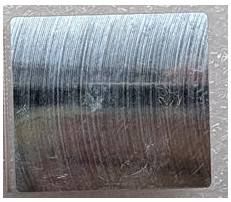
Unit : mm



1.4. Package Contents

See the following list to check if it matches your product package contents. Please contact SINTRONES® sales representatives or our sales partners if any of the items is missing.

Table 1-5 Package Contents for IBOX-604-G2

Item	Photo	Quantity	Description
IBOX-604-G2	See Product Photos *(on page 11)*	1	The edge AI computer
M3 Hex Standoff (H26D5)		4	Used to raise the system to maintain at least 2.6 cm clearance beneath it
Screw I (Type M2.5x5L)		2	Used to fasten M.2 expansion modules
M.2 WWAN Heatsink (34x30x6.8 mm)		1	Used to transfer heat from an installed M.2 3042 or 3052 WWAN module
Thermal pad (30x34x1.0T mm)		1	Used to transfer heat from an installed M.2 3042 or 3052 WWAN module
Power cable for IBOX-604-G2		1	Used to connect the computer to a DC power supply

1.5. Power Consumption

See the following table as the power consumption of IBOX-604-G2.

Table 1-6 IBOX-604-G2-ONX16

Mode	Input Voltage				
	10V	12V	24V	48V	60V
40W (SoM Power Mode)	36.60W	36.48W	36.48W	37.44W	37.20W
Suspend	18.20W	18.60W	19.20W	19.68W	21.00W
Idle	19.60W	19.92W	20.64W	22.08W	22.20W
Standby (Iginition ON)	6.80W	6.96W	6.96W	7.68W	7.80W

Table 1-7 IBOX-604-G2-ONX8

Mode	Input Voltage				
	10V	12V	24V	48V	60V
40W (SoM Power Mode)	36.30W	36.24W	36.24W	36.48W	36.60W
Suspend	18.10W	18.48W	18.96W	19.20W	20.40W
Idle	19.50W	19.92W	20.40W	21.60W	21.60W
Standby (Iginition ON)	6.80W	6.84W	6.72W	7.20W	7.80W

Table 1-8 IBOX-604-G2-ON8

Mode	Input Voltage				
	10V	12V	24V	48V	60V
25W (SoM Power Mode)	31.10W	30.60W	31.20W	32.64W	34.20W
Suspend	17.50W	17.76W	18.24W	19.20W	19.80W

Mode	Input Voltage				
	10V	12V	24V	48V	60V
Idle	18.00W	18.48W	18.96W	19.68W	20.40W
Standby (Iginition ON)	6.80W	7.08W	7.44W	8.16W	7.80W

2. Getting Started

Topics in this chapter include:

2.1. SoM and SSD Info

The pre-installed NVIDIA® Jetson system on modules (SoM) and NVMe SSD are located on the bottom side of the mainboard. It is suggested to consult SINTRONES technical support for expansion requirements for Jetson SoM or an NVMe SSD.

Important:

- **DO NOT** remove the pre-installed SoM, or replace the SSD without first preparing a backup image. Refer to Downloading BSP Image & Checksum Files *(on page 71)* and the associated system recovery procedures if SSD replacement is required.
- A minimum of **256 GB** capacity is recommended on the flash drive for NVMe storage.

Note:
For verified compatibility, It is recommended to purchase the SSD module from SINTRONES®.

See Bottom View of Mainboard *(on page 49)* for the locations of the NVIDIA® Jetson SoM and NVMe SSD.

Item	Description	Specification
SODIMM for Jetson Module	NVIDIA® Jetson module integrates CPU, GPU, and memory with pre-installed JetPack developer tools.	Information about Jetson Modules *(on page 18)*

2.1.1. Information about Jetson Modules

Table 2-1 NVIDIA® Jetson Orin™ NX/Nano Modules Pre-installed in IBOX-604-G2

Series	Jetson Orin NX Series		Jetson Orin Nano Series
Model	Jetson Orin NX 16GB	Jetson Orin NX 8GB	Jetson Orin Nano 8GB
GPU	1024-core NVIDIA® Ampere architecture GPU with 32 Tensor Cores		
CPU Frequency	2.0 GHz		1.5 GHz
Power Consumption	25W / 40W (Super Mode)	20W / 40W (Super Mode)	15W / 25W (Super Mode)
CPU	8-core Arm® Cortex® - A78AE	6-core Arm® Cortex® - A78AE	
DL Accelerator	2 x NVDLA v2	1 x NVDLA v2	-
Vision Accelerator	1 x PVA v2		-
Memory	16GB 128-bit LPDDR5	8GB 128-bit LPDDR5	8GB 128-bit LPDDR5
Storage	Supports external NVMe SSD (Pre-installed JetPack)		

2.1.2. Configuring Power Mode for Jetson Modules

NVIDIA® Jetson modules support multiple predefined power budgets, allowing users to select the **Power Mode** that best fits their needs via the Jetson Linux GUI or CLI.

By default, SINTRONES® system is not configured at the unconstrained power mode (MAXN or MAXN SUPER*). The system operating temperature range is validated under **non-MAXN/MAXN SUPER** high-power mode.

Therefore, when the system runs in **MAXN** or **MAXN SUPER** mode, the system's maximum operating temperature may be reduced. It is **NOT recommended to run prolonged operation under heavy workloads** in these modes.

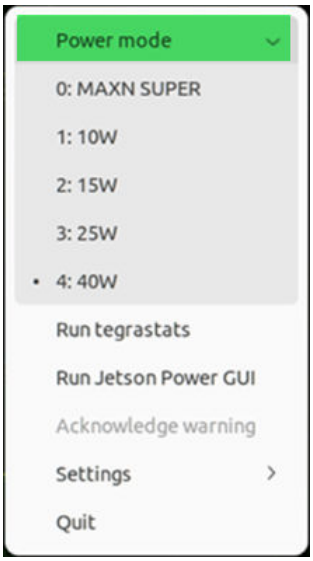
Note:

*The MAXN SUPER mode is only available in JetPack 6.2 for Jetson Orin Nano and Orin NX modules.

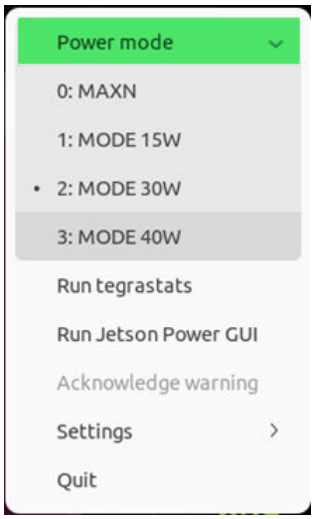
For detailed instructions on configuring power modes to optimize efficiency, please refer to the official NVIDIA® documentation.

Examples of **power mode selection menus** for Jetson Orin NX 16GB and Jetson AGX Orin 32GB:

Jetson Orin NX 16GB



Jetson AGX Orin 32GB



2.1.3. M.2 Key M Slot

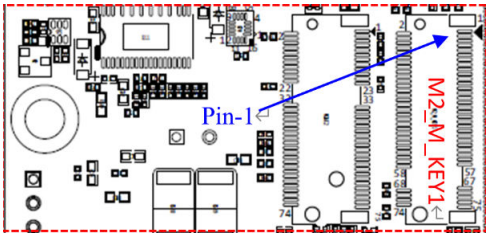
M.2 Key M Slot	Description
Size	NGFF 2242 / 75 Pin
Type	M.2 Key M (H: 8.5mm)
Supported Interface	PCIe x4
Location	See SoM and SSD Info *(on page 17)* for the information.
Drawing	

Table 2-2 Pin Definition of M.2 Key M Slot

Pin	Signal	Pin	Signal
1	GND	2	V3P3_M2M
3	GND	4	V3P3_M2M
5	PCIE0_RX3_N	6	NC
7	PCIE0_RX3_P	8	NC
9	GND	10	NC
11	PCIE0_TX3_N	12	V3P3_M2M
13	PCIE0_TX3_P	14	V3P3_M2M
15	GND	16	V3P3_M2M
17	PCIE0_RX2_N	18	V3P3_M2M
19	PCIE0_RX2_P	20	NC
21	GND	22	NC
23	PCIE0_TX2_N	24	NC
25	PCIE0_TX2_P	26	NC
27	GND	28	NC
29	PCIE0_RX1_N	30	NC
31	PCIE0_RX1_P	32	NC
33	GND	34	NC
35	PCIE0_TX1_N	36	NC

Pin	Signal	Pin	Signal
37	PCIE0_TX1_P	38	NC
39	GND	40	NC
41	PCIE0_RX0_N	42	NC
43	PCIE0_RX0_P	44	NC
45	GND	46	NC
47	PCIE0_TX0_N	48	NC
49	PCIE0_TX0_P	50	PCIE0_RST
51	GND	52	PCIE0_CLKREQ
53	PCIE0_CLK_N	54	PCIE_WAKE
55	PCIE0_CLK_P	56	NC
57	GND	58	NC
59	KEY	60	KEY
61	KEY	62	KEY
63	KEY	64	KEY
65	KEY	66	KEY
67	NC	68	NC
69	NC	70	V3P3_M2M
71	GND	72	V3P3_M2M
73	GND	74	V3P3_M2M
75	GND		

2.2. System Setup

This section describes how to install expansion modules for this computer.

Before you start the installation, check the following safety instructions:

Important:

- Ensure the system is not connected to any power source such as a power adapter or a battery.
- Prior to installing any modules on the mainboard, always touch an unpainted and grounded metal object or wear a grounded anti-static wrist strap to prevent electrostatic discharge (ESD).
- Maintain at least 2.6 cm of clearance beneath/above the IBOX-604-G2 bottom cover to allow proper ventilation and extend system lifespan. See Installing the System *(on page 33)* for more details.
- Install the IBOX-604-G2 inside an enclosed cabinet to prevent accidental contact with the system and ensure user safety.

See the following steps to start the installation:

2.2.1. Opening the Chassis

To install an M.2 3042/3052 **WWAN** or an **M.2 2242 GNSS** module, the front bracket must be removed.

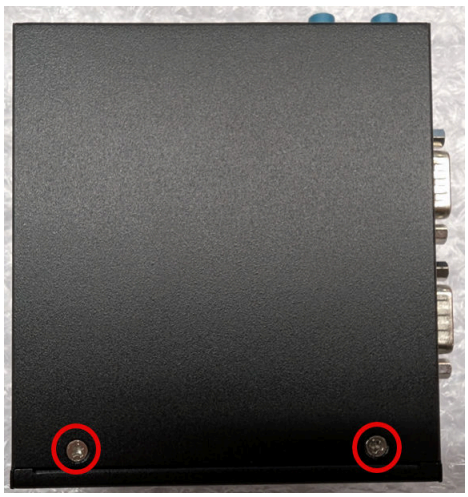
To install an M.2 2230 **WLAN** or **GNSS** module, the bottom cover must be removed following removal of the front bracket.

See the following sections for instructions on how to open the system chassis for internal expansion.

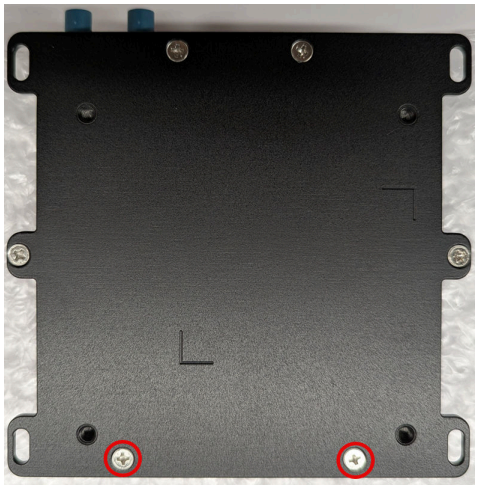
2.2.1.1. Removing the Front Bracket

1. Remove the hex nut from the right panel.

2. Remove the two screws on the bottom cover.

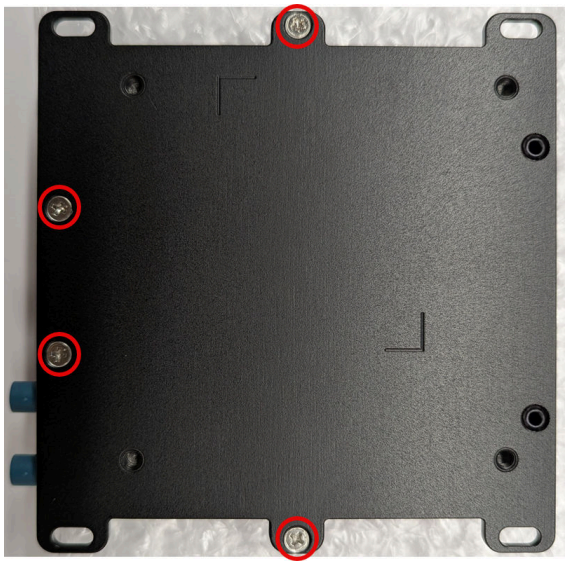


3. Turn over the system, and remove the two screws on the top side of the front bracket.

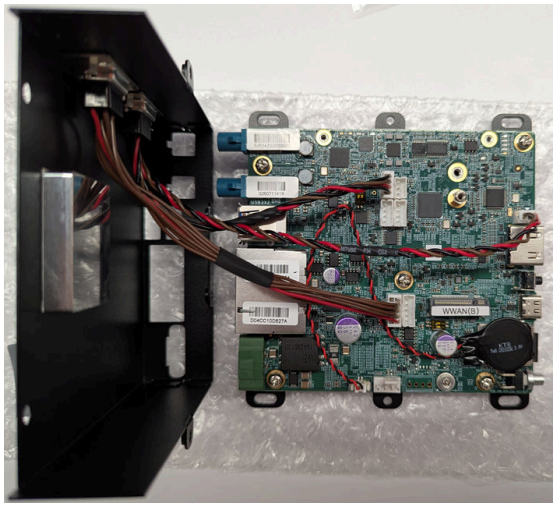


4. Remove the rear panel and place it carefully.

5. Remove the four screws from the top side of the front bracket.

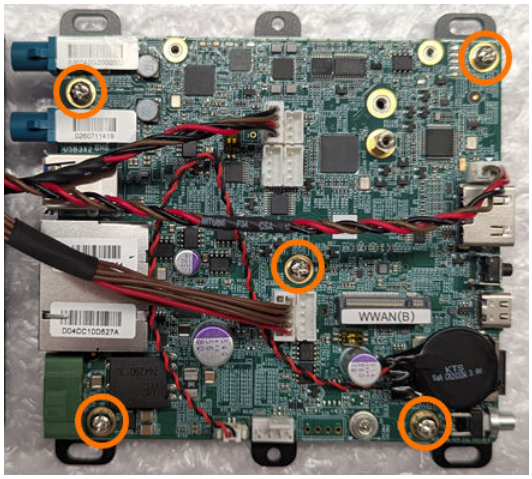


6. After removing all the specified screws, gently lift the front bracket and place it carefully.

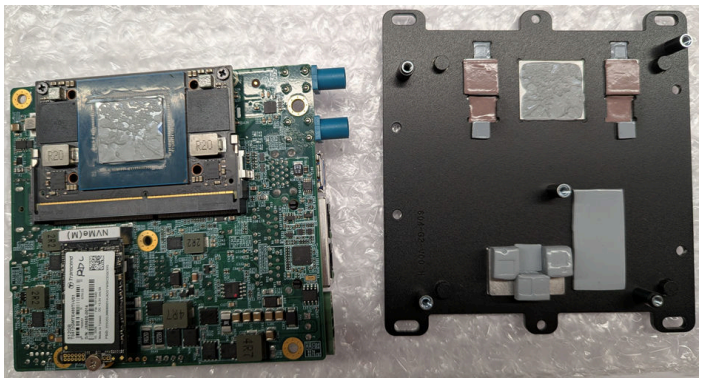


2.2.1.2. Removing the Bottom Cover

1. After removing the front bracket *(on page 22)*, remove the five screws from the mainboard.



2. Detach the mainboard from the bottom cover.



2.2.2. Installing Expansion Modules

The system provides an M.2 3042/3052 Key B slot for installing a WWAN or GNSS module, and an M.2 2230 Key E slot for installing a WLAN or GNSS module.

2.2.2.1. Installing an M.2 2230/2242 GNSS Module

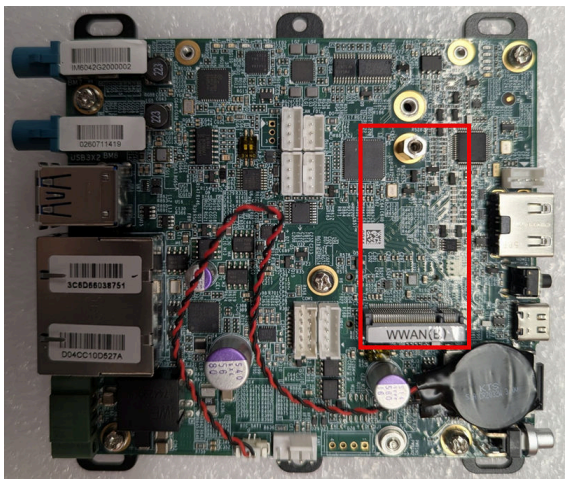
The system supports installation of an M.2 2230 GNSS expansion module in an M.2 2230 Key A-E slot, and an M.2 2242 GNSS expansion module in an M.2 3042/3052 Key B slot. Depending on specific wireless connectivity requirements, users can choose one of the following installation scenarios for the GNSS module:

- **WLAN + GNSS**:
  - Install a WLAN module in the M.2 2230 slot (located on the **bottom** side of the mainboard).
  - Install an M.2 **2242** GNSS module in the M.2 3042/3052 slot (located on the **top** side of the mainboard).
- **5G/LTE + GNSS**:
  - Install a WWAN module in the M.2 3042/3052 slot (located on the **top** side of the mainboard).
  - Install an M.2 **2230** GNSS module in the M.2 2230 slot (located on the **bottom** side of the mainboard).

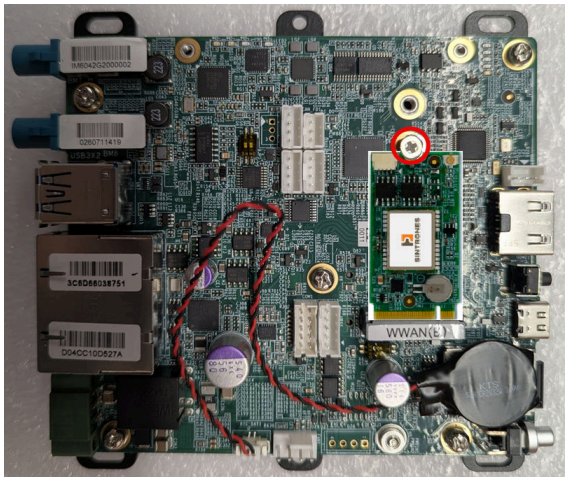
The following instructions assume that the GNSS modules are purchased from SINTRONES®.

2.2.2.1.1. Installing an M.2 2242 GNSS Module

1. Remove the front bracket *(on page 22)*.
2. Depending on the module's design and network support, connect the required number of antenna cables to the M.2 2242 GNSS module.
3. Locate the M.2 3042 key B slot labelled as **WWAN(B)** on the mainboard.



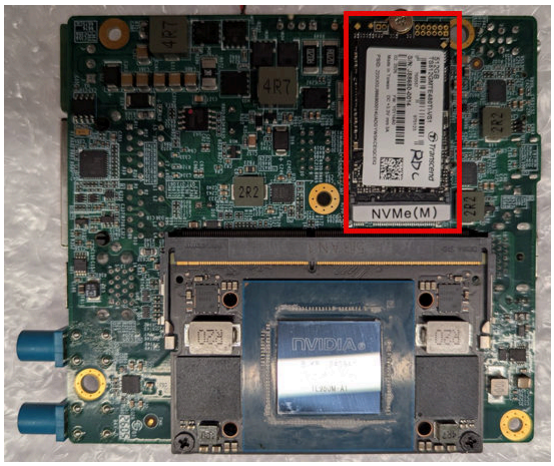
4. Align the notch on the M.2 2242 GNSS module with the tab in the slot and gently insert the module at a 30 degree angle until it is fully embedded, then press it down.
5. Secure the GNSS module to the mainboard using the M2.5x5L screw provided in the package *(on page 13)*.



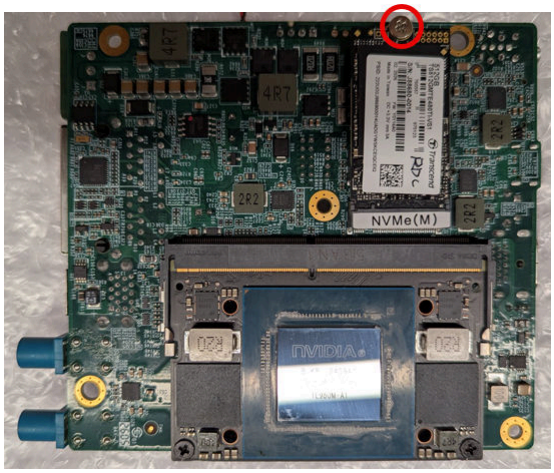
6. Secure the antenna jacks into the antenna holes, and then connect the external antennas to the antenna jacks.

2.2.2.1.2. Installing an M.2 2230 GNSS Module

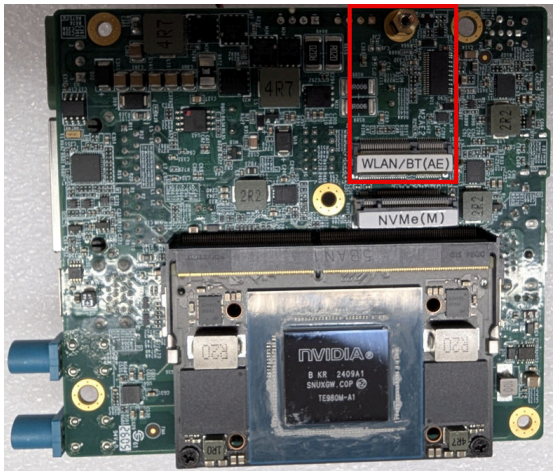
1. Remove the front bracket *(on page 22)*.
2. Remove the bottom cover *(on page 24)*.
3. Depending on the module's design and network support, connect the required number of antenna cables to the M.2 2230 GNSS module.
4. Locate the pre-installed M.2 2242 NVMe SSD module; the M.2 2230 slot is positioned beneath it.



5. Remove the screw to uninstall the NVMe SSD module.



6. Locate the M.2 2230 key E slot labelled as **WLAN/BT(AE)**.



7. Align the notch on the **GNSS** module with the tab in the slot and gently insert the module at a 30 degree angle until it is fully embedded, then press it down.

8. Secure the antenna jacks into the antenna holes, and then connect the external antennas to the antenna jacks.
9. Align the notch on the **NVMe SSD** module with the tab in the slot and gently insert the module at a 30 degree angle until it is fully embedded, then press it down.

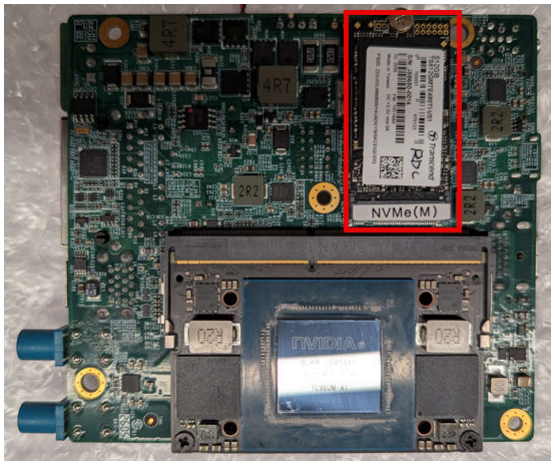
Important:
Due to height limitations, this slot supports only single-sided NVMe SSDs.

10. Secure both the **GNSS and NVMe SSD modules** to the mainboard using the M2.5x5L screw provided in the package *(on page 13)*.
11. Secure the antenna jacks into the antenna holes, and then connect the external antennas to the antenna jacks.

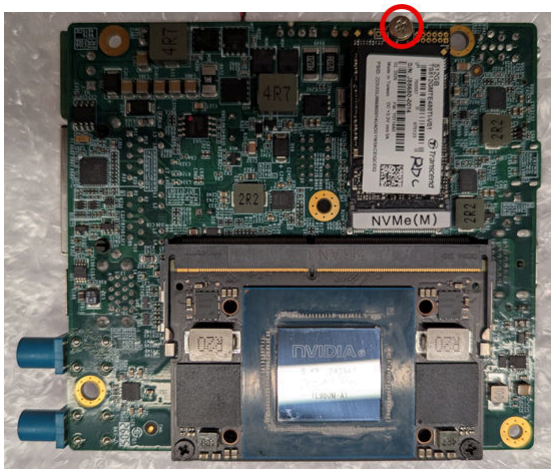
2.2.2.2. Installing an M.2 2230 WLAN Module

The system provides one M.2 key E slot for installing a WLAN module. The following instructions assume that the WWAN module is purchased from SINTRONES®.

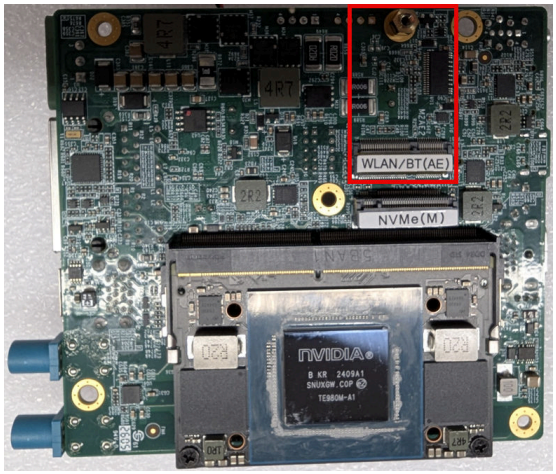
1. Remove the front bracket *(on page 22)*.
2. Remove the bottom cover *(on page 24)*.
3. Depending on the module's design and network support, connect the required number of antenna cables to the M.2 2230 WLAN module.
4. Locate the pre-installed M.2 2242 NVMe SSD module; the M.2 2230 slot is positioned beneath it.



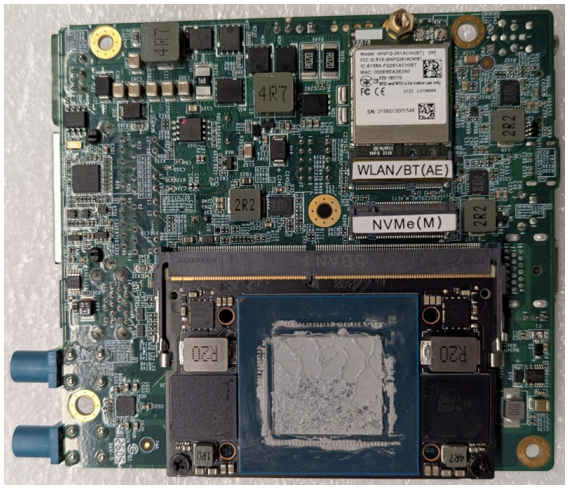
5. Remove the screw to uninstall the NVMe SSD module.



6. Locate the M.2 2230 key E slot labelled as **WLAN/BT(AE)**.



7. Align the notch on the **WLAN** module with the tab in the slot and gently insert the module at a 30 degree angle until it is fully embedded, then press it down.



8. Secure the antenna jacks into the antenna holes, and then connect the external antennas to the antenna jacks.
9. Align the notch on the **NVMe SSD** module with the tab in the slot and gently insert the module at a 30 degree angle until it is fully embedded, then press it down.
10. Secure both the **WLAN and NVMe SSD modules** to the mainboard using the M2.5x5L screw provided in the package *(on page 13)*.

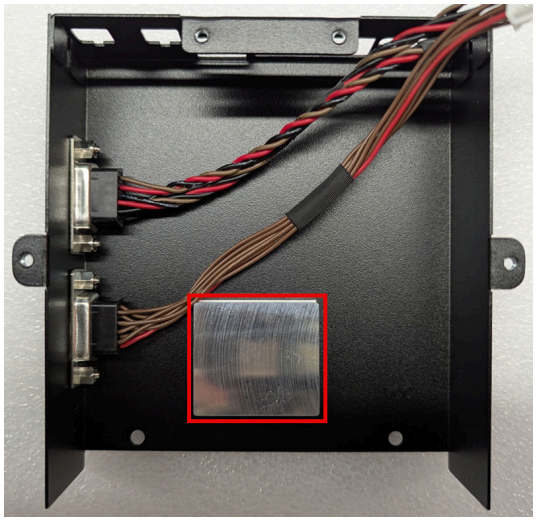
2.2.2.3. Installing an M.2 3042/3052 WWAN Module

The system provides an M.2 3042/3052 key B slot for installing a WWAN expansion module. The following instructions assume that the WWAN module is purchased from SINTRONES®.

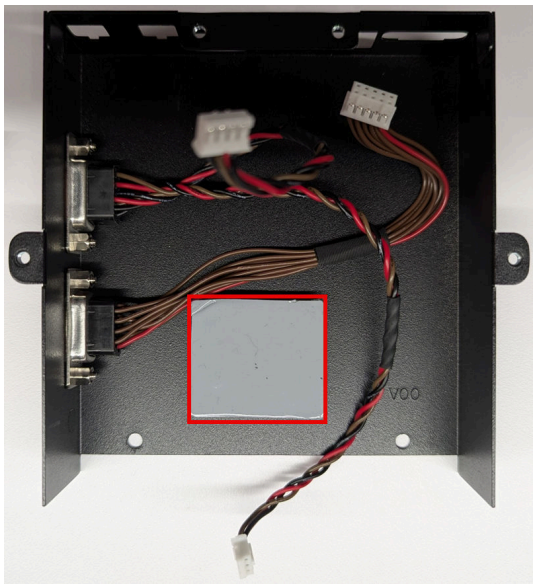
2.2.2.3.1. Installing the Heatsink

Check the accessory package *(on page 13)* for the **heatsink** and **thermal pad** used to dissipate heat for the M.2 3042/3052 WWAN module.

1. Attach the heatsink to the area as indicated below.



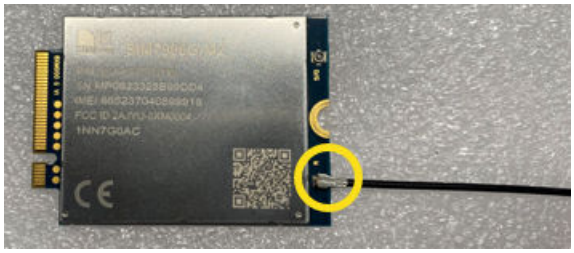
2. Attach the thermal pad to the heatsink.



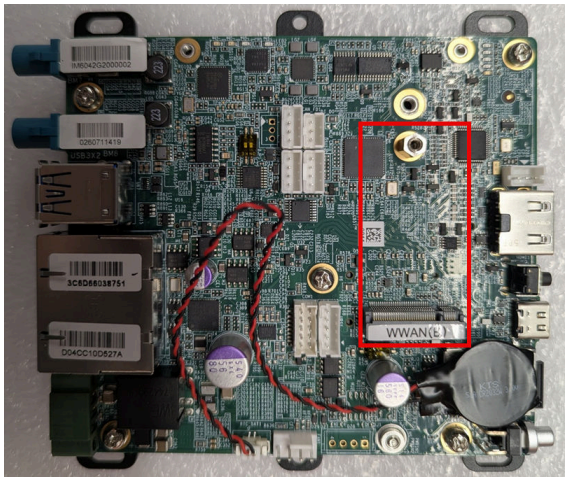
2.2.2.3.2. Installing a 4G LTE Module

Follow the instructions below to install the 4G LTE module into the system.

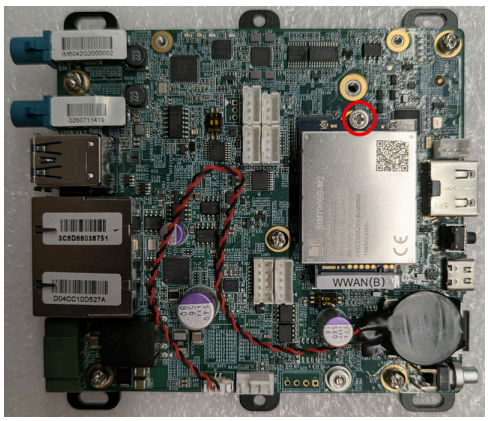
1. Remove the front bracket *(on page 22)*.
2. Install the heatsink *(on page 29)*.
3. Depending on the module's design and network support, connect the required number of antenna cables to the M.2 3042 **4G LTE** WWAN module.



4. Locate the M.2 3042 key B slot labelled as **WWAN(B)** on the mainboard.



5. Align the notch on the M.2 3042 WWAN module with the tab in the slot and gently insert the WWAN card at a 30 degree angle until it is fully embedded, then press it down.
6. Secure the expansion module to the mainboard using the M2.5x5L screw provided in the package *(on page 13)*.



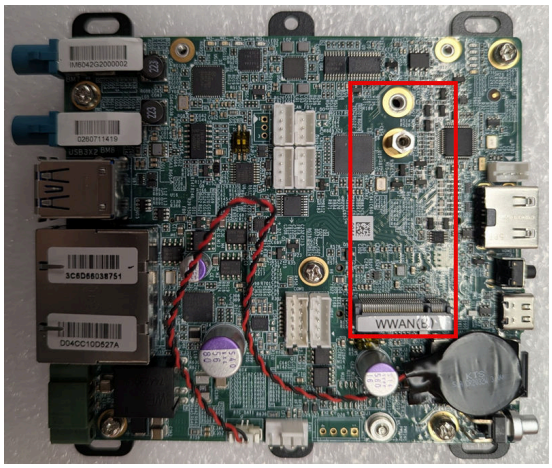
7. Secure the antenna jacks into the antenna holes, and then connect the external antennas to the antenna jacks.
8. Reassemble the system.
9. Insert the paired SIM card(s) into the SIM card slot(s).

2.2.2.3.3. Installing a 5G Module

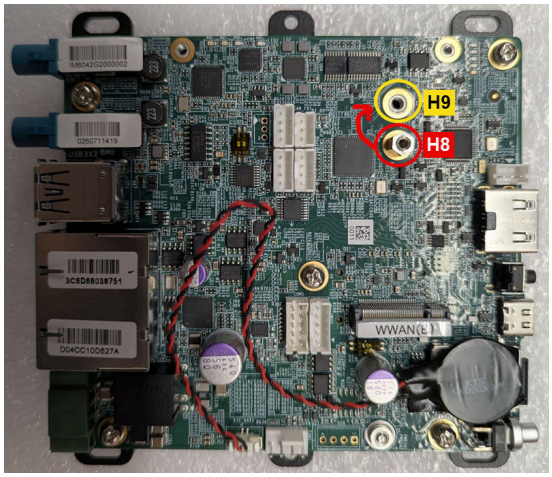
Follow the instructions below to install the 5G module into the system.

1. Remove the front bracket *(on page 22)*.
2. Install the heatsink *(on page 29)*.
3. Depending on the module's design and network support, connect the required number of antenna cables to the M.2 3052 **5G** WWAN module.

4. Locate the M.2 3052 Key B slot labelled as **WWAN(B)** on the mainboard.



5. Move the copper pillar from H8 to H9 as indicated below.



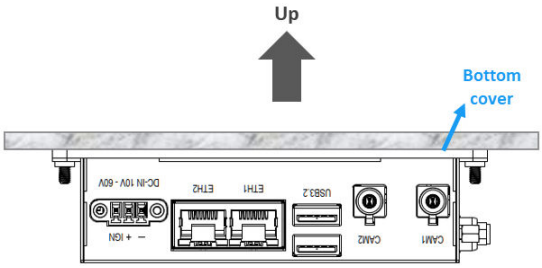
6. Align the notch on the M.2 3052 WWAN module with the tab in the slot and gently insert the WWAN card at a 30 degree angle until it is fully embedded, then press it down.
7. Secure the expansion module to the mainboard using the M2.5x5L screw provided in the package *(on page 13)*.

8. Secure the antenna jacks into the antenna holes, and then connect the external antennas to the antenna jacks.
9. Reassemble the system.
10. Insert the paired SIM card(s) into the SIM card slot(s).

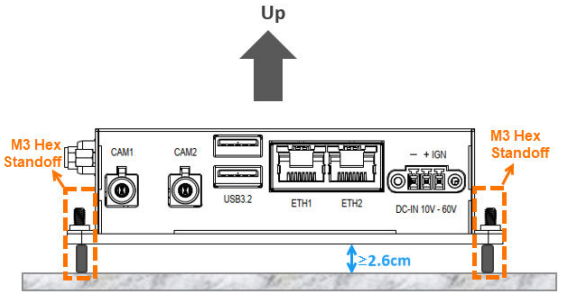
2.2.3. Installing the System

For optimal heat dissipation, install the IBOX-604-G2 either **upside down** or **raise the system** to maintain at least 2.6 cm clearance above or beneath the bottom cover. The following illustrations show the recommended installation methods.

System Upside Down



System Raised



Note:
The M3 Hex Standoff (H26D5) *(on page 13)* provided in the package can be used to raise the system.

Important:

- System performance may degrade if overheating occurs.
- To ensure user safety, **DO NOT** touch the sytem while it is operating.

2.3. Booting the System

This section describes how to boot the system via an ignition switch or the power button.

2.3.1. Wiring the Terminal Block

This system supports a DC power input of **10V to 60V** through a 3-pin terminal block. Follow the steps below to connect the power wires.

1. Get the terminal block provided in the accessory package.
2. Prepare the wires for connecting the terminal block to a power supply.
3. Insert the **positive (+)** wire into the terminal marked "+" and the **negative (-)** wire into the terminal marked "-". See the following diagrams as the illustration of wiring an ignition switch to use in fields with or without an ignition system.

Figure 2-1 Fields WITH an Ignition System

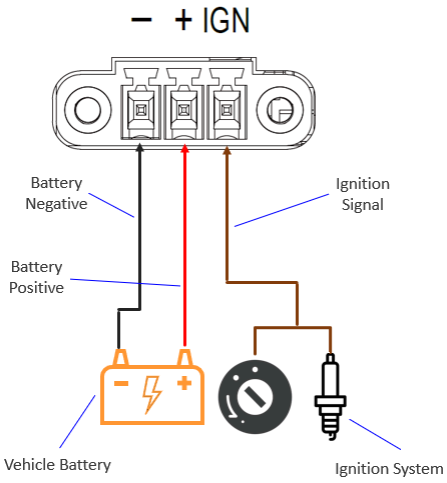
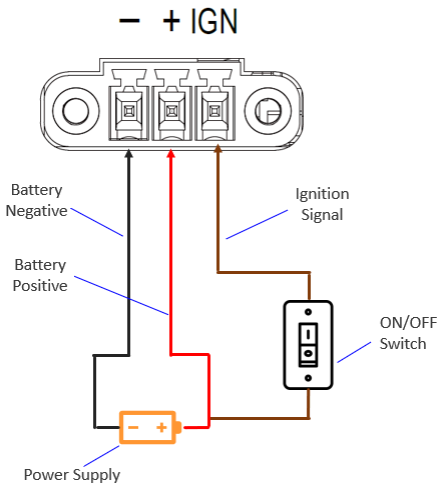


Figure 2-2 Fields WITHOUT an Ignition System



4. Tighten the screws on the terminal block to secure the wires firmly.
5. Plug the terminal block into the DC-IN port of the computer.
6. Connect the other ends of the wires to a stable DC power source within the supported voltage range.
7. Ensure the power source is switched off while wiring, then verify the connection before powering on the system to avoid damage.

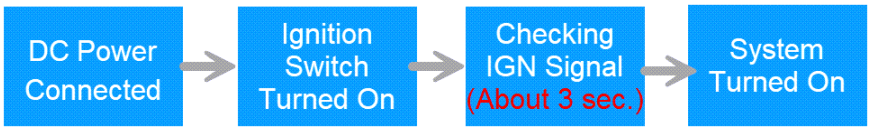
2.3.2. Powering On the System by Ignition Switch

1. When the system is connected to a stable DC power source within the supported voltage range, turn on the ignition switch.
2. The system starts checking if there's any incoming ignition signal. It takes about **3 seconds** for the system to run the ignition control process.

Note:
SINTRONES® provides ignition power management that monitors the ignition signal and controls the system boot process to avoid potential risk of downtime or damages to the connected devices and system.

3. After the ignition control process is complete, the system will boot up.

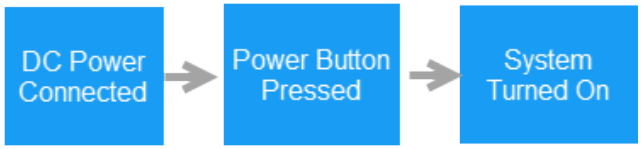
Figure 2-3 Turning On System By Ignition Switch



2.3.3. Powering On the System by Button

1. When the system is connected to a stable DC power source within the supported voltage range, press the power button.
2. The **Power Button** turns to blue light when the system boots up.

Figure 2-4 Turning On the System



3. External I/O Ports

Topics in this chapter include:

3.1. Front Panel

This section provides the front panel view and the description of the associated I/O interface.

Figure 3-1 Front Panel of IBOX-604-G2

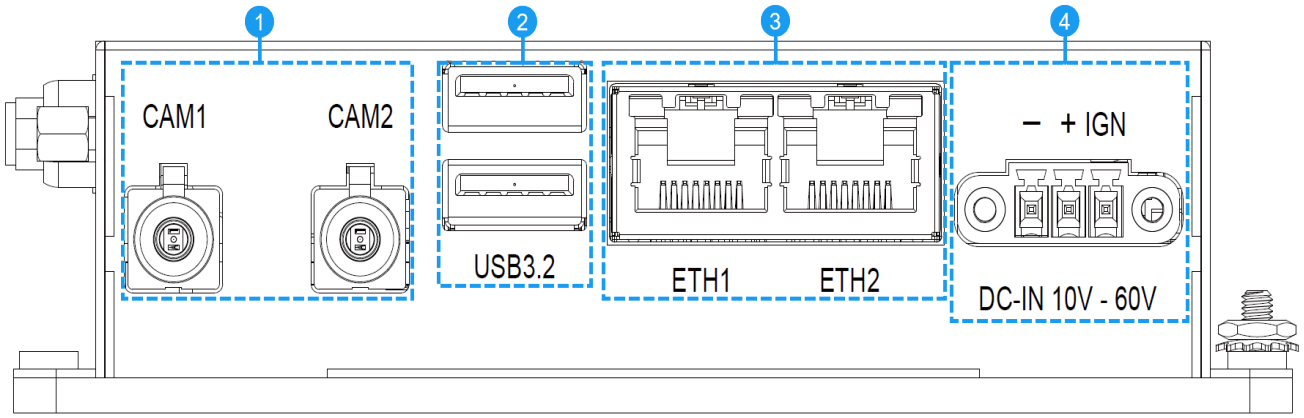


Table 3-1 I/O Interface on Front Panel

Item	I/O Interface	Description	Specification
1	Camera Ports	FAKRA-Z connectors for GMSL-2 cameras **Important:** Only the **CAM2** port supports Stereolabs ZED X Stereo and ZED X One 4K cameras.	-
2	USB Ports	USB 3.2, 5V / 900mA	USB 3.2 Port *(on page 41)*
3	LAN Ports	• LAN/ETH 1: GbE (10/100/1000BASE-T) • LAN/ETH 2: 2.5 GbE (10/100/1000/2500BASE-T)	-
4	DC Input	Input voltage range: DC 10-60V (nominal power input DC 11V/22V/58V) via 3-pin terminal block	DC-IN Port (Terminal Block) *(on page 41)*

3.2. Left Panel

This section provides the side panel view and the description of the associated I/O interface.

Figure 3-2 Left Panel of IBOX-604-G2

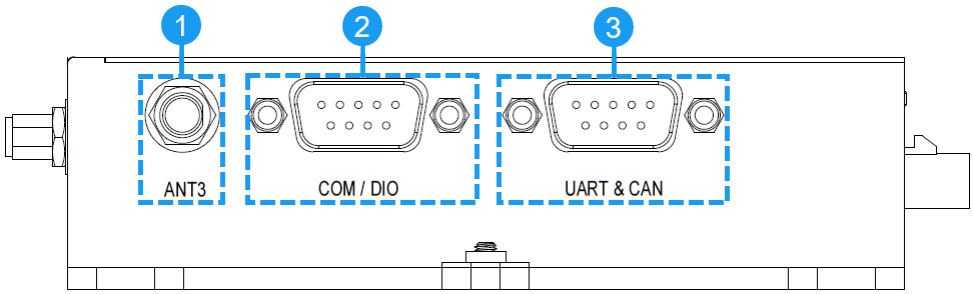


Table 3-2 I/O Interface on Left Panel

Item	I/O Interface	Description	Specification
1	SMA Connector Mounting Holes	Used for connecting antennas	-
2	COM / DIO Port	This port supports either a serial (COM) interface or a digital I/O (DIO) interface, depending on the selected SKU. • COM port: RS-232/422/485 serial port • DIO port (Optional): 1 x DI , 1 x DO , I²C (customized SKU)	• COM Port *(on page 42)*
3	UART & CAN Port	• UART: 3.3V • CAN port: ◦ Supports CAN FD protocol ◦ Backward compatible with CAN bus 2.0	UART & CAN Port *(on page 43)*

3.3. Rear Panel

This section provides the rear panel view and the description of the associated I/O interface.

Figure 3-3 Rear Panel of IBOX-604-G2

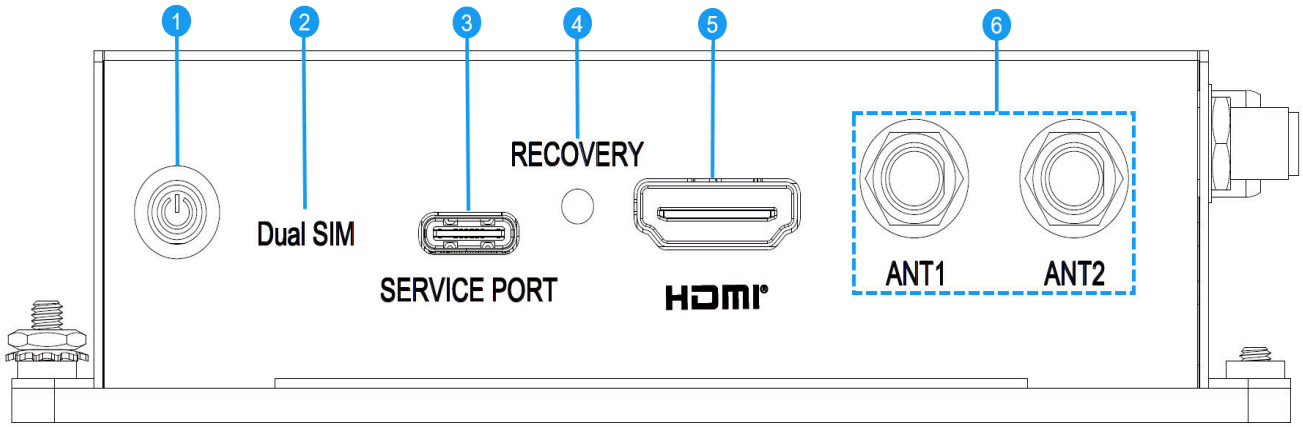


Table 3-3 I/O Interface on Rear Panel

Item	I/O Interface	Description	Specification
1	Power Button	System power status: • Red light: Standby mode • Blue light: System turned on	Power Button (LED Light Status) *(on page 45)*
2	Dual SIM Card Slot	Supports dual nano SIM cards **Note:** Ensure you power off the system before installing or removing the SIM card(s).	-
3	Service Port	Used for system recovery when connected to a host computer containing certain JetPack BSP image via a USB Type-C cable. See System Recovery *(on page 71)* for the detailed instructions. **Note:** This USB port is used only for system recovery. It does not support power or other kinds of data transfer.	USB Type-C Port *(on page 44)*
4	Recovery Button	Used for system recovery **Note:** The recovery button works only when the Service port is connected to a host computer.	-

Item	I/O Interface	Description	Specification
5	HDMI® Port	Supports HD video output via HDMI® Type-A connector	HDMI® Port *(on page 44)*
6	SMA Connector Mounting Holes	Used for connecting antennas	-

3.4. Specifications of External I/O Ports

This section provides drawings and pin definitions of the external I/O ports.

3.4.1. USB 3.2 Port

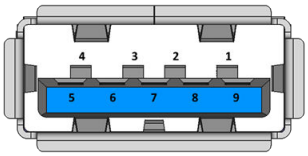


Table 3-4 Pin Definition of USB 3.2 Port

Pin	Signal
1	VBUS
2	D-
3	D+
4	GND
5	StdA_SSRX-
6	StdA_SSRX+
7	GND_DRAIN
8	StdA_SSTX-
9	StdA_SSTX+

3.4.2. DC-IN Port (Terminal Block)

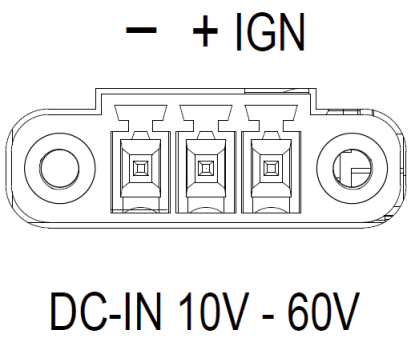


Table 3-5 Pin Definition of DC-IN Port

Pin	Signal
1	V-
2	V+
3	Ignition

3.4.3. COM Port

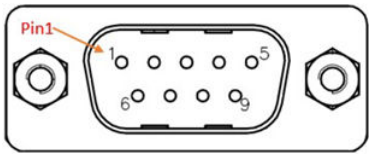


Table 3-6 Pin Definition of COM Port (D-Sub 9 (Male))

Pin	D-Sub 9 (Male)		
	RS-232	RS-422	RS-485
1	NC	TXD-	TXD-/RXD-
2	COM_RXD	TXD+	TXD+/RXD+
3	COM_TXD	RXD+	NC
4	NC	RXD-	NC
5	GND	GND	GND
6	NC	NC	NC
7	COM_RTS	NC	NC
8	COM_CTS	NC	NC
9	NC	NC	NC

3.4.4. UART & CAN Port

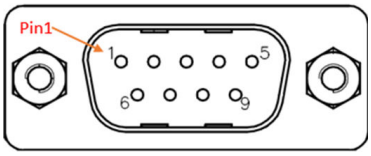


Table 3-7 Pin Definition of UART & CAN Port

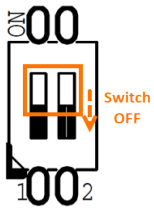
Pin	D-Sub 9 Male
1	UART_GND
2	CAN_L
3	GND
4	UART_RX
5	UART_TX
6	GND
7	CAN_H
8	NC
9	NC

3.4.4.1. DIP Switch for CAN FD Port

The DIP switch (on the mainboard) controls the 120Ω terminating resistors for the CAN FD port.

By default, the DIP switch bits are set to **ON**.

Refer to the table below for the bit assignments and corresponding terminating control.

DIP Switch Illustration	Description
	**Bit 1 and Bit 2 ON**: 120Ω termination enabled for the CAN bus (CLI port name: **can0)**.
	**Bit 1 and Bit 2 OFF**: 120Ω termination disabled for the CAN bus (CLI port name: **can0**).

3.4.5. HDMI® Port

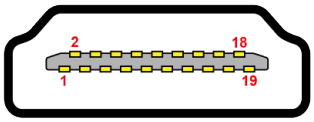


Table 3-8 Pin Definition of HDMI® Port

Pin	Signal	Pin	Signal
1	HDMI_DATA2_P	2	GND
3	HDMI_DATA2_N	4	HDMI_DATA1_P
5	GND	6	HDMI_DATA1_N
7	HDMI_DATA0_P	8	GND
9	HDMI_DATA0_N	10	HDMI_CLK_P
11	GND	12	HDMI_CLK_N
13	NCCEC	14	NC
15	HDMI_SCL	16	HDMI_SDA
17	GND	18	V5P_S_HDMI
19	HDMI_HPDET		

3.4.6. USB Type-C Port

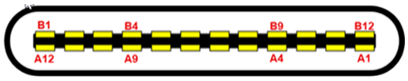


Table 3-9 Pin Definition of USB Type-C Port

Pin	Signal	Pin	Signal
A1	GND	B12	GND
A2	NC	B11	NC
A3	NC	B10	NC
A4	NC	B9	NC
A5	NC	B8	NC
A6	Dp1	B7	Dn2
A7	Dn1	B6	Dp2
A8	NC	B5	NC
A9	NC	B4	NC
A10	NC	B3	NC
A11	NC	B2	NC
A12	GND	B1	GND

3.4.7. Power Button (LED Light Status)

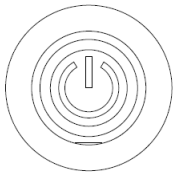


Table 3-10 Pin Definition of Power Button (LED Light Status)

Pin	Signal	Pin	Signal
LED			
A1	POWER_LED+	C1	POWER_LED-

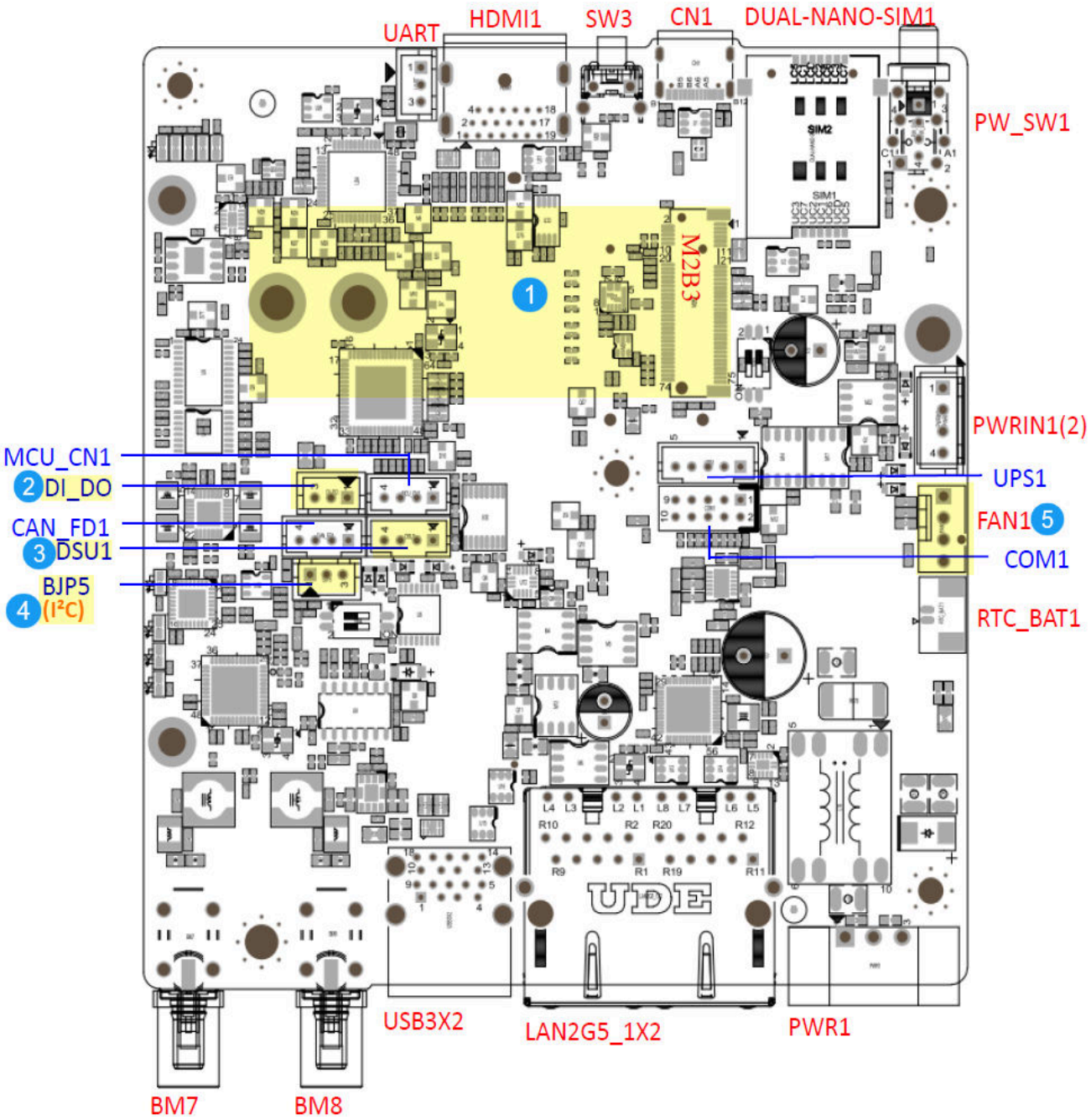
4. Expansion

This chapter provides more details about the internal slots or connectors used for expansion purpose.

Topics in this chapter include:

4.1. Top View of Mainboard

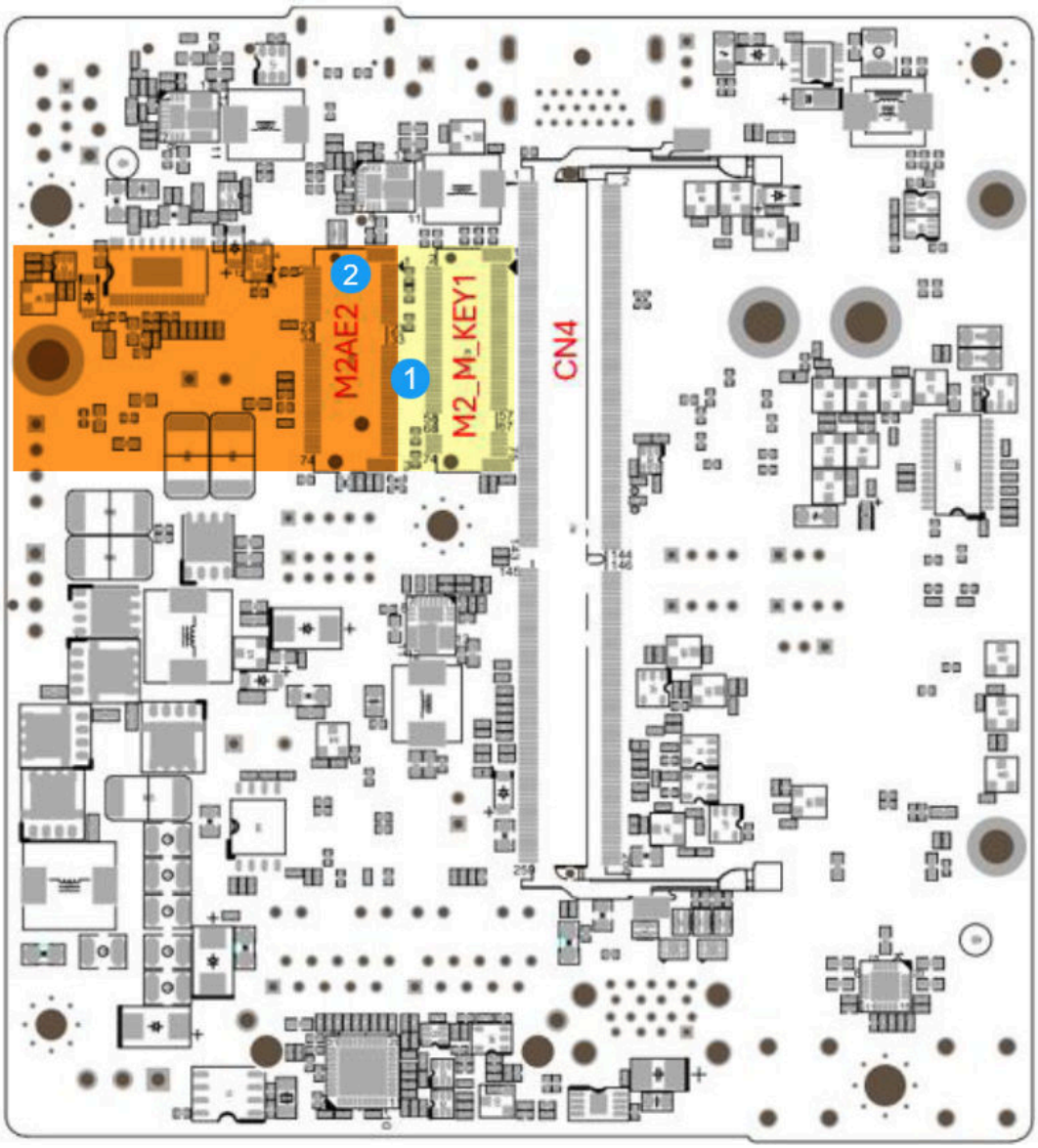
Figure 4-1 Top View of Mainboard



Item	Internal Connector	Description	Specification
1	M2B3	M.2 3042/3052 Key B slot used for installing an expansion module such as a WWAN or GNSS module.	M.2 Key B Slot *(on page 50)*
2	DI_DO	Used for digital peripheral control and signaling.	DI/DO Connector (Optional) *(on page 51)*
3	DSU1	Used for monitoring system status and collecting debug logs	DSU JST Connector *(on page 52)*
4	BJP5 (I²C)	• Reserved I²C connector for customizable IBOX-604 non-GMSL SKU. Contact SIN-TRONES® sales for details. • Supports serial communication with multiple sensors or peripheral devices.	I²C Connector (Customizable SKU) *(on page 53)*
5	PWM Fan	Used for connecting a PWM fan to the mother-board for cooling.	PWM Fan Connector *(on page 53)*

4.2. Bottom View of Mainboard

Figure 4-2 Bottom View of Mainboard



Item	Internal Connector	Description	Specification
1	M.2_M_KEY	M.2 2242 Key M slot used for installing an NVMe SSD. **Important:** Consult SINTRONES technical support for requirements related to NVMe SSD expansion.	M.2 Key M Slot *(on page 20)*
2	M.2_E_KEY	M.2 2230 Key E slot used for installing an expansion module such as a WLAN, Bluetooth, or GNSS module.	M.2 Key E Slot *(on page 54)*

4.3. Specifications of Expansion Slots/Connectors

This section provides drawings and pin definitions about the slots or connectors used to install expansion modules.

4.3.1. M.2 Key B Slot

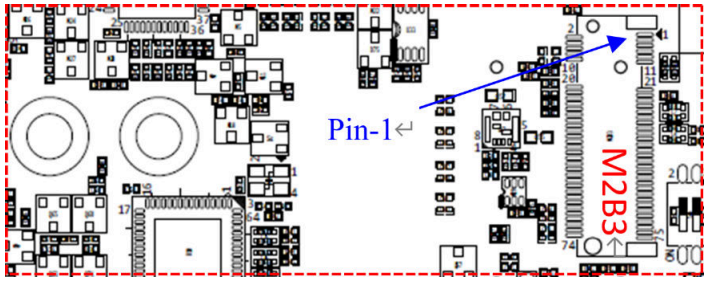
M.2 Key B Slot	Description
Size	NGFF 3042 / 3052 / 75 Pin
Type	M.2 Key B (H: 8.5mm)
Supported Interface	PCIe / USB 3.2 / USB 2.0 / I^2C
Location	See Top View of Mainboard *(on page 47)* for the information.
Drawing	

Table 4-1 Pin Definition of M.2 Key B Slot

Pin	Signal	Pin	Signal
1	NC	2	V3P3A_M2B1
3	GND	4	V3P3A_M2B
5	GND	6	PWR_OFF (Pull-H_1.8V)
7	HUB3_DP_DN1	8	W_DIS1# (Pull-H_3.3V)
9	HUB3_DM_DN1	10	LED# (Pull-H_3.3V)
11	GND	12	KEY
13	KEY	14	KEY
15	KEY	16	KEY
17	KEY	18	KEY
19	KEY	20	NC
21	NC	22	NC
23	WWAN_WAKE (Reserve)	24	NC
25	M2B1DPR	26	WDIS2# (Pull-H_1.8V)
27	GND	28	NC

Pin	Signal	Pin	Signal
29	HUB3_SSRXM_DN1	30	M2UIM_RST_A
31	HUB3_SSRXP_DN1	32	M2UIM_CLK_A
33	GND	34	M2UIM_DAT_A
35	HUB3_SSTXM_DN1	36	M2UIM_PWR_A
37	HUB3_SSTXP_DN1	38	NC
39	GND	40	GPS_CLK_M2B
41	NC	42	GPS_DATA_M2B
43	NC	44	NC
45	GND	46	NC
47	NC	48	NC
49	NC	50	NC
51	GND	52	NC
53	NC	54	WWAN_WAKE (Reserve)
55	NC	56	NC
57	GND	58	NC
59	NC	60	NC
61	NC	62	NC
63	NC	64	NC
65	NC	66	NC
67	M2B1RST2	68	NC
69	NC	70	V3P3A_M2B1
71	GND	72	V3P3A_M2B1
73	GND	74	V3P3A_M2B1
75	NC		

4.3.2. DI/DO Connector (Optional)

DI/DO Connector	Description
Size	1 x 3 = 3 Pin
Type	JST-2.0mm-M-180
Location	See Top View of Mainboard *(on page 47)* for the information.

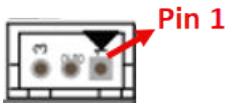
DI/DO Connector	Description
Drawing	

Table 4-2 Pin Definition of DI/DO JST Connector

Pin	Signal
1	AUTO_DO1
2	DI_1
3	GND_DIO

4.3.3. DSU JST Connector

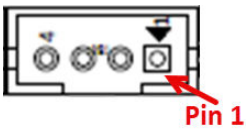
DSU JST Connector	Description
Size	1 x 4 / 4 Pin
Type	JST-2.0mm-M-180
Location	See Top View of Mainboard *(on page 47)* for the information.
Drawing	

Table 4-3 Pin Definition of DSU JST Connector

Pin	Signal
1	Power (Reserve)
2	JETSON_UART2_TX
3	JETSON_UART2_RX
4	GND

The DSU JST connector is used to monitor the system status and collect debug logs via UART protocol by advanced developers. There's no need to use this connector under normal operation.

To retrieve UART logs, connect a host computert to the system with a USB-UART or RS232-UART converter cable (not included in the package), and then configure the UART settings on the host computer as shown below:

- Baud rate: 115,200bps
- Voltage: DC 3.3V
- Data bits: 8

- Parity bit: 0
- Flow control: No

4.3.4. I²C Connector (Customizable SKU)

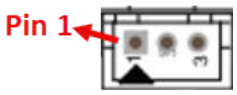
I²C Connector	Description
Size	1 x 3 = 3 Pin
Type	JST-2.0mm-M-180
Location	See Top View of Mainboard *(on page 47)* for the information.
Drawing	

Table 4-4 Pin Definition of I²C JST Connector

Pin	Signal
1	I2C0A_SCL_SB (3.3V)
2	I2C0A_SDA_SB (3.3V)
3	GND

4.3.5. PWM Fan Connector

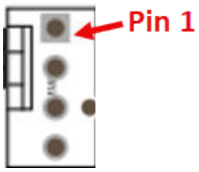
PWM Fan Connector	Description
Size	1 x 4 = 4 Pin
Type	JST-2.54mm-M-180
Location	See Top View of Mainboard *(on page 47)* for the information.
Drawing	

Table 4-5 Pin Definition of PWM Fan JST Connector

Pin	Signal
1	GND
2	FAN_POWER (12V)

Pin	Signal
3	FAN_TACH_OUT
4	FAN_PWM_IN

4.3.6. M.2 Key E Slot

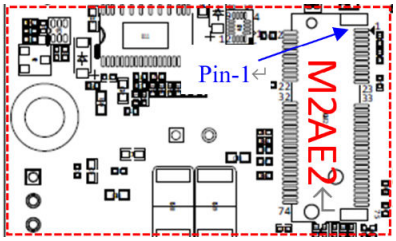
M.2 Key E Slot	Description
Size	NGFF 2230 / 75 Pin
Type	M.2 Key E (H: 4.0mm)
Supported Interface	PCIe x1 / USB 2.0
Location	See Top View of Mainboard *(on page 47)* for the information.
Drawing	

Table 4-6 Pin Definition of M.2 Key E Slot

Pin	Signal	Pin	Signal
1	GND	2	V3P3_A
3	HUB3_DP_DN2	4	V3P3_A
5	HUB3_DM_DN2	6	NC
7	GND	8	NC
9	NC	10	NC
11	NC	12	NC
13	NC	14	NC
15	NC	16	NC
17	NC	18	GND
19	NC	20	NC
21	NC	22	NC
23	NC	24	KEY
25	KEY	26	KEY
27	KEY	28	KEY

Pin	Signal	Pin	Signal
29	KEY	30	KEY
31	KEY	32	NC
33	GND	34	NC
35	PCIE1_TX0_P	36	NC
37	PCIE1_TX0_N	38	NC
39	GND	40	NC
41	PCIE1_RX0_P	42	NC
43	PCIE1_RX0_N	44	NC
45	GND	46	NC
47	PCIE1_CLK_P	48	NC
49	PCIE1_CLK_N	50	NC
51	GND	52	PCIE1_RST
53	PCIE1_CLKREQ	54	M2E_WIFI_DIS2(BT)
55	PCIE_WAKE	56	M2E_WIFI_DIS1(WIFI)
57	GND	58	NC
59	NC	60	NC
61	NC	62	M2E_ALERT
63	GND	64	NC
65	NC	66	NC
67	NC	68	NC
69	GND	70	NC
71	NC	72	V3P3_A
73	NC	74	V3P3_A
75	GND		

5. Software

Topics in this chapter include:

5.1. System Configuration

This section provides procedures about how to install the GMSL camera driver as well as summarizes commands available for configuring smart power management, COM (RS-232/422/485), CAN FD, and DI/DO interface controllers.

5.1.1. Setting the GMSL Camera

This section provides instructions on how to initialize and configure the connected GMSL cameras.

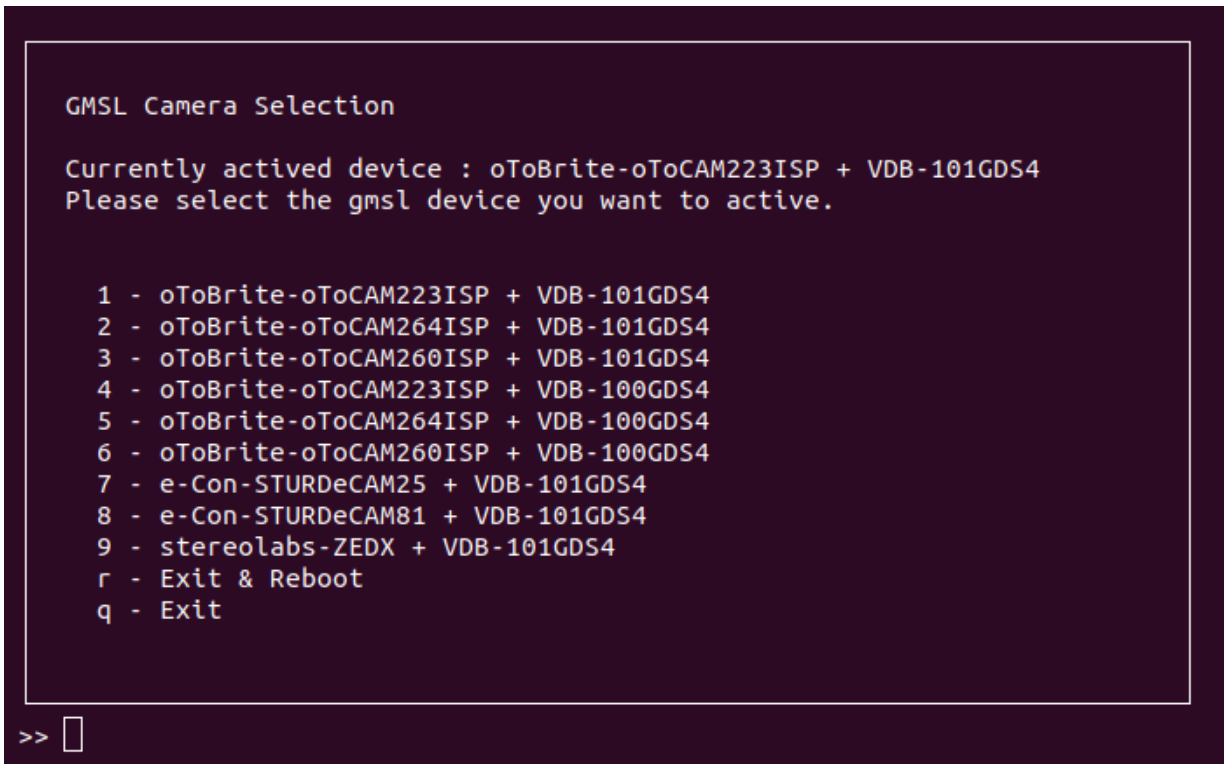
See the following steps:

1. Before connecting GMSL cameras, ensure the system is powered off.
2. Connect the GMSL cameras to the camara ports via the FAKRA-Z cables.

   Important:
   Connect only one type of cameras at a time. Different camera types cannot be mixed because the system does not support running multiple camera drivers simultaneously.

3. Connect the system to an HDMI monitor, a keyboard, and a mouse.
4. Boot the system.
5. Open the terminal window.
6. Enter the following command to pull up a list of the available GMSL camera modules.
   `sudo gmsl-camera-selection`
7. Select the GMSL camera module you want to initialize from the list.

   Figure 5-1 An **Example** of GMSL Camera Module List Available for Selection



Note:
Ensure you select the module that matches the connected cameras. Contact our sales representatives for assistance if needed.

8. Enter `r` to reboot the system.
9. Continue to the next sections for vendor-specific camera configuration instructions.

5.1.1.1. oToBrite GMSL Camera Settings

1. After the system reboots, enter the following command to verify whether the connected oToBrite GMSL cameras have been successfully initialized.
   ```
   ls /dev/vi*
   ```
2. For example, if you connect two oToBrite GMSL cameras to the system via CAM1 and CAM2 ports, the corresponding device names shall appear in the command output.

   Port names and corresponding device names shown in CLI:

   - CAM1: video1
   - CAM2: video2
3. To check a live video feed streaming from a specified camera, enter the following command:
   ```
   gst-launch-1.0 v4l2src device="/dev/<video?>" ! videoconvert ! xvimagesink &
   ```

   Note:
   Replace the texts within the brackets with the actual device name. For example, if the device name is `video1`, namely the camera connected via CAM1 port, enter

   ```
   gst-launch-1.0 v4l2src device="/dev/video1" ! videoconvert ! xvimagesink &
   ```
4. (Optional) If you want to add text overlay to the video streaming from CAM1, enter the following command:
   ```
   gst-launch-1.0 v4l2src device="/dev/video1" ! textoverlay text="VIDEO 1" valignment=top halignment=left font-desc="Sans, 18" ! clockoverlay valignment=top halignment=right time-format="%D %H:%M:%S" ! videoconvert ! xvimagesink
   ```

   Note:
   Likewise, replace `video1` with the actual device name to add text overlay to the video streaming from the specified camera.

   Important:
   To avoid unintended line breaks that may cause the command to fail, **copy the command into a plain text file to ensure it remains a single, continuous line** before pasting it into the terminal.

5.1.1.2. e-con Systems and Stereolabs GMSL Camera Settings

1. Connect the system to the Internet.
2. Download the GMSL camera streaming tool from the e-con Systems or Stereolabs website.
3. Follow the instructions on the e-con Systems or Stereolabs website to complete the respective camera settings.

   Important:
   Only the **CAM2** camera port supports Stereolabs ZED X Stereo and ZED X One 4K cameras.

5.1.2. Smart Power Management Settings

This section summarizes the commands available for configuring the Smart Power Management function.

5.1.2.1. Overview of Smart Power Management Commands

The general formats of the Smart Power Management commands are as follows:

`i2cget -y -f <i2c_num> <device_addr> <reg_addr>`: Used to **check** the current state of certain functionality.

`i2cset -f -y <i2c_num> <device_addr> <reg_addr> <value>`: Used to **change or save** the settings of certain functionality.

5.1.2.2. Smart Power Management Commands

See the following tables as the available commands for Smart Power Management.

Table 5-1 Checking IGN Status

Command	Value	Description	Outcome
`i2cget -y -f 1 0x4a 0x12`	-	Check the ignition (IGN) status	• `0x00`: Ignition turned off • `0x01`: Ignition turned on

Table 5-2 Checking and Setting Ignition Power On Delay Time

Command	Value	Description	Outcome
`i2cget -y -f 1 0x4a 0x60`	-	Check the ignition power on delay time by seconds	If the delay time is set as 3 seconds, the output will be `0x03`
`i2cset -f -y 1 0x4a 0x60`	For example: `0x03`	Set the ignition power on delay time by seconds	For example, if you enter `i2cset -f -y 1 0x4a 0x60 0x03`, and then save the setting *(on page 70)*, the system will wait for 3 seconds to power on after the ignition is turned on.
`i2cget -y -f 1 0x4a 0x61`	-	Check the ignition power on delay time by minutes	If the delay time is set as 2 minutes, the output will be `0x02`
`i2cset -f -y 1 0x4a 0x61`	For example: `0x02`	Set the ignition power on delay time by minutes	For example, if you enter `i2cset -f -y 1 0x4a 0x61 0x02`, and then save the setting *(on page 70)*, the system will wait for 2 minutes to power on after the ignition is turned on.

Command	Value	Description	Outcome
`i2cget -y -f 1 0x4a 0x62`	-	Check the ignition power on delay time by hours	If the delay time is set as 1 hour, the output will be `0x01`
`i2cset -f -y 1 0x4a 0x62`	For example: `0x01`	Set the ignition power on delay time by hours	For example, if you enter `i2cset -f -y 1 0x4a 0x62 0x01`, and then save the setting *(on page 70)*, the system will wait for 1 hour to power on after the ignition is turned on.

Table 5-3 Checking and Setting Ignition Power Off Delay Time

Command	Value	Description	Outcome
`i2cget -y -f 1 0x4a 0x66`	-	Check the ignition power off delay time by seconds	If the delay time is set as 3 seconds, the output will be `0x03`
`i2cset -f -y 1 0x4a 0x66`	For example: `0x03`	Set the ignition power off delay time by seconds	For example, if you enter `i2cset -f -y 1 0x4a 0x66 0x03`, and then save the setting *(on page 70)*, the system will wait for 3 seconds to power off after the ignition is turned off.
`i2cget -y -f 1 0x4a 0x67`	-	Check the ignition power off delay time by minutes	If the delay time is set as 2 minutes, the output will be `0x02`
`i2cset -f -y 1 0x4a 0x67`	For example: `0x02`	Set the ignition power off delay time by minutes	For example, if you enter `i2cset -f -y 1 0x4a 0x67 0x02`, and then save the setting *(on page 70)*, the system will wait for 2 minutes to power off after the ignition is turned off.
`i2cget -y -f 1 0x4a 0x68`	-	Check the ignition power off delay time by hours	If the delay time is set as 1 hour, the output will be `0x01`
`i2cset -f -y 1 0x4a 0x68`	For example: `0x01`	Set the ignition power off delay time by hours	For example, if you enter `i2cset -f -y 1 0x4a 0x68 0x01`, and then save the setting *(on page 70)*, the system will wait for 1 hour to

Command	Value	Description	Outcome
			power off after the ignition is turned off.

Important:

If no further configuration is required, run the **save settings** *(on page 70)* command to store the configuration; otherwise, the system will revert to its default settings after reboot.

5.1.3. COM Port (RS-232/422/485) Configuration

The system supports either a serial (COM) interface or a digital I/O (DIO) interface, depending on the selected SKU. This section provides commands, sample code, and test programs to configure the COM port transceiver mode and validate serial communication.

The name of the COM port: ttyTHS0

See the following tables as the command and relevant information for checking the COM port status.

Table 5-4 Checking the Status

Command	Outcome	Description
`i2cget -f -y 1 0x4a 0x3c`	• 0bxxx01xxx: RS232 • 0bxxx11xxx: RS422 • 0bxxx10xxx: RS485	Check which mode (RS-232/422/485) is enabled.

See the following table as the command and relevant information for configuring transceiver mode by different communication protocols. You can also refer to Sample Code *(on page 63)* and use the keyword **configTransceiverMode** to locate the corresponding configuration snippet.

Important:
Only **bits [4:3]** are used to select the UART transceiver mode. All other bits must remain unchanged to avoid unintended behavior.

Table 5-5 Transceiver Mode Configuration

Command	Value	Description	Outcome
`i2cset -f -y 1 0x4a 0x3c <value>`	`0bxxx00xxx`	Invalid	UART transceiver mode configured
	`0bxxx01xxx`	RS232	
	`0bxxx11xxx`	RS422	
	`0bxxx10xxx`	RS485	

5.1.3.1. Setting Data Transceiver Direction for COM port

This section describes how to control the data transceiver direction for RS-232, RS-422, and RS-485 communication protocols. You can also refer to Sample Code *(on page 63)* for more information.

- RS-232:

  No direction control is needed.

- RS-422:

  Initial state: Set transceiver to **Transmit Mode** by setting `SYSFS_UART0_RTS = H`

  No further changes are required during communication, as RS-422 supports full-duplex transmission.

- RS-485:

Initial state: Set transceiver to **Receive Mode** by setting `SYSFS_UART0_RTS = L`

Data transmit procedure:

1. Switch to **Transmit Mode** by setting `SYSFS_UART0_RTS = H`
2. **Send data**: `write(fd_sp0, &write_buf, strlen(write_buf));`
3. **Hold Transmit Mode** until transmission completes: `usleep(TRANSCEIVER_HOLDING_TIME);`

   Note:

   - TRANSCEIVER_HOLDING_TIME = 10 (unit: milliseconds)
   - Holding time calculation (approximate):

     {(64 bytes * 10 bits) / 115200 BPS} * 2 ≈ 10 ms

     - Assumes 64 bytes of data
     - "8n1" framing = 10 bits per byte
     - BPS = bit per second
     - * 2 = +50% timing tolerance

4. Switch back to **Receive Mode** by setting `SYSFS_UART0_RTS = L`

5.1.3.2. Sample Code

This section provides the sample code (`UART_common.h`) for configuring the COM port.

```
#include <err.h>
#include <errno.h>
#include <fcntl.h>
#include <stdbool.h>
#include <stdio.h>
#include <stdint.h>
#include <stdlib.h>
#include <string.h>
#include <pthread.h>
#include <termios.h>
#include <unistd.h>
#include <asm-generic/ioctls.h>        /* TIOCGRS485 + TIOCSRS485 ioctl definitions */
#include <linux/i2c.h>
#include <linux/i2c-dev.h>
#include <linux/ioctl.h>
#include <linux/serial.h>
#include <linux/types.h>
#include <sys/ioctl.h>
#include <sys/poll.h>

#define TIOCM_DTR  0x002
#define TIOCM_RTS  0x004
#define TIOCM_ST   0x008
#define TIOCM_SR   0x010
#define TIOCM_CTS  0x020
#define TIOCM_CAR  0x040
#define TIOCM_RNG  0x080
#define TIOCM_DSR  0x100
#define TIOCM_CD   TIOCM_CAR
#define TIOCM_RI   TIOCM_RNG

typedef enum {
         RS232, RS422, RS485
} UART_MODE_t;

/* Transceiver */
const char *MCU_I2C_DEVNODE_PATH = "/dev/i2c-1";
const uint8_t MCU_I2C_ADDR = 0x4a;
const uint8_t MCU_I2C_REG_NUM = 0x3c;

const uint8_t UART0_MODE_MASK = 0x18;         /* Valid bit[3:4] */
```

```
const uint8_t UART0_MODE_RS232_VAL = 0x08;    /* Valid bit[3:4] */
const uint8_t UART0_MODE_RS422_VAL = 0x18;    /* Valid bit[3:4] */
const uint8_t UART0_MODE_RS485_VAL = 0x10;    /* Valid bit[3:4] */

/* SYSFS */
#define SYSFS_UART0_RTS                       "/dev/uart0-rts"
#define SYSFS_UARTX_RTS_GPIO_VAL_H    "1"
#define SYSFS_UARTX_RTS_GPIO_VAL_L    "0"

const uint8_t TRANSCEIVER_HOLDING_TIME = 10;          /* Unit: mS, 64bytes @ 115.2kbps = 5.5mS */

int fd_sp0 = 0;
bool finish = false;                           /* Meet the conditions for exit */
char *UART_devnode;

void configTransceiverMode(const uint8_t MASK, const uint8_t VAL) {
          char i2cbuf[3] = {0};
          int fd_MCU;

          fd_MCU = open(MCU_I2C_DEVNODE_PATH, O_RDWR);
          if (fd_MCU < 0)
                   err(errno, "Tried to open '%s'", MCU_I2C_DEVNODE_PATH);

          if (0 > ioctl(fd_MCU, I2C_SLAVE, MCU_I2C_ADDR))
                   err(errno, "Tried to set device address '0x%02x'", MCU_I2C_ADDR);

          /* Get GPIO of MCU state */
          i2cbuf[0] = MCU_I2C_REG_NUM;
          if (write(fd_MCU, i2cbuf, 1) != 1) {
                   perror("Failed to write address of transceiver mode");
                   exit(-EIO);
          }

          if (read(fd_MCU, i2cbuf, 1) != 1) {
                   perror("Failed to read transceiver mode");
                   exit(-EIO);
          }

          /* Set mode of related transceiver */
          i2cbuf[1] = i2cbuf[0] & ~(MASK);
          i2cbuf[1] = i2cbuf[1] | (VAL & MASK);
          i2cbuf[0] = MCU_I2C_REG_NUM;
          if (write(fd_MCU, i2cbuf, 2) != 2) {
                   perror("Failed to write to the I2C device");
                   exit(-EIO);
          }
          /* Using I2C Read, equivalent of i2c_smbus_read_byte(file) */
          if (read(fd_MCU, i2cbuf, 1) != 1) {
                   perror("Failed to read to the I2C device");
                   exit(-EIO);
          }
}
void configUART_mode(void) {
         struct serial_rs485 rs485conf = {0};
         struct termios tio;

         memset(&tio, 0, sizeof(tio));
         tio.c_iflag = 0;
         tio.c_oflag = 0;
         tio.c_cflag = CS8 | CREAD | CLOCAL;
         tio.c_lflag = 0;
         /* Block mode */
         tio.c_cc[VMIN] = 1;
         tio.c_cc[VTIME] = 5;
         tio.c_cflag |= CRTSCTS;                      /* Use RTS/CTS */

         cfsetospeed(&tio, B115200);

         fd_sp0 = open(UART_devnode, O_RDWR);
         if (fd_sp0 < 0) {
                  printf("Error: Can't open: %s\n", UART_devnode);
#if 1
         /* Read the current state of the RS-485 options with ioctl. */
         if (ioctl(fd_sp0, TIOCGRS485, &rs485conf) < 0)
                  printf("Error: TIOCGRS485 ioctl not supported.\n");
         /*
          * On Linux RTS is set when a tty device is open. This is not good
          * when RTS is used for RS485. In this case RTS will prevent the
```

```
          * transceiver to receive any data. So here RTS will be cleared.
          */
         int rtsflag = TIOCM_RTS;
         int ret = ioctl(fd_sp0, TIOCMBIC, &rtsflag);
         if (0 > ret) {
                 printf("Can not set rts pint, ioctl failed %d, (errno: %d)\n", ret, errno);
                 exit(-EIO);
         }

         /* Enable RS485 mode: */
         rs485conf.flags |= SER_RS485_ENABLED;

         /* Set logical level for RTS pin equal to 1 when sending: */
         rs485conf.flags |= SER_RS485_RTS_ON_SEND;

         /* set logical level for RTS pin equal to 0 after sending: */
         rs485conf.flags &= ~(SER_RS485_RTS_AFTER_SEND);

         /* Set rts delay before send, if needed: */
         rs485conf.delay_rts_before_send = 0;

         /* Set rts delay after send, if needed: */
         rs485conf.delay_rts_after_send = 0;

         /* Set this flag if you want to receive data even whilst sending data */
         rs485conf.flags |= SER_RS485_RX_DURING_TX;

         /* Write the current state of the RS-485 options with ioctl. */
         if (ioctl(fd_sp0, TIOCSRS485, &rs485conf) < 0) {
                 //printf("Error: TIOCSRS485 ioctl not supported.\n");
         }
         /* set the termios so the rs485 settings are applied */
         tcsetattr(fd_sp0, TCSANOW, &tio);
         fcntl(fd_sp0, F_SETFL, 0);
#endif
}

void* UART_read_data(void *arg) {
         while (!finish) {
                 char read_buf[64] = {0};
                 int n = read(fd_sp0, &read_buf, sizeof(read_buf));

                 if (n < 0)
                         printf("Error reading from %s\n", UART_devnode);
                 else
                         printf("UART reading: %s\n", read_buf);
                 usleep(100 * 1000);
         }
         pthread_exit(NULL);
}

void UART_write_data_RTSctl(const UART_MODE_t UARTmode, const char *RTS_SYSFS) {
          int fd_RTS = 0;

          if (NULL != RTS_SYSFS) {
                   fd_RTS = open(RTS_SYSFS, O_RDWR);
                   if ( 0 > fd_RTS) {
                            printf("Error: Can't open: %s\n", RTS_SYSFS);
                            exit(-EINVAL);
                   }
          }
          switch (UARTmode) {
                   case RS232:
                            break;
                   case RS422:
                            if (0 > write(fd_RTS, SYSFS_UARTX_RTS_GPIO_VAL_H, sizeof(SYSFS_UARTX_RTS_GPIO_VAL_H))) {
                                     printf("Error: Can't write: %s\n", RTS_SYSFS);
                                     exit(-EINVAL);
                            }
                            break;
                   case RS485:
                            if (0 > write(fd_RTS, SYSFS_UARTX_RTS_GPIO_VAL_L, sizeof(SYSFS_UARTX_RTS_GPIO_VAL_L))) {
                                     printf("Error: Can't write: %s\n", RTS_SYSFS);
                                     exit(-EINVAL);
                            }
                            break;
                   default:
                            printf("Error: UART mode invalid: %d\n", UARTmode);
```

```
                        exit(-EINVAL);
        }
        printf("Please enter:\n");
        do {
                char write_buf[128] = {0};

                fgets(write_buf, sizeof(write_buf), stdin);          /* keyboard key-in */
                if(strcmp(write_buf, "q\n") == 0 || strcmp(write_buf, "Q\n") == 0)
                        finish = true;

                if (finish) {
                        usleep(1000 * 1000);                        /* Waiting reading thread finish */
                        break;                                      /* while (true) */
                }

                if (RS485 == UARTmode)
                        write(fd_RTS, SYSFS_UARTX_RTS_GPIO_VAL_H, sizeof(SYSFS_UARTX_RTS_GPIO_VAL_H));/* Assume error
 free */

                write(fd_sp0, &write_buf, strlen(write_buf));
                if (RS485 == UARTmode) {
                        /* Holding transceiver as transmitting mode, waiting data transmit finished */
                        usleep(TRANSCEIVER_HOLDING_TIME * 1000);
                        write(fd_RTS, SYSFS_UARTX_RTS_GPIO_VAL_L, sizeof(SYSFS_UARTX_RTS_GPIO_VAL_L));/* Assume error
 free */
                }
        } while (true);
        printf("Closing %s\n", UART_devnode);
        if (fd_sp0)
                if (close(fd_sp0) < 0)
                        printf("Error: Can't close: %s\n", UART_devnode);
        if (fd_RTS)
                if (close(fd_RTS) < 0)
                         printf("Error: Can't close: %s\n", RTS_SYSFS);
}
void UART_write_data(const UART_MODE_t UARTmode) {
UART_write_data_RTSctl(UARTmode, NULL);
}
```

5.1.3.3. RS-232 Test Program

This section provides a sample test program (`rs232-demo.c`) for validating RS-232 communication.

```
#include "UART_common.h"

int main(int argc, char* argv[]) {
        UART_devnode="/dev/ttyTHS0";

        configTransceiverMode(UART0_MODE_MASK, UART0_MODE_RS232_VAL);
        configUART_mode();

        pthread_t UART_read_thread;
        if (pthread_create(&UART_read_thread, NULL, UART_read_data, NULL) != 0)
                printf("Error: pthread create UART read thread\n");

        UART_write_data(RS232);

        return 0;
}
```

5.1.3.4. RS-422 Test Program

This section provides a sample test program (`rs422-demo.c`) for validating RS-422 communication.

```
#include "UART_common.h"

int main(int argc, char* argv[]) {
        UART_devnode="/dev/ttyTHS0";

        configTransceiverMode(UART0_MODE_MASK, UART0_MODE_RS422_VAL);
        configUART_mode();

        pthread_t UART_read_thread;
        if (pthread_create(&UART_read_thread, NULL, UART_read_data, NULL) != 0)
                printf("Error: pthread create UART read thread\n");
```

```
        UART_write_data_RTSctl(RS422, SYSFS_UART0_RTS);

        return 0;
}
```

5.1.3.5. RS-485 Test Program

This section provides a sample test program (`rs485-demo.c`) for validating RS-485 communication.

```
#include "UART_common.h"

int main(int argc, char* argv[]) {
        UART_devnode="/dev/ttyTHS0";

        configTransceiverMode(UART0_MODE_MASK, UART0_MODE_RS485_VAL);
        configUART_mode();

        pthread_t UART_read_thread;
        if (pthread_create(&UART_read_thread, NULL, UART_read_data, NULL) != 0)
                printf("Error: pthread create UART read thread\n");

        UART_write_data_RTSctl(RS485, SYSFS_UART0_RTS);

        return 0;
}
```

5.1.4. CAN FD Configuration

The name of the CAN FD port: can0

See the following table as the available commands list for the CAN FD port.

Table 5-6 Initialization & Setup

Command	Description	Outcome
`sudo can_set`	Enable the CAN bus mode	CAN Bus set completed
`sudo ip link set down can0`	Disable the transmission function of the CAN bus	N/A
`sudo ip link set can0 type can bitrate 1000000 dbitrate 2000000 fd on`	Set the standard bit rate for the CAN bus as 1 Mbps and the data phase bit rate for CAN FD (Flexible Data Rate) as 2 Mbps. Enable CAN FD for faster transmission rate and larger data payloads (up to 64 bytes per frame).	N/A
`sudo ip link set up can0`	Enable the transmission function of the CAN bus	N/A

Table 5-7 Data Reception

Command	Description	Outcome
`candump can0 &`	Receive data	N/A

Table 5-8 Data Transmission

Command	Description	Outcome
`cansend can0 123#abcdabcd`	Send data	N/A

5.1.5. DIO Configuration

The system supports either a serial (COM) interface or a digital I/O (DIO) interface, depending on the selected SKU. This section provides commands to configure the DIO interface, which includes one programmable digital input (DI) and one digital output (DO). These signals can be configured to exchange data with connected peripheral devices.

Note:
The DIO pins will be reset to default settings after a cold boot.

The general formats of the DIO commands are as follows:

`i2cget -f -y <i2c_num> <device_addr> <reg_addr>`: Used to **check** the current state of DI or DO channel.

`i2cset -f -y <i2c_num> <device_addr> <reg_addr> <value>`: Used to **change or save** the settings of DI or DO channel.

See the following as the DI data register table:

Table 5-9 DI Data Register – 0x30

Bit	Pin	Value
0	DI_1	• Low: 0 • High: 1

See the following as the DO data register table:

Table 5-10 DO Data Register – 0x31

Bit	Pin	Value
0	DO_1	• Low: 0 • High: 1

5.1.5.1. DIO Commands

See the following table as the available commands list for the DIO port.

Table 5-11 Checking Digital Input (DI) Status

Command	Outcome	Description
`i2cget -f -y 1 0x4A 0x30`	Value of DI status	Get the status of the digital input pin

Table 5-12 Examples of Configuring Digital Output (DO)

Command	Value	Description
`i2cset -f -y 1 0x4A 0x31 <value>`	`0xf`	Set DO1 as High
	`0x0`	Set DO1 as Low

5.1.6. Save the Settings

Enter the following command to preserve the specified configuration. If this command is not executed, after the system reboots, the specified configuration will be restored to its default state.

Table 5-13 Saving the Setting

Command	Description	Outcome
`i2cset -f -y 1 0x4a 0x28 0x02`	Save the setting.	The specified setting will be saved and executed.

5.2. System Recovery

This section describes how to recover the system when needed.

5.2.1. Hardware and System Requirements

A host computer and some accessories are required to perform a system recovery. Before you begin, check the following list as the hardware and system requirements for the host computer and accessories.

Host Computer:

- Memory size: 4 GB or above
- Storage space: 256 GB or above
- Recommended OS: Ubuntu 22.04

Accessories:

- A USB Type-C cable
- An HDMI cable
- A monitor supporting HDMI inputs

5.2.2. Configuring Host Computer

Follow the procedures below to configure the host computer.

1. Connect the host computer to the Internet.
2. Open the terminal window.
3. Enter the following commands to install the required dependencies.

```
$ sudo apt update
$ sudo apt install -y abootimg bzip2 libxml2-utils nfs-kernel-server sshpass
```

5.2.3. Downloading BSP Image & Checksum Files

You must install a board support package (BSP) image on the host computer before performing the system recovery. Follow the procedures below to download the BSP image.

1. Download the BSP image and accompanied SHA256 checksum files from SINTRONES® website or contact our sales representative for the files.
2. Save the downloaded files in the host computer.

Note:
Do not extract the compressed `.tbz2` file.

5.2.4. Setting the System in Recovery Mode

Follow the procedures to set the system in recovery mode and connect it to the host computer.

1. Ensure the system is powered off.
2. Locate the **SERVICE PORT** Type-C port on the rear panel.
3. Plug one end of a USB Type-C cable into the port, and then plug the other end of the cable to the host computer.
4. Locate the **RECOVERY** button on the rear panel.
5. Press and hold the **RECOVERY** button, at the same time, connect the system to a power source and enable ignition if any.

Note:
If the system is not connected with any ignition system, simply press the power button.

6. After powering on the system, hold the **RECOVERY** button for more than 5 seconds, and then release it.

5.2.5. Executing Recovery

Follow the steps below ro run the recovery commands on the host computer and flash the BSP image to the system.

1. On the host computer, enter the following command in the terminal to check if the system is set in recovery mode.

```
$ lsusb | grep NVIDIA
```

2. The following strings with the VID/PID info appear, indicating the system is in recovery mode. If these strings do not appear, repeat the steps in Setting the System in Recovery Mode *(on page 71)*.

Figure 5-2 An **Example** of Strings with the VID/PID Info

Note:
The VID/PID varies depending on different models.

Important:
Ensure the system is in recovery mode before you peform the flashing process.

3. Navigate to the folder that contains the downloaded BSP image and SHA256 checksum files.
4. Enter the following command to verify the integrity of the BSP image.

```
$ sha256sum -c <ProjectName-KernelVersion-JetpackVersion-BuildSystem-BuildVersion-PackageVersion>.tbz2.sha256
```

Note:
Replace the texts within the brackets with the actual BSP file name, for example, `IBOX-602-5.15.148-6.2-ubuntu22.04-R1.00-00`

Tip:
After entering the first few characters of a command or file name, you can enter the **[tab]** key to auto-complete the command or filename.

5. Enter the following command to decompress the BSP image.

```
$ sudo tar -jxvf <ProjectName-KernelVersion-JetpackVersion-BuildSystem-BuildVersion-PackageVersion>.tbz2
```

6. After the BSP image is decompressed, a folder will be automatically generated. Its name matches the BSP file name and follows this format:

```
<ProjectName-KernelVersion-JetpackVersion-BuildSystem-BuildVersion-PackageVersion>
```

7. Enter the following command to navigate to the folder.

```
$ cd <ProjectName-KernelVersion-JetpackVersion-BuildSystem-BuildVersion-PackageVersion>
```

8. Enter the following command to flash the BSP image to the system.

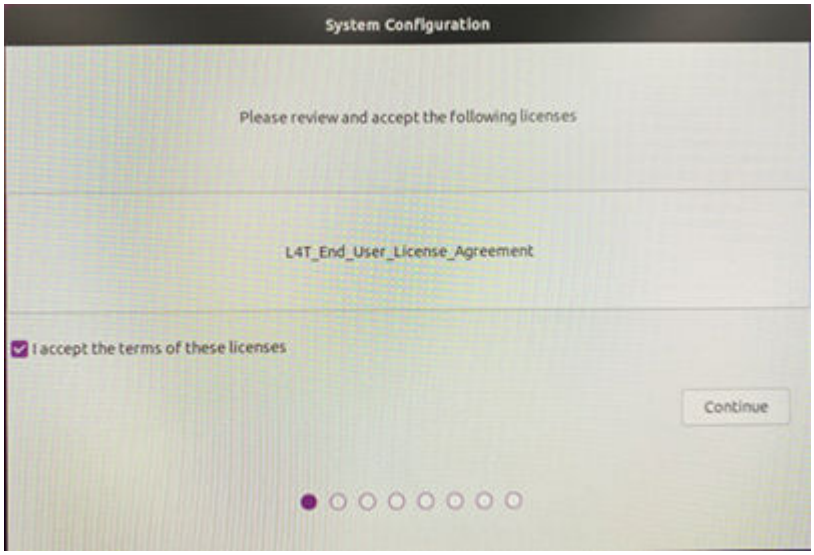
```
$ sudo ./ibox-604_nx_flash.sh
```

9. When the BSP image is successfully flashed to the system, the system will reboot and the configuration window will then appear on the connected monitor screen.

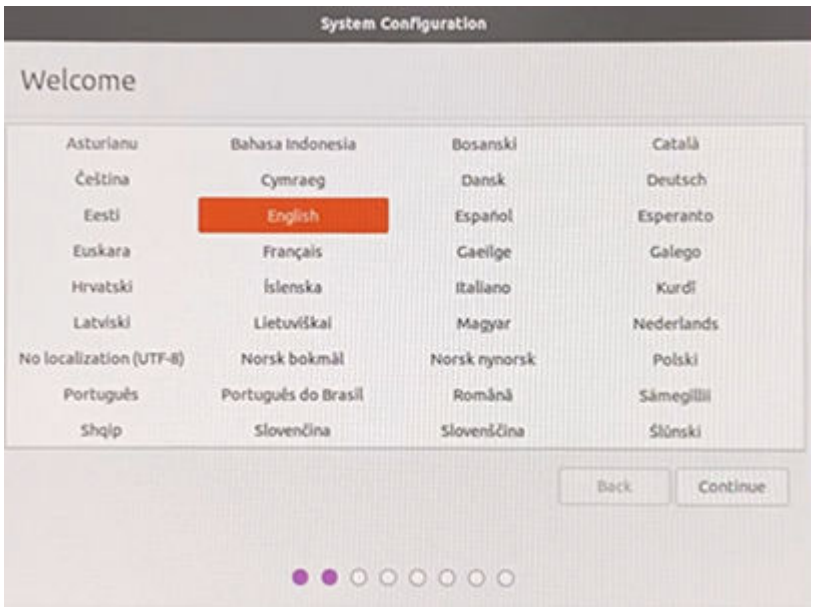
5.2.6. Configuring the Recovered System

After the system is recovered, initial settings must be configured prior to using the system. Follow the procedures below to start the configuration.

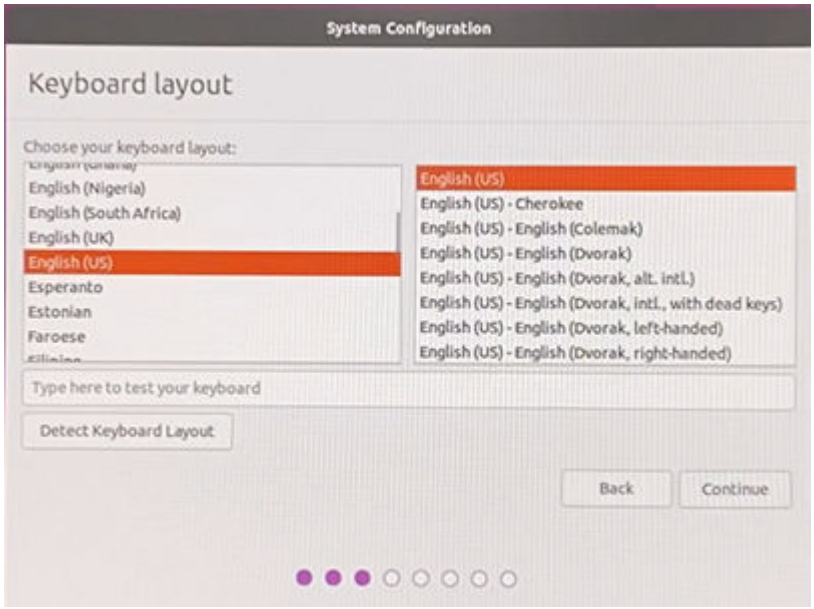
1. After the system reboot, the **System Configuration** wizard appears to guide users to complete the initial basic settings for the Linux for Tegra (L4T) platform.
2. Select **I accept the terms of these licenses** and then select **Continue**.



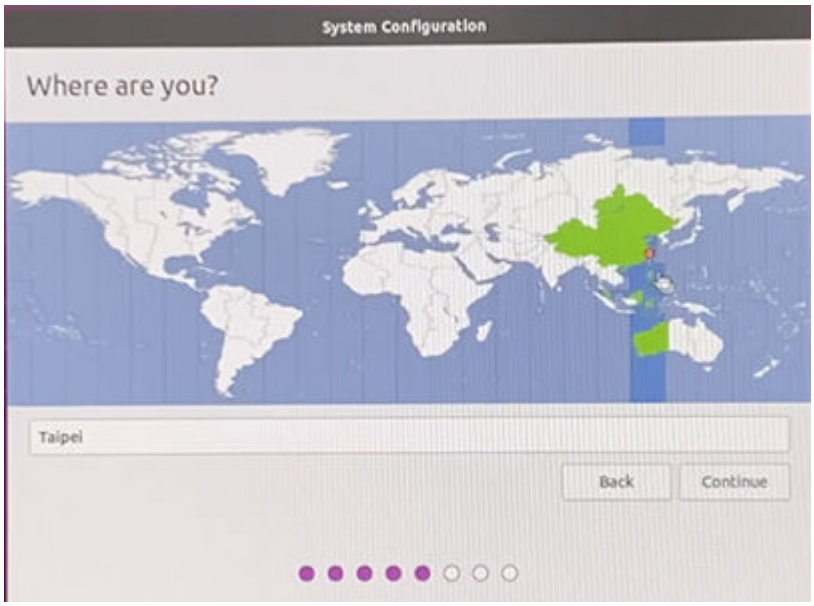
3. Select the preferred language setting and then select **Continue**.



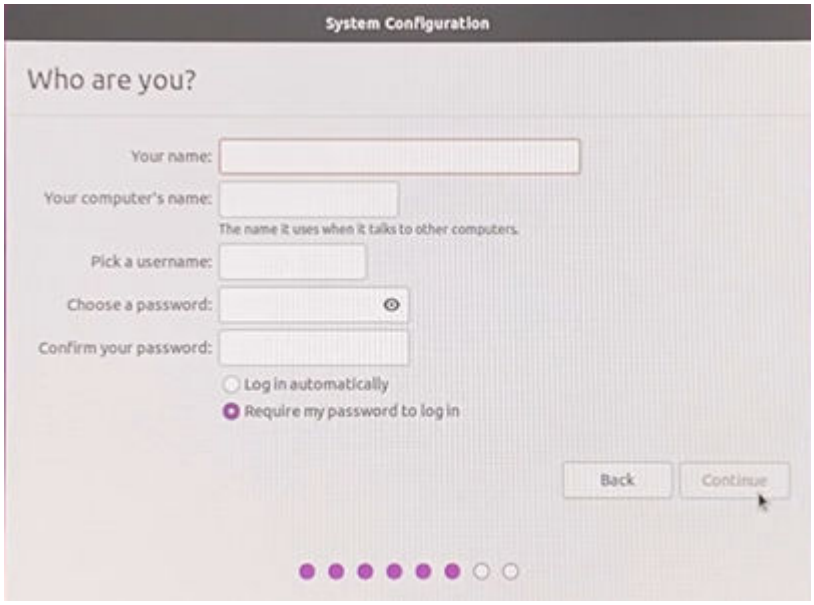
4. Select the preferred keyboard layout and then select **Continue**.



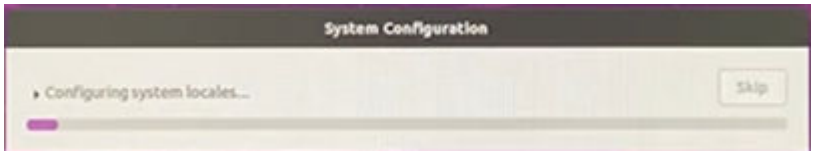
5. Select your location and then select **Continue**.

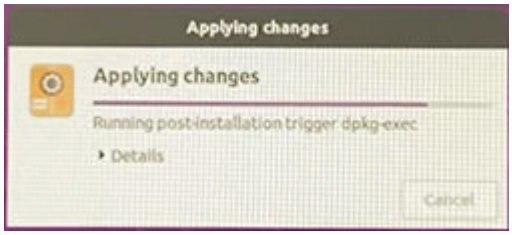


6. Specify the credentials such as a username and a password to create a user account. It is suggested to select **Require my password to log in** for security reasons. Select **Continue** to proceed to next step.



7. Choose **Install Chromium Browser** or **Do not install...** based on your needs, and then select **Continue**.
8. The system starts running the specified configuration and will reboot to complete the system configuration.





9. The **NVIDIA** logo appears and then the restored system will enter the welcome screen, ready to use now.

6. Appendix

Table 6-1 Decimal to Hexadecimal

Time	0	1	2	3	4	5	6	7	8	9
0	None	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	0x09
10	0x0a	0x0b	0x0c	0x0d	0x0e	0x0f	0x10	0x11	0x12	0x13
20	0x14	0x15	0x16	0x17	0x18	0x19	0x1a	0x1b	0x1c	0x1d
30	0x1e	0x1f	0x20	0x21	0x22	0x23	0x24	0x25	0x26	0x27
40	0x28	0x29	0x2a	0x2b	0x2c	0x2d	0x2e	0x2f	0x30	0x31
50	0x32	0x33	0x34	0x35	0x36	0x37	0x38	0x39	0x3a	0x3b