Edge AI Applied Computing

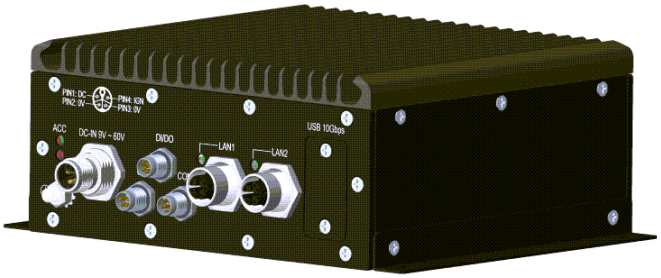
SINTRONES

IBOX-600-IP66

User Manual

Version 1.2

NVIDIA JETSON



SINTRONES

Revision History

Version	Date	Description of Changes
1.0	2025-04-14	Initial release.
1.1	2025-07-08	• Added "DSU JST Connector" info in 4. Expansion
1.2	2025-12-05	Updated the command to flash a BSP image in Section 5.2.5, *Executing System Recovery*

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 70ºC.
- 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

This chapter introduces SINTRONES® IBOX-600-IP66 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)
- NVIDIA® Jetson Orin™ Nano 4GB (512 CUDA cores + 6-core ARM Cortex-A78AE CPU + 4GB LPDDR5)

Network

- 1 x Intel® 2.5GbE, 1 x GbE (Integrated in SoM)

Security

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

Watchdog

- Automatic reset for unresponsive system

Interface

Video

- 1 x HDMI® Type-A with support for 4K resolution

Audio

- 1 x HD audio output from the HDMI®

Ethernet

- 2 x M12 X-coded connector (1 x 2.5GbE & 1 x GbE)

USB

- 2 x USB 3.2 Type-A (with a protection cover)

  *Two ports share 10Gbps bandwidth

DIO

- 1 x M8 A-coded connector for 3 x DI / 4 x DO (12V/100mA)

Serial Port

- 1 x M8 A-coded connector for RS-232/422/485

CAN Bus

- 1 x M8 A-coded connector for CAN FD

Mgmt. Port

- 1 x USB Type-C for system recovery only (without 5V output)

SIM Card

- 1 x Nano SIM card slot

Antenna

- 5 x Pre-cut hole for external SMA antenna (with covers)

Internal Expansion

M.2

- 1 x M.2 3042/52 Key B slot (USB 3.0) with a Nano SIM card slot
- 1 x M.2 2230 Key E slot with support for PCIe and USB 2.0

mPCIe

- 1 x mPCIe full-size slot with support for USB 2.0

Storage

Type

- 1 x M.2 2280 Key M slot for NVMe SSD (Pre-installed in system with BSP)

Power

Power Input

- DC 9V ~ 60V via a M12 K-coded connector

Power Protection

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

Power Management

- Smart power and ignition management for various vehicles

UPS (Optional)

- Backup battery for system power backup

  *UPS backup time varies depending on actual overall system power consumption.
  **Operating temperature will be -20ºC ~ 60ºC with the battery kit.
  Patent No.: M447854 (Built-in Battery)

RTC Battery

- High-capacity coin cell battery for RTC

Software

Operating System / BSP

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

Environmental

Operating Temp.

- -25°C ~ 70°C (-13°F ~ 158°F) with 0.6 m/s airflow

*Operating temperature varies by accessories installed.

Storage Temp.

- -40°C ~ 80°C (-40°F ~ 176°F)

Relative Humidity

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

Vibration

- Random - IEC60068-2-64, random, 2.5G@5~500Hz, 1hr/axis with SSD
- MIL-STD-810G, Method 514.6, Procedure I, Category 4 with SSD

Shock

- MIL-STD-810G, Method 516.6, Procedure I, Trucks and semi-trailers = 15G (11ms) with SSD

Certification / Standard

Certification

- CE, FCC Class A, E-Mark

Standard

- EN 50155, EN 45545-2 (R25)

Mechanical

Construction

- Aluminum alloy

Mounting

- Wall

Weight

- 1.63 kg (3.60 lb)

Dimensions (L x W x H)

- 150 x 135 x 66 mm (5.91 x 5.32 x 2.60 in.)

Ingress Protection

- IP66

Table 1-2 Ordering Information

Model Number	IBOX-600-IP66-ONX16	NVIDIA® Jetson Orin™ NX 16GB / Model IP66
	IBOX-600-IP66-ONX8	NVIDIA® Jetson Orin™ NX 8GB / Model IP66
	IBOX-600-IP66-ON8	NVIDIA® Jetson Orin™ Nano 8GB / Model IP66
	IBOX-600-IP66-ON4	NVIDIA® Jetson Orin™ Nano 4GB / Model IP66
Description	NVIDIA® Jetson Orin™ NX / Nano SoM w/ 1x HDMI® / 2 x LAN / 1 x CAN FD / DC 9V ~ 60V / IP66 Edge AI Computer	

Table 1-3 Mandatory Accessory

Storage	M.2 2280 NVMe SSD 240GB / 480GB / 960GB available for selection **Note:** It is recommended to install an NVMe SSD with at least 480GB storage capacity for optimal data transmission and processing.

Table 1-4 Optional Accessories

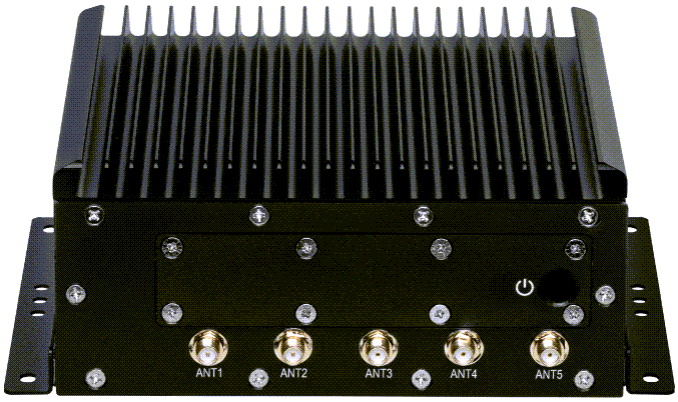
Wi-Fi	M.2 2230 Wi-Fi Module
WWAN	M.2 3042/52 WWAN Modem
GPS	mPCIe GPS module
UPS	BAT-2300v2 (Operating Temp.: -20°C ~ 60°C /-4°F ~ 140°F)

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

1.2. Product Photos

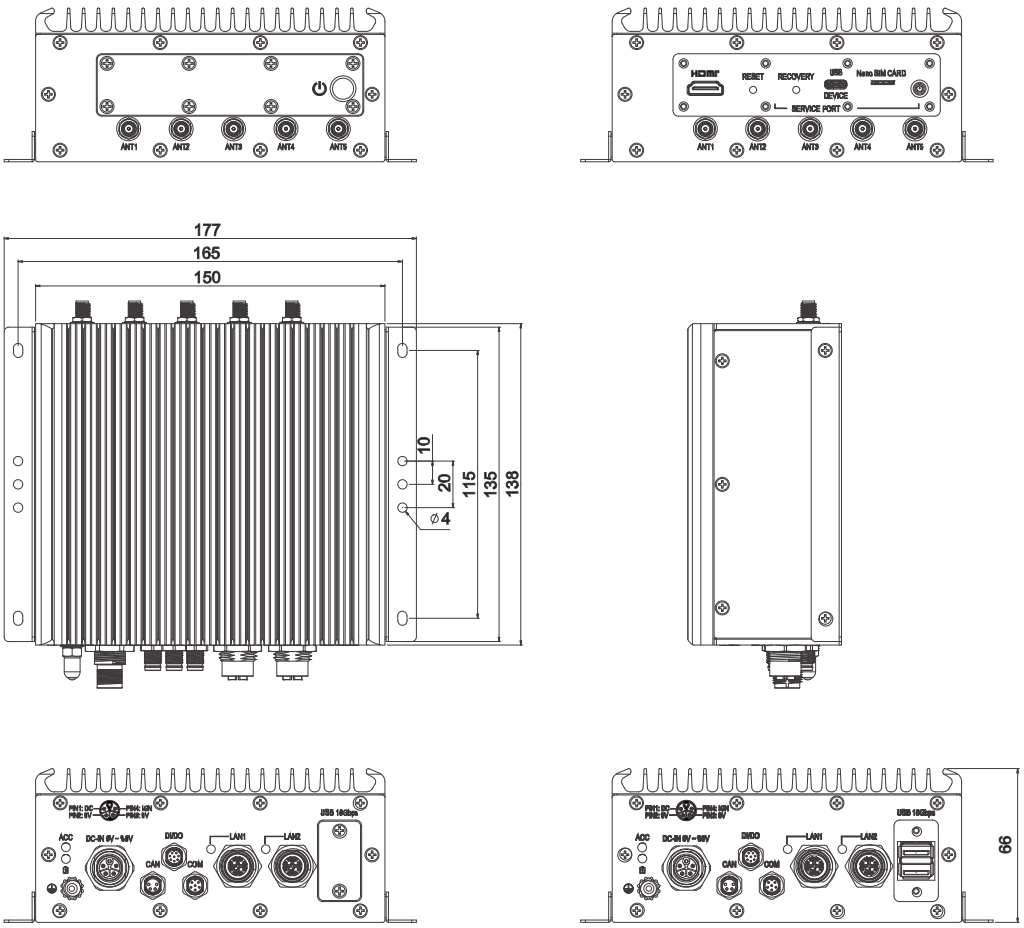
Figure 1-1 Front View of IBOX-600-IP66

Figure 1-2 Rear View of IBOX-600-IP66



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-600-IP66

Item	Photo	Quantity	Description
IBOX-600-IP66	See Product Photos *(on page 10)*	1	The edge AI computer
Screw I (Type M2*5L ISO)		2	Used to fasten an mPCIe card
Screw I (Type M2.5*5L)		2	Used to fasten an M.2 module
Screw I (Type M2.5*10L)		2	Used to fasten a heatsink for an M.2 3042 or 3052 module.
Standoff screw (H75D50)		2	Served as the mounting screw-holes to secure the screw used to fasten a heatsink for an M.2 3042 or 3052 module
M.2 WWAN heatsink Type 12		1	Used to transfer heat from an installed M.2 3042 or 3052 module
Thermal pad (18x30x1.0T mm)		1	Used to transfer heat from an installed M.2 3042 or 3052 module
Thermal pad (30x34x1.0T mm)		1	Used to transfer heat from an installed M.2 3042 or 3052 module

Item	Photo	Quantity	Description
Power cable for IBOX-600-IP66		1	Used to connect the computer with a M12 K-coded connector to a DC power supply

1.5. Power Consumption

The power consumption varies depending on the built-in NVIDIA® Jetson module and the installed JetPack version.

See the following sections for power consumption measured based on JetPack 5.1.3 or 6.2 installed on different Jetson modules.

.

1.5.1. Power Consumption Based on JetPack 5.1.3

Table 1-6 IBOX-600-IP66-ONX16

Mode	Input Voltage				
	9V	12V	24V	48V	60V
25W (SoM Power Mode)	35.55W	34.32W	34.08W	35.04W	37.80W
Idle	11.7W	12.72W	12.96W	14.88W	16.80W
Standby	1.80W	2.52W	4.08W	6.24W	7.20W

Table 1-7 IBOX-600-IP66-ONX8

Mode	Input Voltage				
	9V	12V	24V	48V	60V
20W (SoM Power Mode)	30.24W	30.48W	30.24W	34.08W	34.80W
Idle	9.72W	9.24W	10.80W	12.96W	14.40W
Standby	2.25W	2.64W	3.12W	5.28W	6.00W

Table 1-8 IBOX-600-IP66-ON8

Mode	Input Voltage				
	9V	12V	24V	48V	60V
15W (SoM Power Mode)	28.53W	27.84W	29.04W	32.16W	32.40W

Mode	Input Voltage				
	9V	12V	24V	48V	60V
Idle	8.91W	9.12W	9.84W	12.96W	13.20W
Standby	2.07W	3.60W	4.32W	7.20W	7.20W

Table 1-9 IBOX-600-IP66-ON4

Mode	Input Voltage				
	9V	12V	24V	48V	60V
10W (SoM Power Mode)	26.16W	26.16W	27.36W	29.28W	30.00W
Idle	9.60W	9.60W	10.32W	12.96W	13.80W
Standby	2.76W	2.76W	3.36W	4.80W	5.40W

1.5.2. Power Consumption Based on JetPack 6.2

Table 1-10 IBOX-600-IP66-ONX16

Mode	Input Voltage				
	9V	12V	24V	48V	60V
40W (SoM Power Mode)	42.39W	43.08W	42.96W	46.56W	48.60W
Idle	11.97W	12.24W	12.96W	14.88W	16.20W
Standby	3.42W	3.36W	4.08W	6.24W	7.20W

Note:
For the IBOX-600-IP66-ONX16 run on NVIDIA® JetPack 6.2, the performance efficiency will decrease when the operating temperature exceeds 70°C (158°F). It is recommended to keep the operating temperature below 60°C (140°F) for optimal performance.

Table 1-11 IBOX-600-IP66-ONX8

Mode	Input Voltage				
	9V	12V	24V	48V	60V
40W (SoM Power Mode)	40.77W	39.60W	40.08W	43.68W	45.00W
Idle	13.59W	13.56W	14.16W	16.32W	17.40W
Standby	2.70W	2.88W	3.36W	5.28W	6.00W

Note:
The above test results are measured based on NVIDIA® JetPack 6.2 with 12V voltage supply for the internal Jetson Orin™ NX module.

Table 1-12 IBOX-600-IP66-ON8

Mode	Input Voltage				
	9V	12V	24V	48V	60V
25W (SoM Power Mode)	32.76W	33.00W	33.36W	35.52W	36.60W

Mode	Input Voltage				
	9V	12V	24V	48V	60V
Idle	10.44W	10.68W	11.52W	13.92W	15.60W
Standby	3.24W	3.24W	3.84W	5.76W	6.60W

Table 1-13 IBOX-600-IP66-ON4

Mode	Input Voltage				
	9V	12V	24V	48V	60V
25W (SoM Power Mode)	31.05W	32.40W	32.40W	34.08W	36.00W
Idle	10.17W	10.56W	11.28W	13.44W	15.00W
Standby	3.06W	3.24W	3.84W	5.76W	6.60W

Note:
The above test results are measured based on NVIDIA® JetPack 6.2 with 5V voltage supply for the internal Jetson Orin™ Nano module.

2. Getting Started

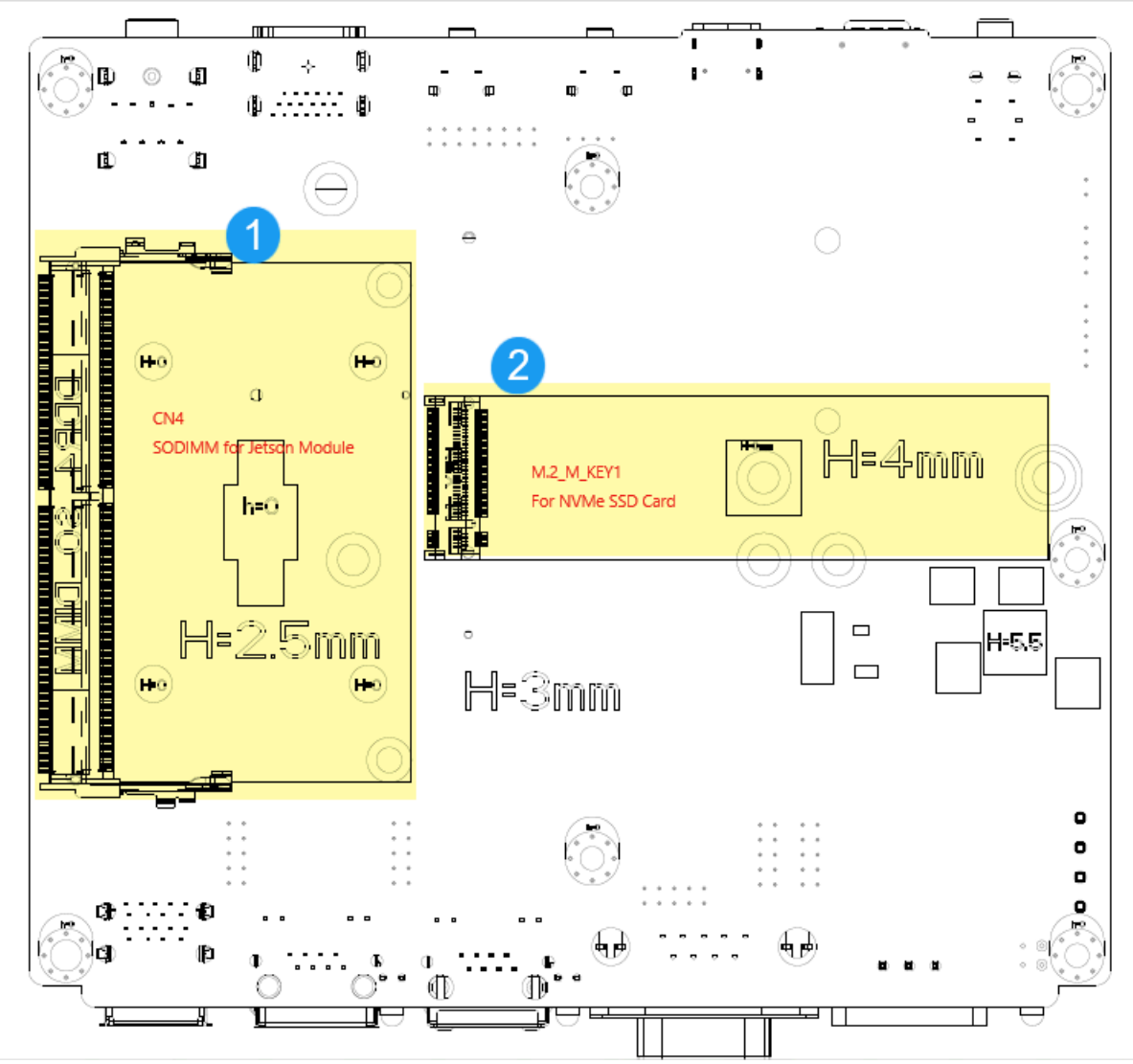
Topics in this chapter include:

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 SSD, or install an empty one without preparing any backup image in advance.

Figure 2-1 Bottom View of Mainboard



Item	Internal Connector	Description	Specification
1	SODIMM for Jetson Module	NVIDIA® Jetson module integrates CPU, GPU, and memory with pre-installed JetPack developer tools.	Information about Jetson Modules *(on page 20)*
2	M2M1	M.2 2280 Key M slot used for installing an NVMe SSD	M.2 Key M Slot *(on page 20)*

2.1.1. Information about Jetson Modules

Table 2-1 NVIDIA® Jetson Orin™ NX/Nano Modules Pre-installed in IBOX-600-IP66

Series	Jetson Orin NX Series		Jetson Orin Nano Series	
Model	Jetson Orin NX 16GB	Jetson Orin NX 8GB	Jetson Orin Nano 8GB	Jetson Orin Nano 4GB
GPU	1024-core NVIDIA® Ampere architecture GPU with 32 Tensor Cores			512-core NVIDIA® Ampere architecture GPU with 16 Tensor Cores
CPU Frequency	2.0 GHz		1.5 GHz	
Power Consumption	25W	20W	15W	10W
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	4GB 64-bit LPDDR5
Storage	Supports external NVMe SSD (Pre-installed JetPack)			

2.1.2. M.2 Key M Slot

M.2 Key M Slot	Description
Size	NGFF 2280 / 75 Pin
Type	M.2 Key M H:8.5mm
Location	See SoM and SSD Info *(on page 19)* for the information.
Drawing	

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

Pin	Signal	Pin	Signal
1	GND	2	3VSB
3	GND	4	3VSB
5	PCIE0_RX3_N	6	NC
7	PCIE0_RX3_P	8	NC
9	GND	10	NC
11	PCIE0_TX3_N	12	3VSB
13	PCIE0_TX3_P	14	3VSB
15	GND	16	3VSB
17	PCIE0_RX2_N	18	3VSB
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
37	PCIE0_TX1_P	38	NC
39	GND	40	JS_I2C2_SCL
41	PCIE0_RX0_N	42	JS_I2C2_SDA
43	PCIE0_RX0_P	44	M2M_ALERT
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

Pin	Signal	Pin	Signal
67	NC	68	NC
69	NGFF3_PEDET	70	3VSB
71	GND	72	3VSB
73	GND	74	3VSB
75	GND		

2.2. System Setup

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

Important:

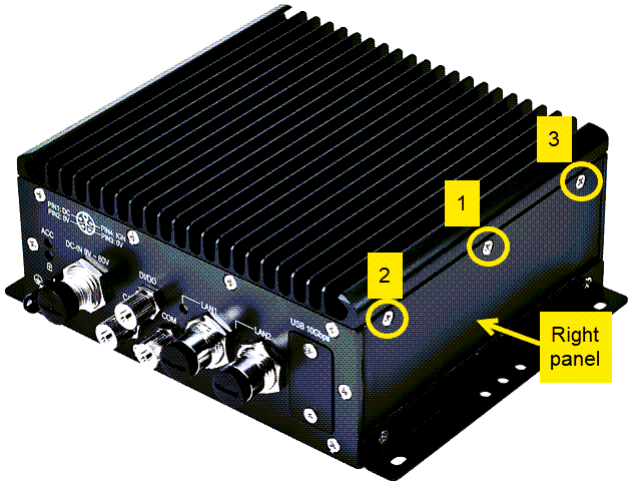
- Ensure the device is not connected to any power source such as a power adapter or a battery.
- Prior to installing any modules on the mainboard, always touch an unpainted and grounded metal object or wear a grounded anti-static wrist strap to prevent electrostatic discharge (ESD).

See the following steps to start the installation:

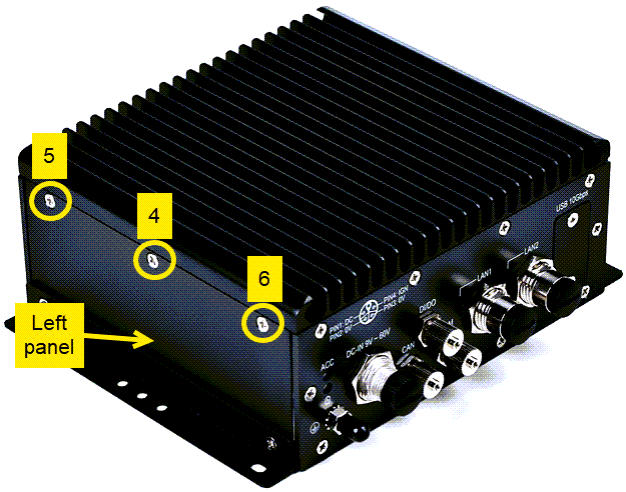
2.2.1. Removing the Bottom Cover

The bottom cover must be removed prior to installing expansion modules or an optional backup battery.

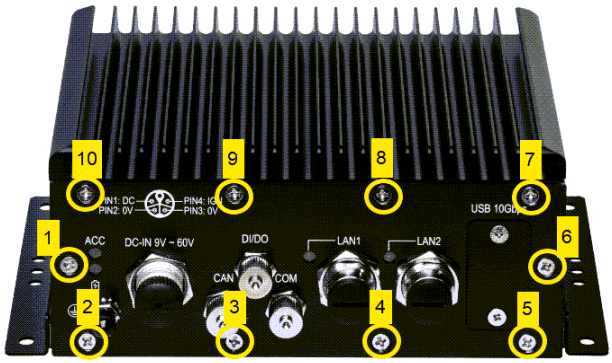
1. Facing the front panel of the system, the side cover on the right side is the right panel. On the right panel, remove the three screws in the order indicated in the following figures.



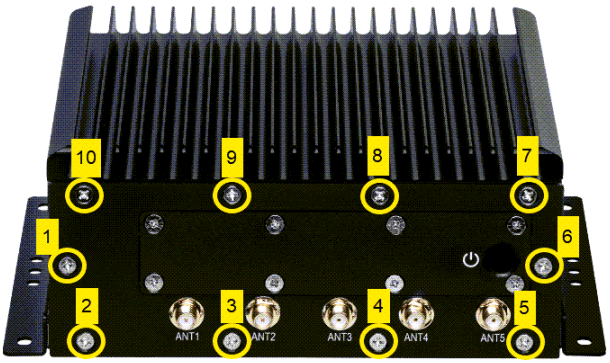
2. Facing the front panel of the system, the side cover on the left side is the left panel. On the left panel, remove the three screws in the order indicated in the following figures.



3. Remove the ten screws on the front panel in the order indicated in the following figure.



4. Remove the ten screws on the rear panel in the order indicated in the following figure.



5. After removing all the specified screws, gently lift the bottom cover and place it carefully.

Important:

- When reassembling the system, ensure you follow the order from Step 1 to Step 4.
- In order to maintain the system's IP66 waterproof funcationality, ensure you fasten the screws by the specified order given in the figures above.

Figure 2-2 Bottom Cover of IBOX-600-IP66

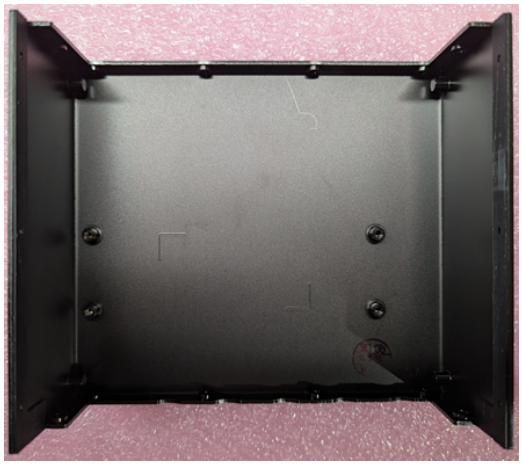


Figure 2-3 Mainboard of IBOX-600-IP66

2.2.2. Installing Expansion Modules

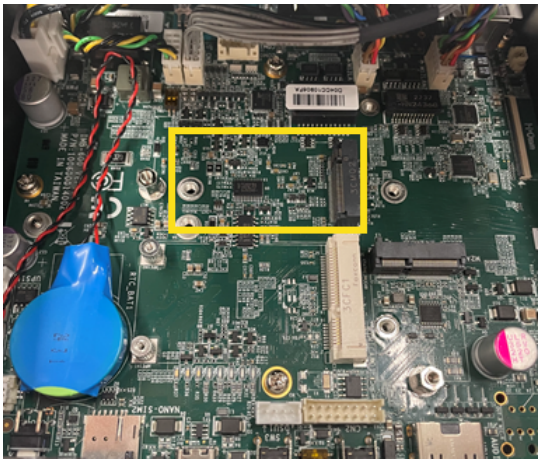
This section provides step-by-step instructions on how to install M.2 3042/3052, M.2 2230, and mPCIe expansion modules. You can also refer to Expansion *(on page 45)* for more details such as pin definitions about the internal slots or connectors used for expansion purpose.

You need to remove the bottom cover to install expansion modules. See Removing the Bottom Cover *(on page 23)* for the instructions.

2.2.2.1. Installing an M.2 3042 or 3052 Module

1. **Installing an M.2 3042 Key B LTE WWAN card (USB 3.0)**:

   a. Locate the M.2 3042 Key B slot on the mainboard.



   b. Move the standoff screw (H73D50) from location H8 to H9 as shown below:

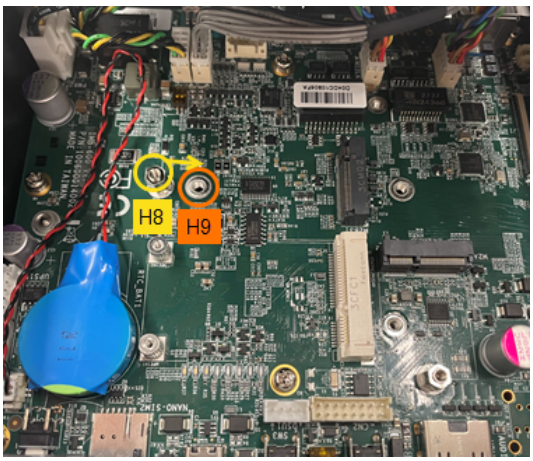


Figure 2-4 Standoff Screw (H73D50)

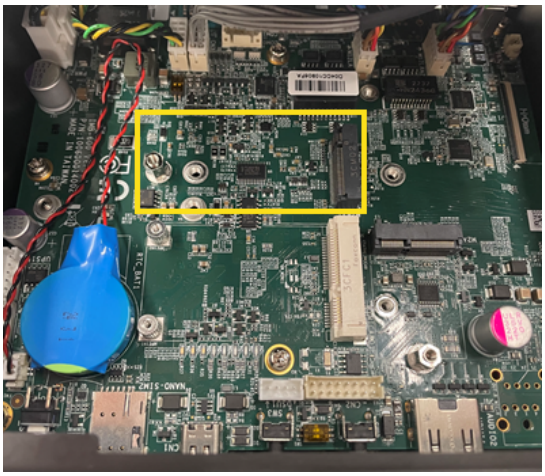
   c. Align the notch on the M.2 3042 WWAN card with the tab in the slot and gently insert the WWAN card at a 30 degree angle until it is fully embedded, and then press it down.

   d. Secure the expansion module to the mainboard with the M2.5x5L screw provided in the package (see Package Contents *(on page 12)*).

e. To install a heatsink for the installed M.2 3042 module, go to Installing a Heatsink for the M.2 3042 or 3052 Module *(on page 27)* for the instructions.

2. **Installing an M.2 3052 Key B 5G WWAN card (USB 3.0)**:

a. Locate the M.2 3052 Key B slot on the mainboard.



b. Align the notch on the M.2 3052 WWAN card with the tab in the slot and gently insert the WWAN card at a 30 degree angle until it is fully embedded, and then press it down.

c. To install a heatsink for the installed M.2 3052 module, go to Installing a Heatsink for the M.2 3042 or 3052 Module *(on page 27)* for the instructions.

2.2.2.1.1. Installing a Heatsink for the M.2 3042 or 3052 Module

1. Before you begin, check the Package Contents *(on page 12)* and get the items below:
   - 2 x standoff screw (H75D50)
   - 1 x M.2 WWAN heatsink Type 12
   - 1 x thermal pad (18x30x1.0T mm)
   - 1 x thermal pad (30x34x1.0T mm)
2. Attach the thermal pads to the heatsink as shown below.

a. Figure 2-5 Thermal pad (30x34x1.0T mm) attached to the heatsink

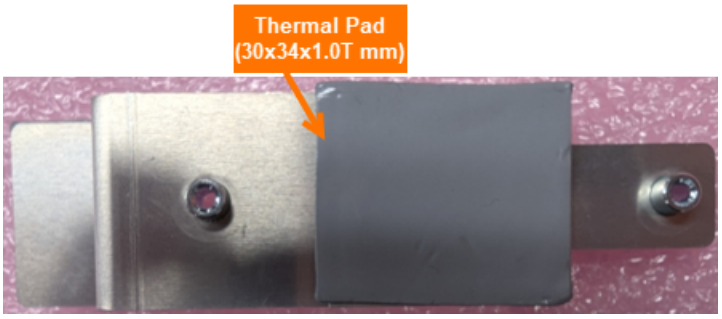
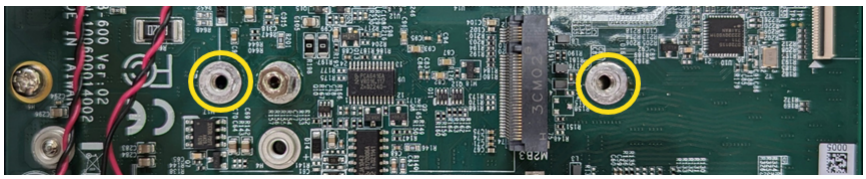


Figure 2-6 Thermal pad (18x30x1.0T mm) attached to the heatsink

3. Fasten the standoff screws (H75D50) to the mainboard as indicated in the figure below.



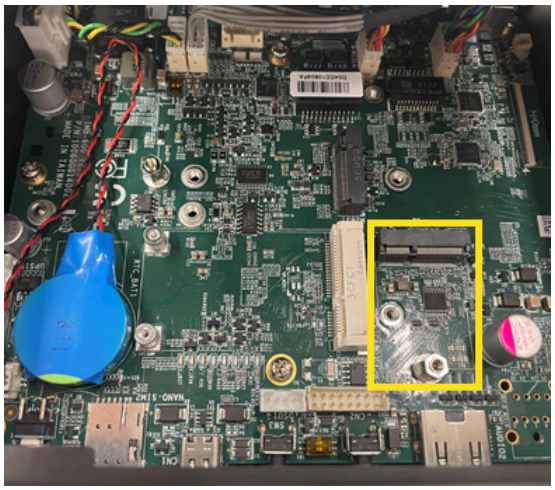
4. Align the two mounting holes on the heatsink with the mounting screw-holes on the mainboard, and then place the heatsink above the installed M.2 3042 or 3052 module.
5. Fasten the heatsink to the mainboard with the M2.5x10L screws.

Note:
One of the M2.5x10L screw is also used to secure the M.2 3052 module to the mainboard.

6. After completing the installation of the M.2 3042 LTE or 3052 5G module, go to Installing a Nano SIM Card *(on page 32)* for how to install a SIM card.

2.2.2.2. Installing an M.2 2230 Module

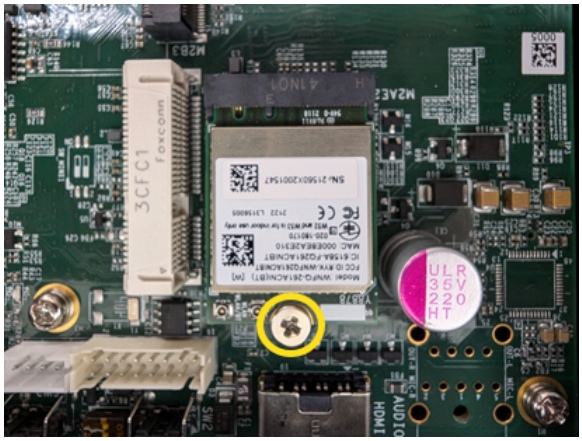
1. Locate the M.2 2230 Key E slot on the mainboard.



2. Align the notch on the expansion module with the ridge in the slot and gently insert the module at a 30 degree angle until it is fully embedded, and then press it down.

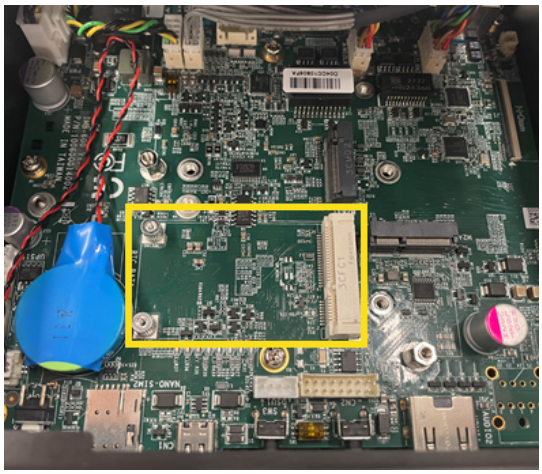
Note:
Ensure you align the semicircular mounting hole at the center of the opposite edge on the module with the copper pillar bump when pressing the module down.

3. Secure the expansion module to the mainboard with the M2.5x5L screw included in the package (see Package Contents *(on page 12)*).

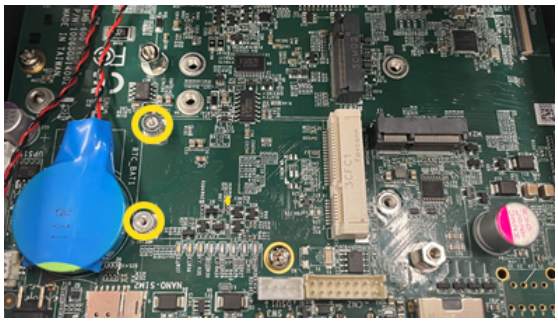


2.2.2.3. Installing an mPCIe Module

1. Locate the mPCIe connector on the mainboard.



2. Align the notch on the mPCIe card with the ridge in the slot and gently insert the card at a 30 degree angle until it is fully embedded, and then press it down.
3. Secure the mPCIe card to the mainboard with the M2x5L screw(s) included in the package (see Package Contents *(on page 12)*.



2.2.3. Installing a Backup Battery (BBU)

1. Before you begin, check the optional backup battery kit purchased from SINTRONES®, which includes:
    - 1 x battery cover

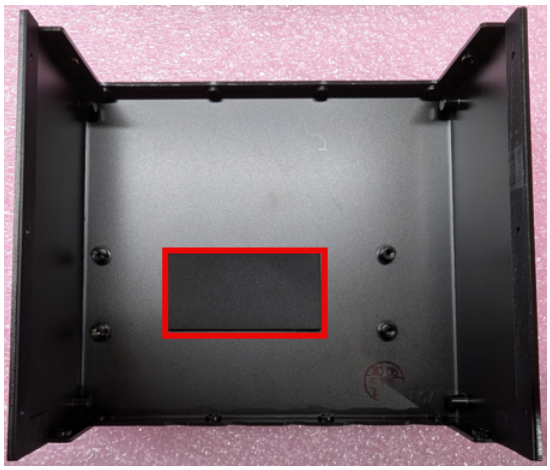
    - 1 x battery with a power cable

    - 4 x screw P3*6L

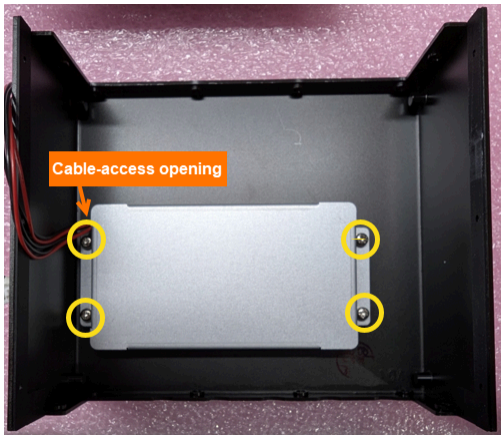
- 1 x sponge

2. On both sides of the battery, peel off the release liners of the double-sided tapes.
3. Turn over the battery and adhere it to the battery cover with the information label face up.

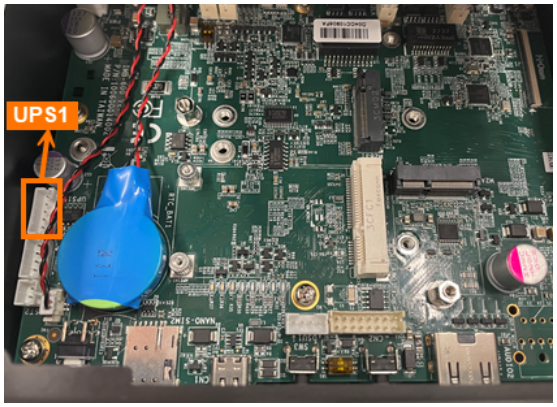
4. Remove the bottom cover of the system to install the backup battery. See Removing the Bottom Cover *(on page 23)* for the instructions.
5. On both sides of the sponge, peel off the release liners of the double-sided tape.
6. Attach the sponge to the area on the back of the bottom cover, as indicated below.



7. Turn over the battery cover and fasten it to the bottom cover with the P3*6L srews. Get the power cable out from the cable-access opening as shown below.

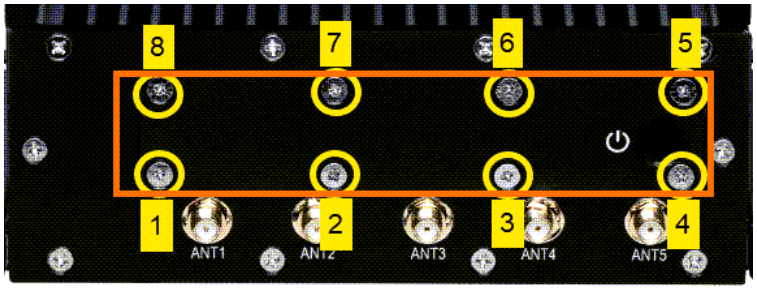


8. Connect the other end of the power cable to the UPS1 position on the mainboard as shown below.



2.2.4. Installing a Nano SIM Card

1. Ensure the IBOX-600-IP66 is powered off.
2. Remove the 8 screws in the order, as indicated in the following figure, to remove the protection cover from the rear panel.



3. Locate the **Nano SIM CARD** slot on the panel.

4. With the gold/bronze chip side facing up, insert the nano SIM card into the slot.
5. Use your fingernail or a small paperclip to press the card inwards until it locks in place.

Note:
To remove the SIM card, ensure you power off the system, and then use your fingernail or a small paperclip to push the card until it pops out.

6. Fasten the protection cover back onto the rear panel.

Important:

- In order to maintain the system's IP66 waterproof funcationality, ensure you fasten the screws by the specified order given in the figure in **Step 2** *(on page 32)*.

2.3. Booting the System

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

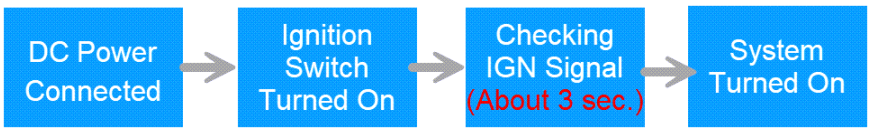
2.3.1. Turning On System By Ignition Switch

1. You can use the power cable that came with the package directly in fields without an ignition system.
2. Plug one end of the power cable into the DC-IN port of the computer and then plug the other end to a power source.
3. When the power is ready, turn on the ignition switch.
4. 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.

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

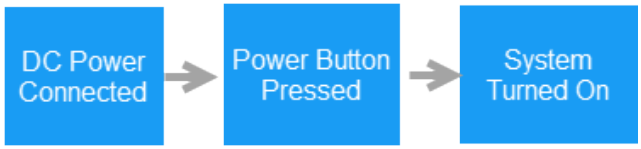
Figure 2-7 Turning On System By Ignition Switch



2.3.2. Turning On System By Power Button

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

Figure 2-8 Turning On System By Power Button



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-600-IP66

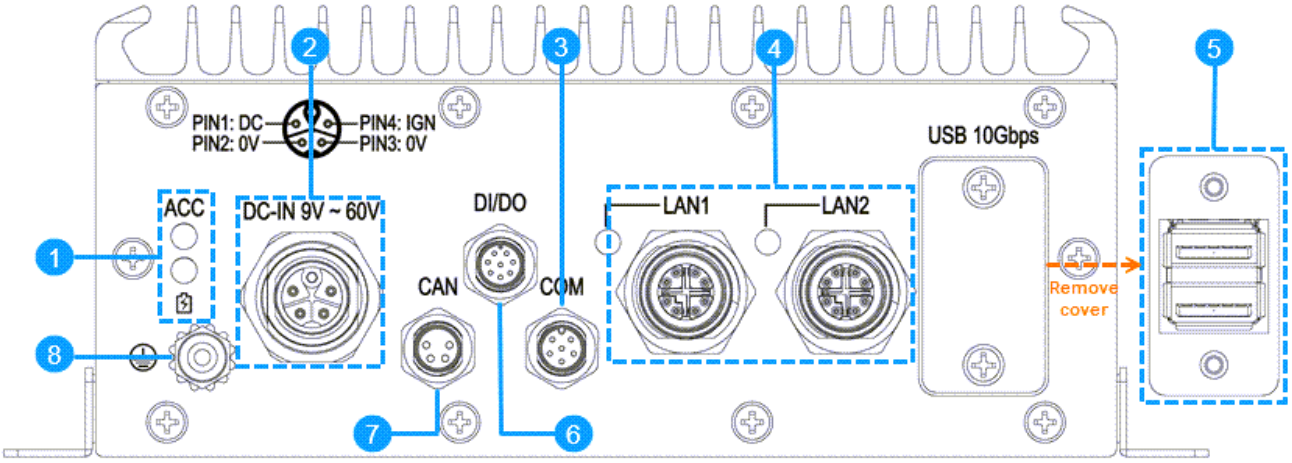


Table 3-1 I/O Interface on Front Panel

Item	I/O Interface	Description	Specification
1	LED Indicators	ACC: • ON: Ignition enabled • OFF: Ignition disabled	-
		(UPS): • ON: Internal backup battery enabled • OFF: Powered supplied from external power source or no backup battery installed	-
2	DC Input	M12 K-coded connector: • Input voltage range: 9 ~ 60 VDC	DC-IN Port (M12 K-coded Connector) *(on page 40)*
3	COM Port	• Supports RS-232/422/485 interfaces • Programmable via software configuration	COM Port (M8 A-coded Connector) *(on page 40)*
4	LAN Ports	M12 X-coded connectors: • LAN/ETH 1: 2.5 GbE (10/100/1000/2500BASE-T) • LAN/ETH 2: GbE (10/100/1000BASE-T)	LAN Port (M12 X-coded Connector) *(on page 42)*
5	USB (10Gbps) Ports	USB 3.2 Gen 2 (10 Gbps), 5V / 900mA	USB 3.2 Port *(on page 42)*

Item	I/O Interface	Description	Specification
		Note: Two USB Type-A ports share 10Gbps bandwidth.	
6	DI/DO Port	3 x DI (5 ~ 60 VDC), 4 x DO (12V / 100mA)	DI/DO Port (M8 A-coded Connector) *(on page 41)*
7	CAN FD Port	• Supports CAN FD protocol • Backward compatible with CAN bus 2.0	CAN FD Port (M8 A-coded Connector) *(on page 41)*
8	Grounding Terminal	M5 bolt with nut for connecting a grounding wire	-

3.2. Rear Panel

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

Figure 3-2 Rear Panel of IBOX-600-IP66

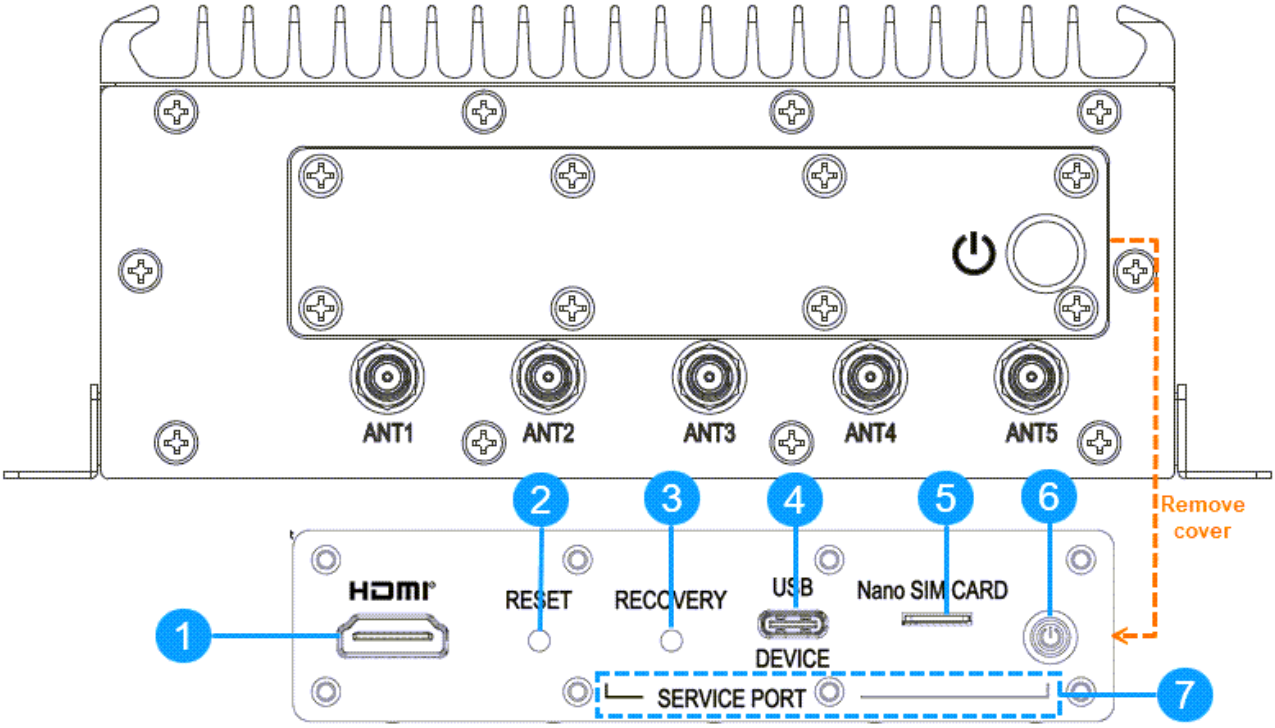


Table 3-2 I/O Interface on Rear Panel

Item	I/O Interface	Description	Specification
1	HDMI® Port	Supports resolutions up to 3840 x 2160p @60Hz	HDMI® Port *(on page 43)*
2	Reset Button	Used for system reboot	-
3	Recovery Button	Used for system recovery **Note:** The recovery button works only when the USB (Device) port (as described below) is connected to a host computer.	-
4	USB (Device) 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 65)* for the detailed instructions.	USB Type-C Port *(on page 43)*

Item	I/O Interface	Description	Specification
		Note: This USB port is used only for system recovery. It does not support power or other kinds of data transfer.	
5	Nano SIM Card Slot	Supports a nano SIM card **Note:** Ensure you power off the system before installing or removing the SIM card.	-
6	Power Button	System power status: • Red light: Standby mode • Blue light: System turned on	Power Button (LED Light Status) *(on page 44)*
7	Service Port	Indicates the recovery button and USB (Device) port	-

3.3. Specifications of External I/O Ports

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

3.3.1. DC-IN Port (M12 K-coded Connector)

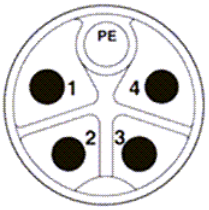


Table 3-3 Pin Definition of DC-IN Port (M12 K-coded Connector)

Pin	Signal
1	9 ~ 60 VDC
2	0V
3	0V
4	IGN

3.3.2. COM Port (M8 A-coded Connector)

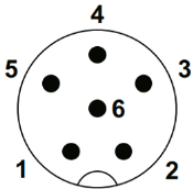


Table 3-4 Pin Definition of COM Port (M8 A-coded)

Pin	M8 A-coded		
	RS-232	RS-422	RS-485
1	NC	TXD-	Data-
2	RXD	TXD+	Data+
3	TXD	RXD+	NC
4	NC	RXD-	NC
5	GND	GND	GND
6	NC	NC	NC

3.3.3. CAN FD Port (M8 A-coded Connector)

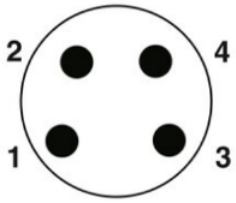


Table 3-5 Pin Definition of CAN Bus Port

Pin	M8 A-coded
	CAN Bus
1	CAN-L
2	GND
3	CAN-H
4	GND

3.3.4. DI/DO Port (M8 A-coded Connector)

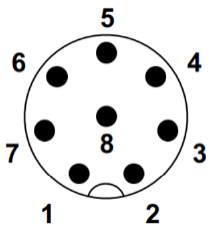


Table 3-6 Pin Definition of DI/DO Port

Pin	Signal	Pin	Signal
1	DO_2	2	DO_3
3	DO_4	4	DI_3
5	DI_2	6	DI_1
7	DO_1	8	GND

3.3.5. LAN Port (M12 X-coded Connector)

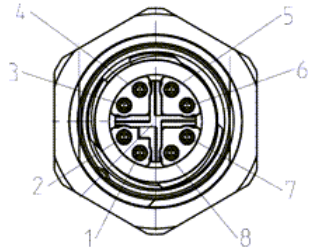


Table 3-7 Pin Definition of LAN Port (M12 X-coded Connector)

Pin	Signal
1	LAN_MDI0P
2	LAN_MDI0N
3	LAN_MDI1P
4	LAN_MDI1N
5	LAN_MDI3P
6	LAN_MDI3N
7	LAN_MDI2N
8	LAN_MDI2P

3.3.6. USB 3.2 Port

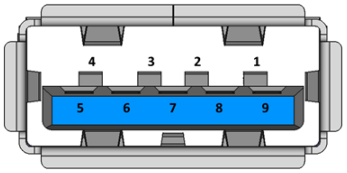


Table 3-8 Pin Definition of USB 3.2 Port

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+

3.3.7. HDMI® Port

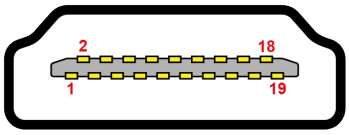


Table 3-9 Pin Definition of HDMI® Port

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		

3.3.8. USB Type-C Port

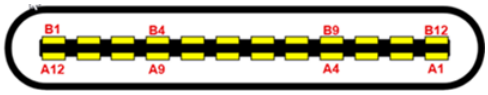


Table 3-10 Pin Definition of USB Type-C Port

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

3.3.9. Power Button (LED Light Status)

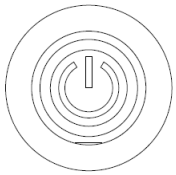


Table 3-11 Pin Definition of Power Button (LED Light Status)

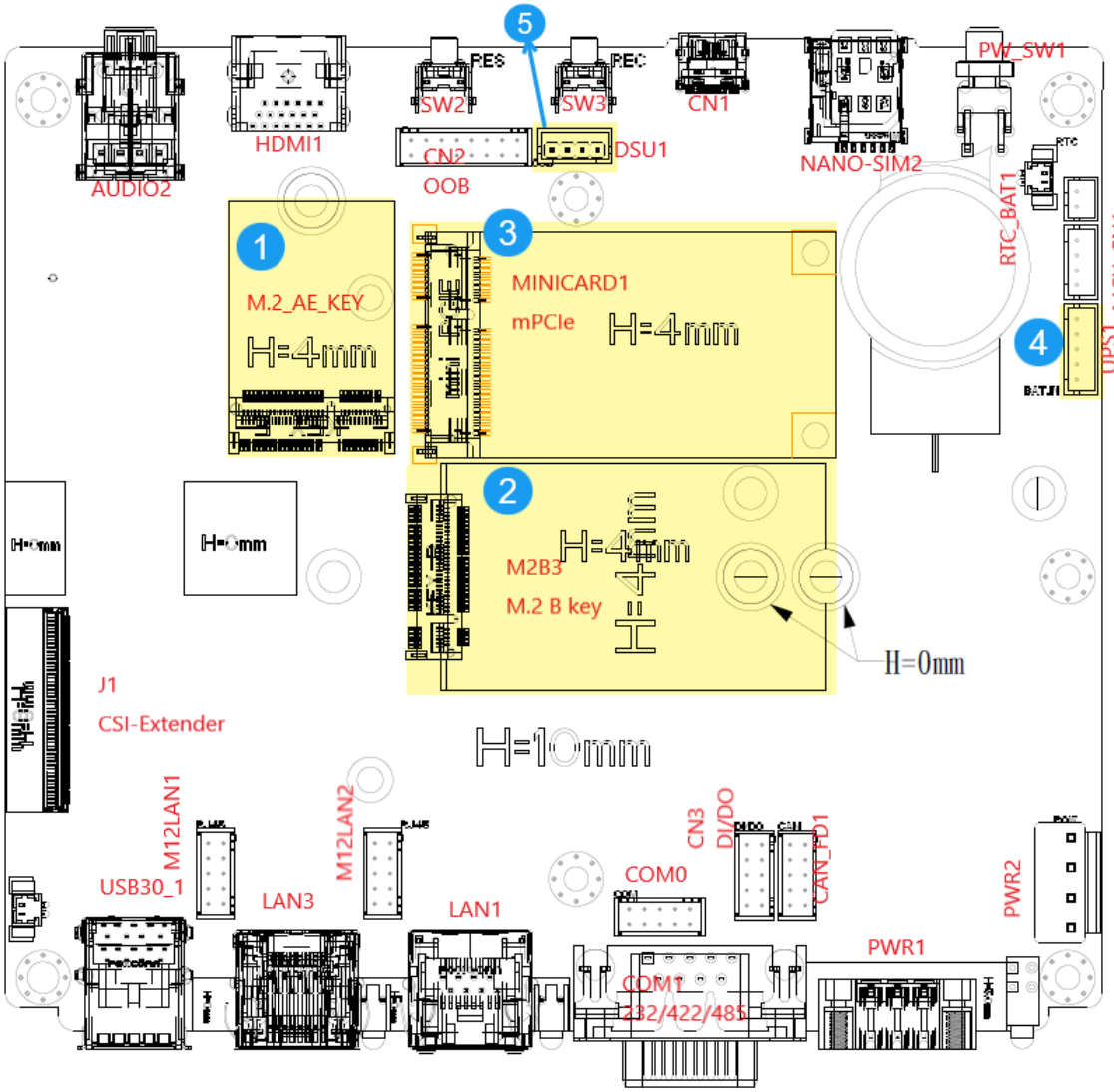
Pin	Signal	Pin	Signal
LED			
A1	POWER_LED+	C1	POWER_LED-

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



Item	Internal Connector	Description	Specification
1	M.2_AE_KEY	M.2 2230 Key E slot used for installing an expansion module such as a WLAN or Bluetooth module	M.2 Key E Slot *(on page 47)*
2	M2B3	M.2 3042/3052 Key B slot used for installing an expansion module such as a WWAN module	M.2 Key B Slot *(on page 48)*
3	MINICARD1	mPCIe full-size slots used for installing up to two mini PCIe cards supporting USB 2.0 interface	mPCIe Connector *(on page 50)*
4	UPS1	Used for installing the backup battery	UPS1 (BBU) JST Connector *(on page 51)*
5	DSU1	Used for monitoring system status and collecting debug logs	DSU JST Connector *(on page 52)*

4.2. Specifications of Expansion Slots/Connectors

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

4.2.1. M.2 Key E Slot

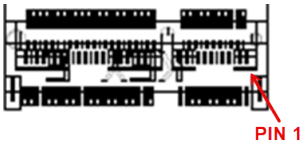
M.2 Key E Slot	Description
Size	NGFF 2230 / 75 Pin
Type	M.2 Key E H:8.5mm
Location	See Top View of Mainboard *(on page 46)* for the information.
Drawing	

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

Pin	Signal	Pin	Signal
1	GND	2	V3P3_A
3	HUBA_USB_2P	4	V3P3_A
5	HUBA_USB_2N	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	NC
19	NC	20	BT_UART_WAKE_B
21	NC	22	UART0_RXD
23	NC	24	KEY
25	KEY	26	KEY
27	KEY	28	KEY
29	KEY	30	KEY
31	KEY	32	UART0_TXD
33	GND	34	UART0_CTS
35	PCIE1_TX0_P	36	UART0_RTS

Pin	Signal	Pin	Signal
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	BT_OSC_32KHZ
51	GND	52	PCIE1_RST
53	PCIE1_CLKREQ	54	M2E_WIFI_DIS2(BT)
55	PCIE_WAKE	56	M2E_WIFI_DIS1(WIFI)
57	GND	58	JS_I2C2_SDA
59	NC	60	JS_I2C2_SCL
61	NC	62	M2E_ALERT
63	GND	64	NC
65	NC	66	NC
67	NC	68	NC
69	GND	70	WIFI_WAKE
71	NC	72	V3P3_A
73	NC	74	V3P3_A
75	GND		

4.2.2. M.2 Key B Slot

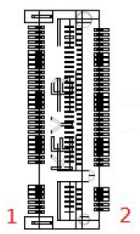
M.2 Key B Slot	Description
Size	NGFF 3042 / 3052 / 75 Pin
Type	M.2 Key B H:8.5mm
Location	See Top View of Mainboard *(on page 46)* for the information.
Drawing	

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

Pin	Signal	Pin	Signal
1	NC	2	3VSB
3	GND	4	3VSB
5	GND	6	Reserve
7	M2B_USB1_DP	8	Reserve
9	M2B_USB1_DN	10	Reserve
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	24	NC
25	M2B1DPR	26	Reserve
27	GND	28	NC
29	M2B_USB3.2_RXN	30	M2UIM_RST_A
31	M2B_USB3.2_RXP	32	M2UIM_CLK_A
33	GND	34	M2UIM_DAT_A
35	M2B_USB3.2_TXN	36	M2UIM_PWR_A
37	M2B_USB3.2_TXP	38	NC
39	GND	40	NC
41	NC	42	NC
43	NC	44	M2B_ALERN
45	GND	46	NC
47	NC	48	NC
49	NC	50	DEV_3V3RST#
51	GND	52	Reserve
53	NC	54	WWAN_WAKE
55	NC	56	NC
57	GND	58	NC
59	NC	60	NC
61	NC	62	NC
63	NC	64	NC
65	NC	66	NC

Pin	Signal	Pin	Signal
67	M2B1RST2	68	NC
69	CONFIG_1	70	3VSB
71	GND	72	3VSB
73	GND	74	3VSB
75	NC		

4.2.3. mPCIe Connector

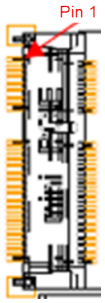
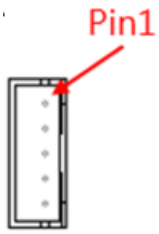
mPCIe Connector	Description
Size	2 x 26 / 52 Pin
Type	MINI PCI-E CON H:9.2mm
Location	See Top View of Mainboard *(on page 46)* for the information.
Drawing	

Table 4-3 Pin Definition of mPCIe Connector (MINICARD1 & 2)

Pin	Signal	Pin	Signal
1	WAKE#	2	3V3_VSB
3	NC	4	GND
5	NC	6	1V5(Reserve)
7	NC	8	NC
9	GND	10	NC
11	NC	12	NC
13	NC	14	NC
15	GND	16	NC
17	NC	18	GND
19	NC	20	W_DIS
21	GND	22	RESET#
23	NC	24	3V3_VSB

Pin	Signal	Pin	Signal
25	NC	26	GND
27	GND	28	1V5(Reserve)
29	GND	30	I2C_SCL
31	NC	32	I2C_SDA
33	NC	34	GND
35	GND	36	USB_DN
37	GND	38	USB_DP
39	3V3_VSB	40	GND
41	3V3_VSB	42	NC
43	GND	44	NC
45	NC	46	NC
47	NC	48	1V5(Reserve)
49	NC	50	GND
51	NC	52	3V3_VSB

4.2.4. UPS1 (BBU) JST Connector



See Top View of Mainboard *(on page 46)* for the location of the UPS1 JST connector.

Table 4-4 Pin Definition

Pin	Signal
1	+12V UPS
2	+12V UPS
3	NC
4	GND
5	GND

4.2.5. 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 46)* for the information.
Drawing	

Table 4-5 Pin Definition

Pin	Signal
1	Power
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

5. Software

Topics in this chapter include:

5.1. System Configuration

This section summarizes commands available for configuring smart power management, COM (RS-232/422/485), CAN FD, and DI/DO interface controllers.

5.1.1. Smart Power Management Settings

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

5.1.1.1. 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.

5.1.1.2. Smart Power Management Commands

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

Table 5-1 Checking IGN/UPS Status

Command	Value	Description	Outcome
`i2cget -y -f 1 0x4a 0x12`	-	Check the ignition (IGN) status	• `0x00`: Ignition turned off • `0x01`: Ignition turned on
`i2cget -y -f 1 0x4a 0x10`	-	Check the UPS backup battery status	• `0x07`: UPS mode

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 56)*, 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`

Command	Value	Description	Outcome
`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 56)*, the system will wait for 2 minutes to power on after the ignition is turned on.
`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 56)*, 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 56)*, 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 56)*, the system will wait for 2 minutes to power off after the ignition is turned off.

Command	Value	Description	Outcome
`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 56)*, the system will wait for 1 hour to power off after the ignition is turned off.

Table 5-4 Saving the Setting

Command	Description	Outcome
`i2cset -f -y 1 0x4a 0x28 0x02`	Save the setting.	The specified setting will be saved. **Important:** Ensure you enter `i2cset -f -y 1 0x4a 0x28 0x02` to save and allow the system to excute the specified setting.

5.1.2. Commands for COM Port (RS-232/422/485 Configuration)

The name of the COM port: ttyTHS1

See the following tables as the available commands list for the COM port.

Table 5-5 Switching between RS-232/422/485

Command	Outcome	Description
`sudo rs232_set`	RS232 Mode Set	Enable the RS-232 mode
`sudo rs422_set`	RS422 Mode Set	Enable the RS-422 mode
`sudo rs485_set`	RS485 Mode Set	Enable the RS-485 mode

Table 5-6 Checking the Status

Command	Outcome	Description
`i2cget -f -y 1 0x4a 0x3c`	• 0x09 rs232 • 0x1b rs422 • 0x12 rs485	Check which mode (RS-232/422/485) is enabled.

5.1.2.1. RS-232 Test Program

This section provides some sample programs for sending commands over RS-232.

Enter the following command to complie the RS-232 program:

```
test@test-desktop:~/uarta$ gcc rs232-demo.c -o rs232-demo
```

A Test Program for RS-232:

```
test@test-desktop:~/uarta$ sudo ./rs232-demo
/dev/i2c-1: device 0x4a at address 0x3c: 0x08
writing 'ABCDEF' to THS1
writing 'ABCDEF' to THS1
writing 'ABCDEF' to THS1
writing 'ABCDEF' to THS1
writing 'ABCDEF' to THS1
123567890
123567890
123567890
123567890
123567890
closing THS0
test@test-desktop:~/uarta$
```

5.1.2.2. RS-422 Test Program

This section provides some sample programs for sending commands over RS-422.

Enter the following command to complie the RS-422 program:

```
test@test-desktop:~/uarta$ gcc rs422-demo.c -o rs422-demo
```

A Test Program for RS-422:

```
test@test-desktop:~/uarta$ sudo ./rs422-demo
/dev/i2c-1: device 0x4a at address 0x3c: 0x18
writing 'ABCDEF' to THS1
writing 'ABCDEF' to THS1
writing 'ABCDEF' to THS1
writing 'ABCDEF' to THS1
writing 'ABCDEF' to THS1
123567890
123567890
123567890
123567890
123567890
closing THS0
test@test-desktop:~/uarta$
```

5.1.2.3. RS-485 Test Program and Source Code

This section provides some sample programs for sending commands over RS-485.

Enter the following command to complie the RS-485 program:

```
test@test-desktop:~/uarta$ gcc rs485-demo.c -o rs485-demo
```

A Test Program for RS-485:

```
test@test-desktop:~/uarta$ sudo ./rs485-demo
/dev/i2c-1: device 0x4a at address 0x3c: 0x10
writing 'ABCDEF' to THS1
writing 'ABCDEF' to THS1
writing 'ABCDEF' to THS1
writing 'ABCDEF' to THS1
writing 'ABCDEF' to THS1
123567890
123567890
123567890
123567890
123567890
closing THS0
test@test-desktop:~/uarta$
```

The source code:

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

#include <linux/types.h>
#include <linux/i2c.h>
#include <linux/i2c-dev.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

#define SYSFS_GPIO_EXPORT           "/sys/class/gpio/export"
#define SYSFS_GPIO_RST_PIN_VAL      "PX.06"
#define SYSFS_GPIO_RST_DIR          "/sys/class/gpio/PX.06/direction"
#define SYSFS_GPIO_RST_DIR_VAL      "out"
#define SYSFS_GPIO_RST_VAL          "/sys/class/gpio/PX.06/value"
#define SYSFS_GPIO_RST_VAL_H        "1"
#define SYSFS_GPIO_RST_VAL_L        "0"

int sp0 = 0;
int escPress = 0;
int fd;
static struct termios oldChars, newChars;

void set_term_quiet_input()
{
  //struct termios tc;
  tcgetattr(0, &oldChars);
  newChars = oldChars;
  newChars.c_lflag &= ~(ICANON | ECHO);
  newChars.c_cc[VMIN] = 0;
  newChars.c_cc[VTIME] = 0;
  tcsetattr(0, TCSANOW, &newChars);
}

/* Restore old terminal i/o settings */
void resetTermios(void)//struct termios &oldChars)
{
  tcsetattr(0, TCSANOW, &oldChars);
}

void * getch(void *arg)
{
    struct pollfd pfd = { .fd = 0, .events = POLLIN };
    set_term_quiet_input();
    while (!escPress)
    {
        if (poll(&pfd, 1, 0)>0)
        {
            int ch = getchar();
            printf("Key pressed: %d \n", ch);
            if (ch==27)
            {
                escPress = 1;
                //break;
            }
        }
    usleep(10); // Some work
    }
    resetTermios();
}

void * readData(void *arg)
{
    while(!escPress) {
        char readbuf[64] = {0};
        int n = read(sp0, &readbuf, sizeof(readbuf));
        if (n < 0) {
          printf("error reading from THS0");
        } else {
          printf("uarta reading: %s\n", readbuf);
        }
        usleep(100 * 1000);
    }
}

int main(void) {
    struct serial_rs485 rs485conf = {0};
    char buf[8] = {0};
    int n = 0;
    //int sp0 = 0;
    //int fd;
    struct termios tio;

    uint8_t data, addr = 0x4a, reg = 0x3C;
    //const char *path = argv[1];
    const char *path = "/dev/i2c-1";
```

```
    int file, rc;
    char i2cbuf[3] = {0};

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

    rc = ioctl(file, I2C_SLAVE, addr);
    if (rc < 0)
 err(errno, "Tried to set device address '0x%02x'", addr);

    i2cbuf[0] = reg;
    i2cbuf[1] = 0x12;
    if (write(file, i2cbuf, 2) != 2) {
        perror("Failed to write to the I2C device");
        exit(1);
    }
    /* Using I2C Read, equivalent of i2c_smbus_read_byte(file) */
    if (read(file, i2cbuf, 1) != 1) {
        perror("Failed to read to the I2C device");
        exit(1);
    } else {
        printf("Please enter:\n");
        //printf("%s: device 0x%02x at address 0x%02x: 0x%02x\n", path, addr, reg, i2cbuf[1]);
   }

    memset(&tio, 0, sizeof(tio));

    tio.c_iflag = 0;
    tio.c_oflag = 0;
    tio.c_cflag = CS8 | CREAD | CLOCAL;
    tio.c_lflag = 0;
    tio.c_cc[VMIN] = 1;
    tio.c_cc[VTIME] = 5;
    tio.c_cflag |= CRTSCTS;            // use RTS/CTS

    cfsetospeed(&tio, B115200);

    sp0 = open ("/dev/ttyTHS1", O_RDWR);
    if (sp0 < 0) {
       printf("Error: Can't open: /dev/ttyTHS1\n");
       goto e_close;
    }
#if 1

    /* Read the current state of the RS-485 options with ioctl. */
    if (ioctl (sp0, TIOCGRS485, &rs485conf) < 0) {
            printf("Error: TIOCGRS485 ioctl not supported.\n");
    }

    int rtsflag;

    /*
     * 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.
     */
    rtsflag = TIOCM_RTS;
    int ret = ioctl(sp0, TIOCMBIC, &rtsflag);
    if (ret < 0) {
        printf("Can not set rts pint, octl failed %d, (errno: %d)\n", ret, errno);
        return -1;
    }


     /* 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 (sp0, TIOCSRS485, &rs485conf) < 0) {
            //printf("Error: TIOCSRS485 ioctl not supported.\n");
    }

    /* set the termios so the rs485 settings are applied */
    tcsetattr(sp0, TCSANOW, &tio);
    fcntl(sp0, F_SETFL, 0);
#endif
    fd = open(SYSFS_GPIO_EXPORT, O_WRONLY);
    if(fd == -1)
    {
          printf("ERR: Radio hard reset pin open error.\n");
          close(fd);
          return -1;
    }
    write(fd, SYSFS_GPIO_RST_PIN_VAL ,sizeof(SYSFS_GPIO_RST_PIN_VAL));
    close(fd);

    fd = open(SYSFS_GPIO_RST_DIR, O_WRONLY);
    if(fd == -1)
    {
         printf("ERR: Radio hard reset pin direction open error.\n");
         close(fd);
         return -1;
    }
    write(fd, SYSFS_GPIO_RST_DIR_VAL, sizeof(SYSFS_GPIO_RST_DIR_VAL));
    close(fd);

     fd = open(SYSFS_GPIO_RST_VAL, O_RDWR);

    pthread_t t2;
    if (pthread_create(&t2, NULL, readData, NULL) != 0) {
        printf("Error: pthread_create2\n");
    }


    while(!escPress) {

        char wstr[128] = {0};
        gets (wstr);
         if(wstr[0] == 'q' && strlen(wstr) == 1 )
        {
            escPress = 1;
            usleep(1000 * 1000);
            break;
        }
        //puts (wstr);
        write(fd, SYSFS_GPIO_RST_VAL_H, sizeof(SYSFS_GPIO_RST_VAL_H));
        write(sp0, &wstr, strlen(wstr));
        usleep(100 * 1000);
        write(fd, SYSFS_GPIO_RST_VAL_L, sizeof(SYSFS_GPIO_RST_VAL_L));

    }

    close(fd);
    printf("closing THS1\n");
    if (close (sp0) < 0) {
            printf("Error: Can't close: /dev/ttyTHS1\n");
    }

e_close:
    if(sp0) {
        close(sp0);
    }

    return 0;
}
```

5.1.3. 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.

Command	Description	Outcome
`sudo can_set`	Enable the CAN bus mode	CAN Bus set completed
`candump can0 &`	Receive data	N/A
`cansend can0 123#abcdabcd`	Send data	N/A
`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

5.1.4. DIO Configuration

IBOX-600-IP66supports 3 programmable digital inputs (DI) and 4 digital outputs (DO), which can be configured to communicate or exchange data with the 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 certain DI or DO channel.

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

See the following as the DO data register table:

Table 5-7 DO Data Register – 0x31

Bit	Pin	Value
3	DO_4	• Low: 0 • High: 1
2	DO_3	
1	DO_2	
0	DO_1	

5.1.4.1. DIO Commands

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

Table 5-8 Checking Digital Input (DI) Status

Command	Value	Description
`i2cget -f -y 1 0x4A 0x30`	N/A	Get the state of all the digital input pins

Table 5-9 Examples of Configuring Digital Outputs (DO)

Command	Value	Description
`i2cset -f -y 1 0x4A 0x31 <value>`	`0xf`	Set DO1 ~ DO4 as High (1111)
	`0x0`	Set DO1 ~ DO4 as Low (0000)
	`0x1`	Set DO1 as High (0001: DO1 = 1)

Command	Value	Description
	`0x2`	Set DO2 as High (0010: DO2 = 1)
	`0x4`	Set DO3 as High (0100: DO3 = 1)
	`0x8`	Set DO4 as High (1000: DO4 = 1)

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: 4GB or above
- Storage space: 256GB or above
- Recommended OS: Ubuntu 20.04

Accessories:

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

5.2.2. Configuring a 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-get update
$ sudo apt-get install sshpass
$ sudo apt-get install abootimg
$ sudo apt-get install nfs-kernel-server
$ sudo apt-get install libxml2-utils
$ sudo apt-get install binutils
```

5.2.3. Downloading a BSP Image

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.

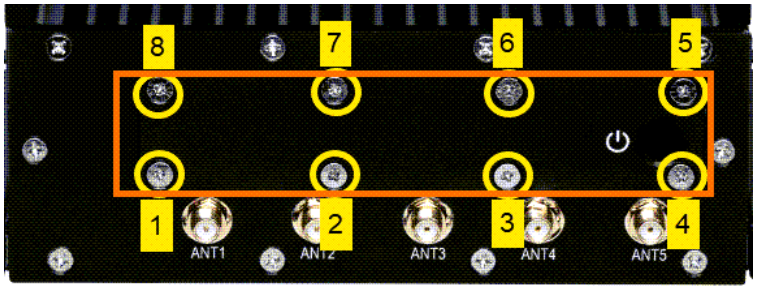
1. Download the BSP image from SINTRONES® website or contact our sales representative for the BSP image.
2. Save the downloaded BSP image in the host computer.

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. Remove the 8 screws in the order, as indicated in the figure below, to remove the protection cover on the rear panel.



3. Locate the **USB (DEVICE)** Type-C port on the rear panel.
4. 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.

5. Locate the **RECOVERY** button on the rear panel.

6. 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.

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

5.2.5. Executing System 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 65)*.

   Figure 5-1 An Example of Strings with the VID/PID Info

   ```
   Bus 002 Device 005: ID 0955:7423 NVIDIA Corp. APX
   ```

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

VID/PID by built-in NVIDIA® Jetson modules:

- Orin™ Nano 4GB: `ID 0955:7623`
- Orin™ Nano 8GB: `ID 0955:7523`
- Orin™ NX 8GB: `ID 0955:7423`
- Orin™ NX 16GB: `ID 0955:7323`

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

3. Open the downloaded BSP image on the host computer.
4. Enter the following command to decompress the BSP image.

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

Note:
Replace the texts within the brackets with the actual BSP file name. For example, if the BSP image file name is `IBOX-600-5.10.192-5.1.3-ubuntu20.04-R2.00-00`, enter

```
$ sudo tar -jxvf IBOX-600-5.10.192-5.1.3-ubuntu20.04-R2.00-00.tbz2
```

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

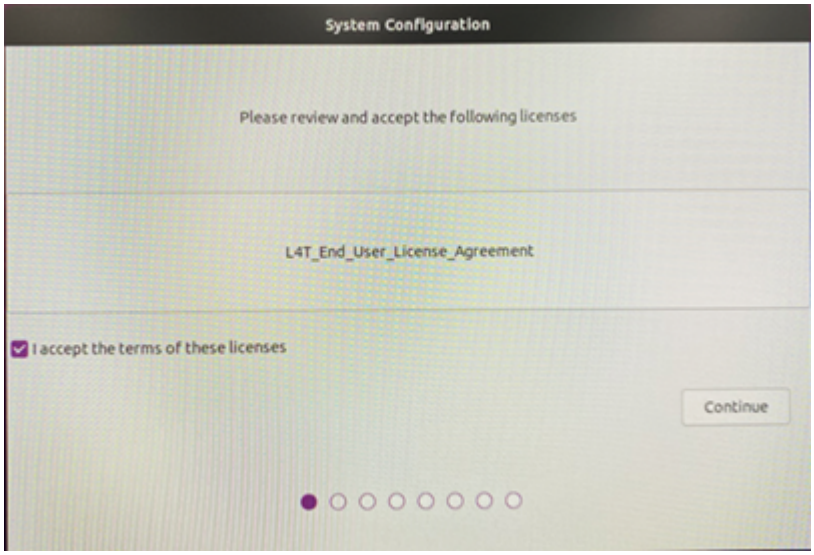
5. After the BSP image is decompressed, the folder `Linux_for_Tegra` will be automatically generated. Enter the following command to navigate to the folder.
   `$ cd Linux_for_Tegra`
6. Enter one of the following mode-specific commands to flash the BSP image to the system.
   - Original Mode: `$ sudo ./ibox600_na_flash.sh`
   - Super Mode: `$ sudo ./ibox600_nx_flash.sh`
7. Connect the system to a monitor via an HDMI cable for later use.
8. 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.

Important:
In order to maintain the system's IP66 waterproof funcationality, ensure you fasten the screws by the specified order given in the figure in **Setting the System in Recovery Mode: Step 2** *(on page 66)* when fastening the protection cover back onto the rear panel.

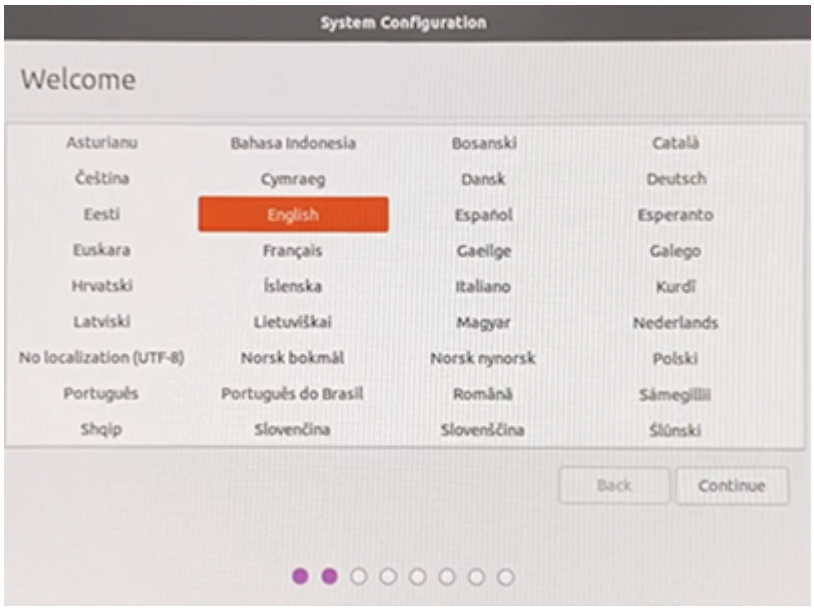
5.2.6. Configuring the Recovered System

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

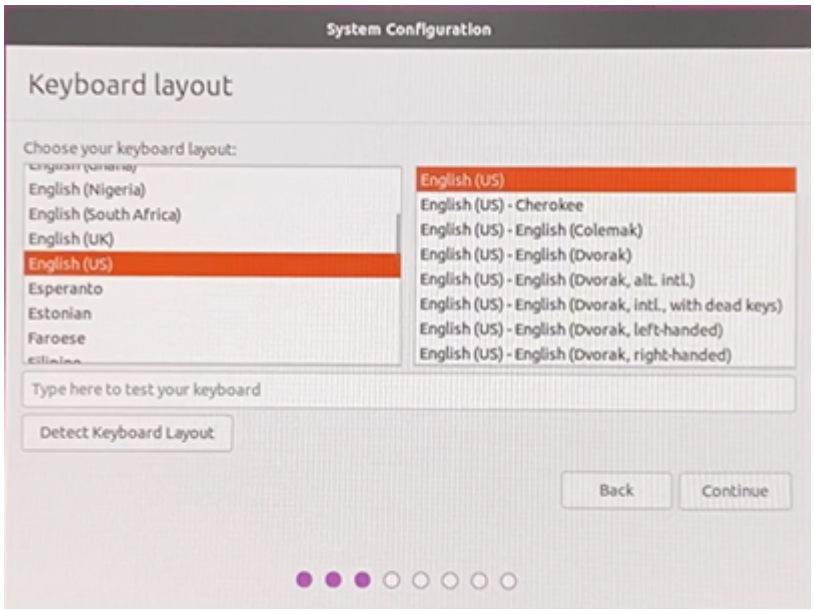
1. After the system reboot, the **System Configuration** wizard appears to guide users to complete the initial basic settings for the Linux for Tegra (LT4) platform.
2. Select **I accept the terms of these licenses** and then select **Continue**.



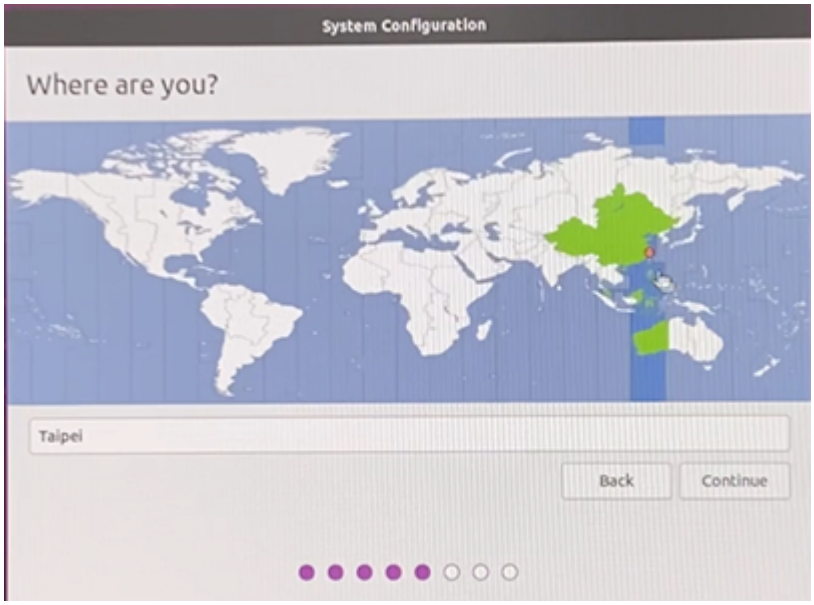
3. Select the preferred language setting and then select **Continue**.



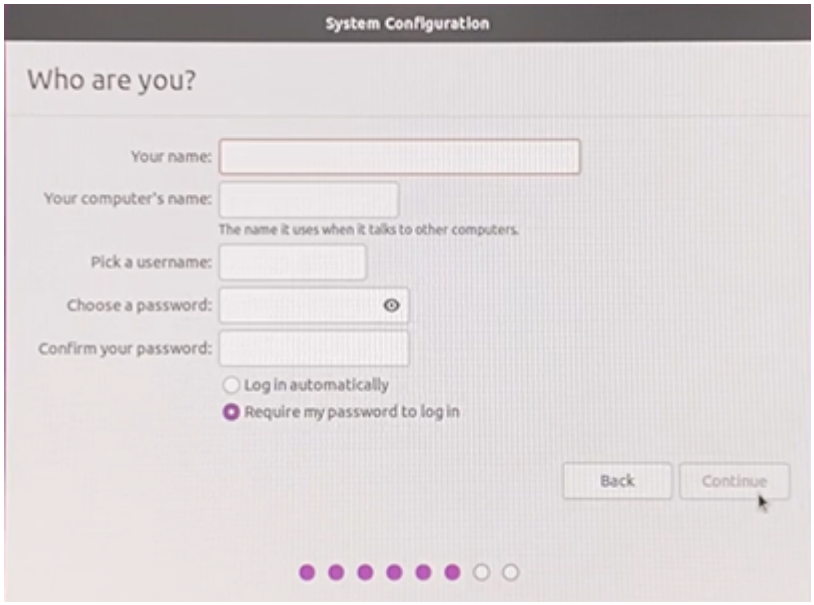
4. Select the preferred keyboard layout and then select **Continue**.



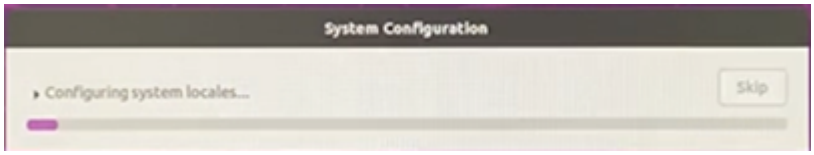
5. Select your location and then select **Continue**.

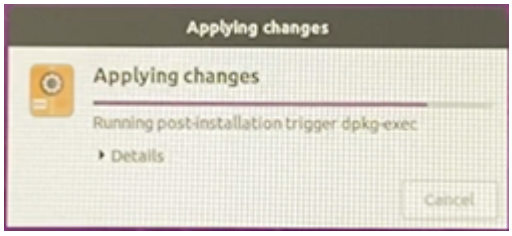


6. Specify the credentials such as a username and a password to create a user account. It is suggested to select **Require my password to log in** for security reasons. Select **Continue** to proceed to next step.



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