ARMxy Raspberry Pi CM5 Embedded Computer

BLIIOT®

MAKE IIOT EASIER

BL460
User Manual

Version: V1.0

Date: 2026-1-26

Shenzhen Beilai Technology Co.,Ltd

Website: https://www.bliiot.com

Preface

Thanks for choosing BLIIOT ARM based Embedded Computer. These operating instructions contain all the information you need for operation of BL460.

Copyright

Disclaimer

This document is designed for assisting user to better understand the device. As the described device is under continuous improvement, this manual may be updated or revised from time to time without prior notice. Please follow the instructions in the manual. Any damages caused by wrong operation will be beyond warranty.

This product can only be used with the system version provided by our company. We shall not be held liable for any issues such as software installation failures, usage problems, incompatibility, property loss, or personal injury caused by the use of other systems.

Revision History

Revision Date	Version	Description	Owner
2026/1/26	V1.0	Initial Release	ZJP

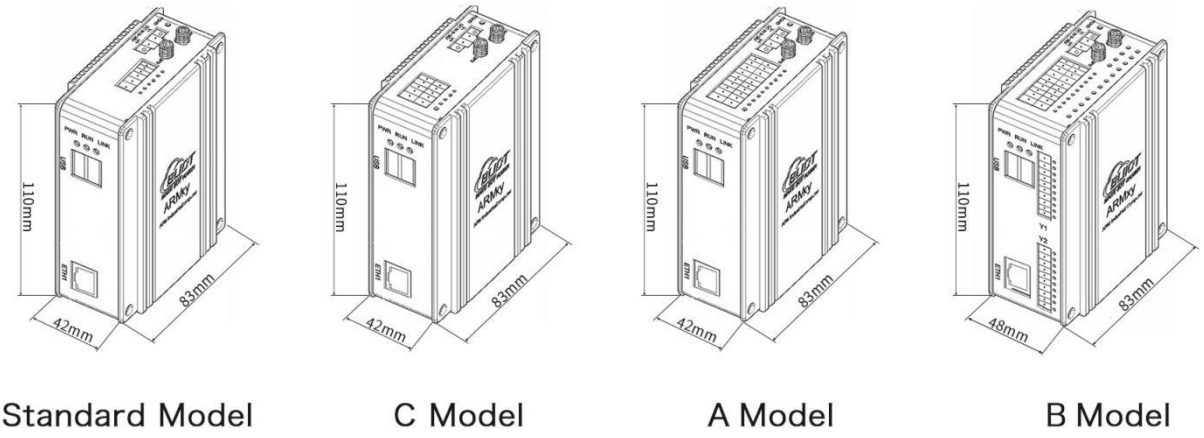
Table of Contents

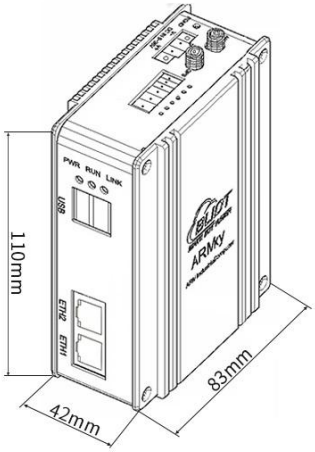
1 Introduction

1.1 Overview

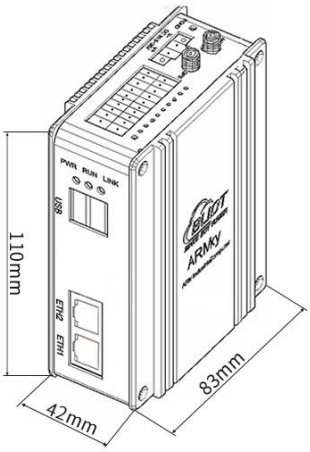
BL460 is an industrial embedded computer based on the Raspberry Pi CM5 module, featuring a quad-core Cortex-A76 processor that delivers high performance with low power consumption. It is fully compatible with the Raspberry Pi software ecosystem and supports flexible industrial I/O expansion. With an onboard M.2 interface for SSD storage or AI accelerator modules, BL460 is well suited for edge computing, industrial control, AIoT gateways, and intelligent device applications. Its industrial-grade design ensures reliable operation in harsh environments.

1.2 Appearance

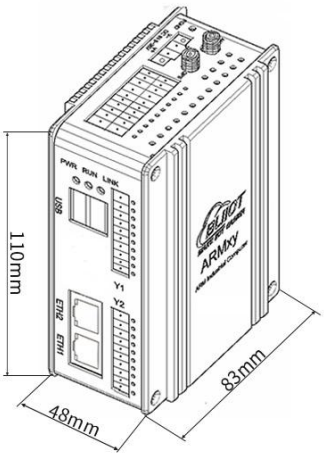




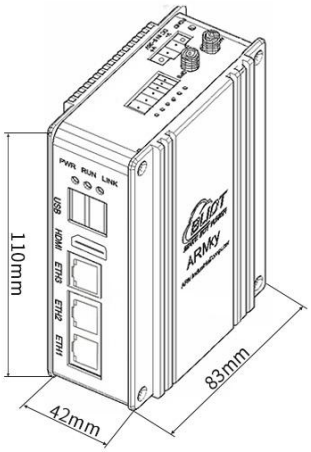
Standard Model



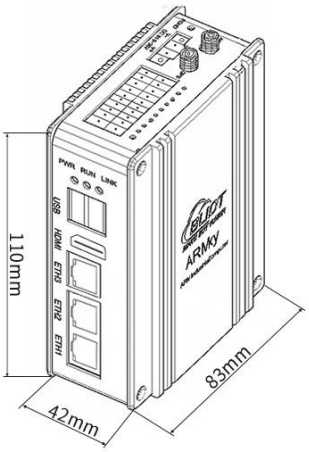
A Model



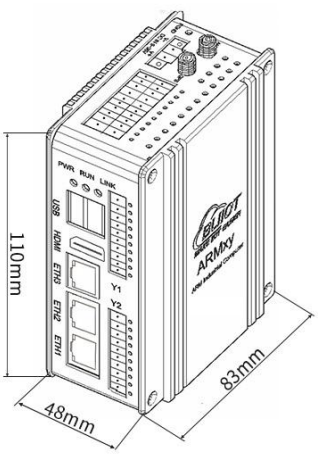
B Model



Standard Model



A Model



B Model

1.3 Technical Specifications

	Parameter	Description
System	CPU	Broadcom BCM2712
	Clock Speed	Quad-core Arm Cortex-A76 processor Clock speed: up to 2.4 GHz, with cryptographic extensions, 512KB L2 cache per core, and 2MB shared

		L3 cache
	GPU	GPU: VideoCore VII GPU, supporting OpenGL ES 3.1, Vulkan 1.2, and 4Kp60 HEVC decoding
	RAM	2/4/8/16GByte LPDDR4X
	Storage	8/16/32/64GByte eMMC
Power	Input Voltage	DC 12～24V
	Consumption	Normal: 318mA@12V(with 4G module), 285mA@12V (without 4G module) Maximum: 700mA@12V
	Reverse Polarity	Reverse Polarity Protection
Ethernet	Specification	RJ-45 ports, 1 to 3 available: 1 ports support 10/100/1000Mbps 2 port supports 10/100Mbps adaptive speed
	Protection	ESD ±4kV (contact), ±8kV (air);
SIM Card	Slot Quantity	1
	Type	Drawer interface
USB	Quantity	1*micro USB, 2*USB 3.0 HOST
SD Card	Quantity	1
	Type	Support SD, SDHC and SDXC(UHS-I) card
HDMI	Quantity	1
M.2 Interface	Quantity	1
Antenna	Interface	1xWi-Fi/4G, 1xGPS antenna
	Type	SMA
Serial Port (optional)	Channels	2/4/8 channels x RS232/RS485
	Baud Rate	300bps–115200bps
	Data Bits	7, 8
	Parity	None, Even, Odd
	Stop Bits	1, 1.5, 2
X board digital input (optional)	Channels	2/4/8/16 channels
	Input Type	Supports dry contact or wet contact
	Dry Contact	Close = short circuit Open = open circuit
	Wet Contact	Logic 0 = 0–3V DC Logic 1 = 10–30V DC
	Isolation protection	2KVrms
X board digital output (optional)	Channels	2/4/8/16 channels
	Output Type	SINK
	Capacity	100mA per channel
4G	L-E	GSM/EDGE:900,1800MHz

Module(Optional)		WCDMA:B1,B5,B8 FDD-LTE:B1,B3,B5,B7,B8,B20 TDD-LTE:B38,B40,B41
	L-CE	GSM/EDGE:900,1800MHz WCDMA:B1,B8 TD-SCDMA:B34,B39 FDD-LTE:B1,B3,B8 TDD-LTE:B38,B39,B40,B41
	L-A	WCDMA:B2,B4,B5 FDD-LTE:B2,B4,B12
	L-AU	GSM/EDGE:850,900,1800MHz WCDMA:B1,B2,B5,B8 FDD-LTE:B1,B3,B4,B5,B7,B8,B28 TDD-LTE:B40
	L-AF	WCDMA:B2,B4,B5 FDD-LTE:B2,B4,B5,B12,B13,B14,B66,B71
	CAT-1	GSM:900,1800 FDD-LTE:B1,B3,B5,B8 TDD-LTE:B34,B38,B39,B40,B41
Wi-Fi (Optional)	Interface	PCIe
	Protocol	IEEE 802.11b/g/n
	Mode	STA, AP
	Frequency	2.4GHz
	Channel Quantity	Ch1 ~ Ch13
	Security	Open, WPA, WPA2
	Encryption	AES, TKIP, TKIPAES
	Number of connections	8 (Max)
	Speed	150Mbps (Max)
	SSID broadcast switch	Support
LED	Quantity	LEDx3(with two programmable LEDs)
Environment	Working	-20 to 85°C, 5～95% RH
	Storage	-20 to 85°C, 5 to 95% RH
Others	Housing	Aluminium housing + stainless steel
	Dimensions	110x83x42mm or 110x83x48mm
	Protection Level	IP30
	Installation	DIN35 rail mounted, wall mounting
	System	Operating System: Raspberry Pi OS

		Kernel: Linux 6.6.78-v8-16k

1.4 Model Selection

1.4.1 Host Model Selection

Model	ETH	USB	HDMI	X board I/O Slot	Y board I/O Slot	Dimension
BL460	1x10/100/1000M	2	X	1x6PIN	X	42x83x110mm
BL460A	1x10/100/1000M	2	X	1x20PIN	X	42x83x110mm
BL460B	1x10/100/1000M	2	X	1x20PIN	2	48x83x110mm
BL460C	1x10/100/1000M	2	X	1x10PIN	X	42x83x110mm
BL461	1x10/100/1000, 1x1x10/100M	2	X	1x6PIN	X	42x83x110mm
BL461A	1x10/100/1000M, 1x10/100M	2	X	1x20PIN	X	42x83x110mm
BL461B	1x10/100/1000, 1x10/100M	2	X	1x20PIN	2	48x83x110mm
BL462	1x10/100/1000M, 2x10/100M	2	1	1x6PIN	X	42x83x110mm
BL462A	1x10/100/1000M, 2x10/100M	2	1	1x20PIN	X	42x83x110mm
BL462B	1x10/100/1000M, 2x10/100M	2	1	1x20PIN	2	48x83x110mm

1.4.2 SOM Selection

Model	MCU	Clock Speed	Wireless	eMMC	LPDDR4X	Temperature
CM5002000	BCM2712	2.4GHz	x	0GB (Lite)	2GB	-20~85℃
CM5002016	BCM2712	2.4GHz	x	16GB	2GB	-20~85℃
CM5002032	BCM2712	2.4GHz	x	32GB	2GB	-20~85℃
CM5004000	BCM2712	2.4GHz	x	0GB (Lite)	4GB	-20~85℃
CM5004016	BCM2712	2.4GHz	x	16GB	4GB	-20~85℃
CM5004032	BCM2712	2.4GHz	x	32GB	4GB	-20~85℃
CM5008000	BCM2712	2.4GHz	x	0GB (Lite)	8GB	-20~85℃
CM5008016	BCM2712	2.4GHz	x	16GB	8GB	-20~85℃
CM5008032	BCM2712	2.4GHz	x	32GB	8GB	-20~85℃
CM5016000	BCM2712	2.4GHz	x	0GB (Lite)	16GB	-20~85℃
CM5016016	BCM2712	2.4GHz	x	16GB	16GB	-20~85℃
CM5016032	BCM2712	2.4GHz	x	32GB	16GB	-20~85℃
CM5016064	BCM2712	2.4GHz	x	64GB	16GB	-20~85℃
CM5102000	BCM2712	2.4GHz	PCB/ext	0GB (Lite)	2GB	-20~85℃
CM5102016	BCM2712	2.4GHz	PCB/ext	16GB	2GB	-20~85℃
CM5102032	BCM2712	2.4GHz	PCB/ext	32GB	2GB	-20~85℃
CM5104000	BCM2712	2.4GHz	PCB/ext	0GB (Lite)	4GB	-20~85℃
CM5104016	BCM2712	2.4GHz	PCB/ext	16GB	4GB	-20~85℃
CM5104032	BCM2712	2.4GHz	PCB/ext	32GB	4GB	-20~85℃
CM5108000	BCM2712	2.4GHz	PCB/ext	0GB (Lite)	8GB	-20~85℃
CM5108016	BCM2712	2.4GHz	PCB/ext	16GB	8GB	-20~85℃
CM5108032	BCM2712	2.4GHz	PCB/ext	32GB	8GB	-20~85℃
CM5108064	BCM2712	2.4GHz	PCB/ext	64GB	8GB	-20~85℃
CM5116000	BCM2712	2.4GHz	PCB/ext	0GB (Lite)	16GB	-20~85℃
CM5116016	BCM2712	2.4GHz	PCB/ext	16GB	16GB	-20~85℃
CM5116032	BCM2712	2.4GHz	PCB/ext	32GB	16GB	-20~85℃
CM5116064	BCM2712	2.4GHz	PCB/ext	64GB	16GB	-20~85℃

1.4.3 X Series I/O Board Selection

You can choose the X-series I/O board based on your needs. The number of pins on the X-series I/O board must match the housing. X-series IO boards with CAN ports, such as the X11, are currently not compatible with the BL460 series.

Note: The default port for this device is RS485. If you need RS232, please specify this to the sales team.

Model	RS232/RS485	CAN	DI	DO	GPIO	PIN	
X10	2	x	x	x	x	6PIN	
X11	x	2	x	x	x	6PIN	Not support
X12	1	1	x	x	x	6PIN	Not support
X13	x	x	2	2	x	6PIN	
X14	x	x	4	x	x	6PIN	
X15	x	x	x	4	x	6PIN	
X16	x	x	x	x	4	6PIN	
X20	4	x	x	x	x	10PIN	
X21	3	1	x	x	x	10PIN	Not support
X22	2	2	x	x	x	10PIN	Not support
X23	4	x	4	4	x	20PIN	
X24	3	1	4	4	x	20PIN	Not support
X25	2	2	4	4	x	20PIN	Not support
X26	2	x	8	4	x	20PIN	
X27	1	1	8	4	x	20PIN	Not support
X28	2	x	12	x	x	20PIN	
X29	1	1	12	x	x	20PIN	Not support
X30	x	x	x	x	16	20PIN	Not support

1.4.4 Y Series I/O Board Selection

You can select the Y-series I/O board based on your needs. Y-series I/O modules are compatible with all Y slots. When the Y63 is selected, you can not choose second Y-series IO board.

Model	Description		Model	Description
Y01	4xDI+4xDO(NPN)		Y41	4xAO, 0~20mA/4~20mA

Y02	4xDI+4xDO(PNP)		Y43	4xAO, 0~5V/0~10V
Y11	8xDI(NPN)		Y46	4xAO, ±5V/±10V
Y12	8xDI(PNP)		Y51	2xRTD, 3-Wire PT100
Y13	8xDI(Dry Contact)		Y52	2xRTD, 3-Wire PT1000
Y21	8xDO(PNP)		Y53	2xRTD, 4-Wire PT100
Y22	8xDO(NPN)		Y54	2xRTD, 4-Wire PT1000
Y24	4xDO(Relay)		Y56	Resistance Measurement
Y31	4xAI, Single-ended, 0~20mA/4~20mA		Y57	Voltage Measurement
Y33	4xAI, Single-ended, 0~5V/0~10V		Y58	4xTC
Y34	4xAI, Differential, 0~5V/0~10V		Y63	4xRS485 or RS232
Y36	4xAI, Differential, ±5V/±10V		Y95	4xPWM Output(NPN) + 4xPulse Counter Input
Y37	4xIEPE		Y96	4xPWM Output(PNP) + 4xPulse Counter Input

Ordering Notes

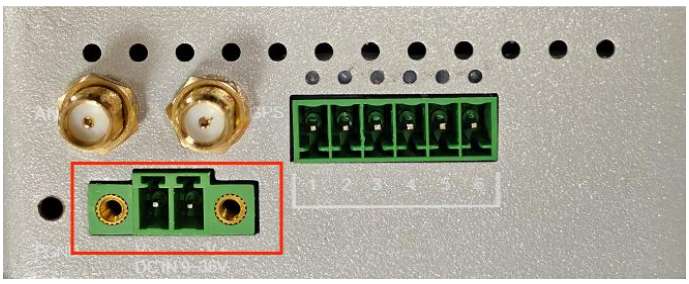
Y01: DI channels support dry contacts or NPN-type wet contact sensors.

Y02: DI channels support dry contacts or PNP-type wet contact sensors.

Y58: Supports thermocouples of types J, K, T, E, R, S, B, and N.

2 Hardware

2.1 Power Interface



Supports 1CH DC12~24V input, with reverse polarity protection.

2.2 I/O Module Port Description

Different X/Y boards offer various serial port options. The currently available board types are as follows. Note: POWER serves as the common negative terminal for DO, COM is used for DI wet contact, and GND is used for DI dry contact.

2.2.1 X Series I/O Board Port Description

X10(2 RS485 or RS232 serial ports)						
Port Number	1	2	3	4	5	6
Name	ttyACM1-A	ttyACM1-B	GND	ttyACM0-A	ttyACM0-B	GND

X13 (2 DO, 2 DI, COM used for DI wet contact and GND used for DI dry contact)						
Port Number	1	2	3	4	5	6
Name	DI1	DI2	GND	DO3	DO4	COM

X14 (4 DI, COM used for DI wet contact and the GND used for DI dry contact.)						
Port Number	1	2	3	4	5	6
Name	DI1	DI2	GND	DI3	DI4	COM

X15 (4 DO)						
Port Number	1	2	3	4	5	6
Name	DO1	DO2	GND	DO3	DO4	GND

X16 (4 GPIO)						
Port Number	1	2	3	4	5	6
Name	GPIO24	GPIO23	GND	GPIO7	GPIO3	GND

X20 (4 RS485 or RS232 serial ports)					
Port Number	1	3	5	7	9
Name	GND	ttyAMA2-A	ttyAMA4-A	ttyACM1-A	ttyACM0-A
Port Number	2	14	16	18	20
Name	GND	ttyAMA2-B	ttyAMA4-B	ttyACM1-B	ttyACM0-B

X23 (4 RS485 or RS232 serial ports, 4 DO, and 4 DI)										
Port Number	1	3	5	7	9	11	13	15	17	19
Name	DI4	DI3	DI2	DI1	COM	GND	ttyAMA2-A	ttyAMA4-A	ttyACM1-A	ttyACM0-A
Port Number	2	4	6	8	10	12	14	16	18	20
Name	DO4	DO3	DO2	DO1	POWER	GND	ttyAMA2-B	ttyAMA4-B	ttyACM1-B	ttyACM0-B

X26 (2 RS485 or RS232 serial ports, 4 DO, and 8 DI)										
Port Number	1	3	5	7	9	11	13	15	17	19
Name	DI6	DI5	DI4	DI3	COM	GND	DI2	DI1	ttyACM1-A	ttyACM0-A
Port Number	2	4	6	8	10	12	14	16	18	20
Name	DO4	DO3	DO2	DO1	POWER	GND	DI8	DI7	ttyACM1-B	ttyACM0-B

X28 (2 RS485 or RS232 serial ports, 12 DI)										
Port Number	1	3	5	7	9	11	13	15	17	19
Name	DI6	DI5	DI4	DI3	COM	GND	DI2	DI1	ttyACM1-A	ttyACM0-A
Port Number	2	4	6	8	10	12	14	16	18	20
Name	DI12	DI11	DI10	DI9	POWER	GND	DI8	DI7	ttyACM1-B	ttyACM0-B

Note: Before using the X board, please first define the GPIO ports according to the input/output modes described above. Restarting the device will reset the port modes to their default state. For usage, refer to the GPIO instructions below. Logic 0 corresponds to a closed state, and logic 1 corresponds to an open state. The mapping of each port to the GPIO pins on the core board can be found in the appendix.

2.2.2 Y Series I/O Board Port Description

Y01 (4-channel DI(Dry contact or NPN type) and 4-channel NPN type DO module)										
Port Number	1	2	3	4	5	6	7	8	9	10
Name	DO1	DO2	DO3	DO4	GND_IOS	DI_COM	DI1	DI2	DI3	DI4

Note: GND_IOS is the common terminal for DO wet contact, and DI_COM is the common terminal for

dry contact.

Y02 (4-channel DI(Dry contact or PNP type) and 4-channel PNP type DO module)										
Port Number	1	2	3	4	5	6	7	8	9	10
Name	DO1	DO2	DO3	DO4	GND_IOS	DI_COM	DI1	DI2	DI3	DI4

Note: GND_IOS is the common terminal for DO wet contact, and DI_COM is the common terminal for dry contact.

Y11 (8-channel NPN Type Digital Input Module)										
Port Number	1	2	3	4	5	6	7	8	9	10
Name	DI1	DI2	DI3	DI4	DI_COM	DI_COM	DI5	DI6	DI7	DI8

Note: DI_COM is the common terminal for dry contacts.

Y12 (8-channel PNP Type Digital Input Module)										
Port Number	1	2	3	4	5	6	7	8	9	10
Name	DI1	DI2	DI3	DI4	DI_COM	DI_COM	DI5	DI6	DI7	DI8

Note: DI_COM is the common terminal for dry contacts.

Y21 (8-channel PNP Type Digital Output Module)										
Port Number	1	2	3	4	5	6	7	8	9	10
Name	DO1	DO2	DO3	DO4	GND_IOS	GND_IOS	DO5	DO6	DO7	DO8

Note: GND_IOS is the common terminal for wet contacts of the DO ports.

Y22 (8-channel NPN Type Digital Output Module)										
Port Number	1	2	3	4	5	6	7	8	9	10
Name	DO1	DO2	DO3	DO4	GND_IOS	GND_IOS	DO5	DO6	DO7	DO8

Note: GND_IOS is the common terminal for wet contacts of the DO ports.

Y24 (4-channel Relay Module)										
Port Number	1	2	3	4	5	6	7	8	9	10
Name	DO1	COM1	DO2	COM2	/	/	DO3	COM3	DO4	COM4

Y31 (4-channel 4~20mA / 0~20mA Analog Single-ended Input Module)										
Port Number	1	2	3	4	5	6	7	8	9	10
Name	AI1+	AI1-	AI2+	AI2-	GND	GND	AI3+	AI3-	AI4+	AI4-

Y33 (4-channel 0~5V / 0~10V Analog Single-ended Input Module)										
Port Number	1	2	3	4	5	6	7	8	9	10
Name	AI1+	AI1-	AI2+	AI2-	/	/	AI3+	AI3-	AI4+	AI4-

Y34 (4-channel 0~5V / 0~10V Analog Differential Input Module)										
Port Number	1	2	3	4	5	6	7	8	9	10
Name	AI1+	AI1-	AI2+	AI2-	/	/	AI3+	AI3-	AI4+	AI4-

Y36 (4-channel -5~5V / -10~10V Analog Differential Input Module)										
Port Number	1	2	3	4	5	6	7	8	9	10
Name	AI1+	AI1-	AI2+	AI2-	/	/	AI3+	AI3-	AI4+	AI4-

Y41 (4-channel 4~20mA / 0~20mA Analog Single-ended Output Module)										
Port Number	1	2	3	4	5	6	7	8	9	10
Name	AO1+	AO1-	AO2+	AO2-	/	/	AO3+	AO3-	AO4+	AO4-

Y43 (4-channel 0~5V / 0~10V Analog Single-ended Output Module)										
Port Number	1	2	3	4	5	6	7	8	9	10
Name	AO1+	AO1-	AO2+	AO2-	/	/	AO3+	AO3-	AO4+	AO4-

Y46 (4-channel -5~5V / -10~10V Analog Single-ended Output Module)										
Port Number	1	2	3	4	5	6	7	8	9	10
Name	AO1+	AO1-	AO2+	AO2-	/	/	AO3+	AO3-	AO4+	AO4-

Y51 (2-channel RTD Module, 3-wire PT100)										
Port Number	1	2	3	4	5	6	7	8	9	10
Name	/	PT1+	PT1-	PT1-	/	/	/	PT2+	PT2-	PT2-

Y52 (2-channel RTD Module, 3-wire PT1000)										
Port Number	1	2	3	4	5	6	7	8	9	10
Name	/	PT1+	PT1-	PT1-	/	/	/	PT2+	PT2-	PT2-

Y53 (2-channel RTD Module, 4-wire PT100)										
Port Number	1	2	3	4	5	6	7	8	9	10
Name	PT1+	PT1+	PT1-	PT1-	/	/	PT2+	PT2+	PT2-	PT2-

Y54 (2-channel RTD Module, 4-wire PT1000)										
Port Number	1	2	3	4	5	6	7	8	9	10
Name	PT1+	PT1+	PT1-	PT1-	/	/	PT2+	PT2+	PT2-	PT2-

Y58 (4-channel Thermocouple Temperature Measurement Module)										
Port Number	1	2	3	4	5	6	7	8	9	10
Name	T1+	T1-	T2+	T2-	/	/	T3+	T3-	T4+	T4-

Y63(4-channels RS485 or RS232)										
Port Number	1	2	3	4	5	6	7	8	9	10
Name	ttyWCH0-A	ttyWCH0-B	ttyWCH1-A	ttyWCH1-B	GND	GND	ttyWCH2-A	ttyWCH2-B	ttyWCH3-A	ttyWCH3-B

2.2.3 RS485 Usage

If the device is equipped with four RS485 ports, first check whether the serial devices ttyAMA and ttyACM are detected.

```
ls /dev
```

If the devices ttyAMA2 and ttyAMA4 are not detected, check whether the following configuration has been added in the startup file to multiplex the UART to GPIO:

Enable UART2 and multiplex it to GPIO4 (TXD) and GPIO5 (RXD).

```
sudo nano /boot/firmware/config.txt          //Open the startup file

dtoverlay=uart2,txd2_pin=4,rxd2_pin=5     //Add the following configuration at the end of the file
```

Enable UART4 and multiplex it to GPIO12 (TXD) and GPIO13 (RXD).

```
dtoverlay=uart4,txd4_pin=12,rxd4_pin=13
```

In the BL460, ttyACM0-TX and ttyACM0-RX represent a single RS485 serial line. When using the RS485 interface, connect the RS485 cable to the corresponding port. For example, the X10 module has two ports, ttyACM0-A and ttyACM0-B, with the device file /dev/ttyACM0. Configure the serial port

with a baud rate of 115200, 8 data bits, no parity, and 1 stop bit (8N1).

```
stty -F /dev/ttyACM0 ispeed 115200 ospeed 115200 cs8
echo 12345 > /dev/ttyACM0          //Send data via the RS485-1 port.
cat /dev/ttyACM0                   //Wait to check the received data.
```

Press “Ctrl+C” to stop.

2.2.4 RS232 Usage

In the BL460, ttyACM0-RX and ttyACM0-TX represent a single RS232 serial line. When using the RS232 interface, connect the RS232 cable to the corresponding port. For example, the X10 module has two ports, ttyACM0-RX and ttyACM0-TX, with the device file /dev/ttyACM0. Configure the serial port with a baud rate of 115200, 8 data bits, no parity, and 1 stop bit (8N1).

```
stty -F /dev/ttyACM0 ispeed 115200 ospeed 115200 cs8
echo 12345 > /dev/ttyACM0          //Send data via the RS232-  1 port
cat /dev/ttyACM0                   //Wait to check the received data.
```

Press "Ctrl+C" to stop.

2.2.5 GPIO Usage

Enter the /sys/class/beilai/ directory and type ls to view the controllable GPIO ports:
The device comes with WiringPi preinstalled by default. If WiringPi is not installed, you can navigate to the /usr/demo/wiringpi/ directory and enter the following command to install the WiringPi package.

```
root@BL460:~#usr/demo/wiringpi# dpkg -i wiringpi_3.16_arm64.deb
```

Enter gpio readall to view all controllable GPIO pins:

```
root@BL460:~# gpio readall
```

```
root@BL460:~# gpio readall
 +-----+-----+---------+------+---+---CM5----+---+------+---------+-----+-----+
 | BCM | wPi |   Name  | Mode | V | Physical | V | Mode | Name    | wPi | BCM |
 +-----+-----+---------+------+---+----++----+---+------+---------+-----+-----+
 |     |     |    3.3v |      |   |  1 || 2  |   |      | 5v      |     |     |
 |   2 |   8 |   SDA.1 |  OUT | 0 |  3 || 4  |   |      | 5v      |     |     |
 |   3 |   9 |   SCL.1 |    - | 0 |  5 || 6  |   |      | 0v      |     |     |
 |   4 |   7 | GPIO. 7 |    - | 0 |  7 || 8  | 1 | ALT4 | TxD     |  15 |  14 |
 |     |     |      0v |      |   |  9 || 10 | 1 | ALT4 | RxD     |  16 |  15 |
 |  17 |   0 | GPIO. 0 |  OUT | 1 | 11 || 12 | 0 |    - | GPIO. 1 |   1 |  18 |
 |  27 |   2 | GPIO. 2 |    - | 0 | 13 || 14 |   |      | 0v      |     |     |
 |  22 |   3 | GPIO. 3 |    - | 0 | 15 || 16 | 0 |    - | GPIO. 4 |   4 |  23 |
 |     |     |    3.3v |      |   | 17 || 18 | 0 |    - | GPIO. 5 |   5 |  24 |
 |  10 |  12 |    MOSI |    - | 0 | 19 || 20 |   |      | 0v      |     |     |
 |   9 |  13 |    MISO |    - | 0 | 21 || 22 | 0 |    - | GPIO. 6 |   6 |  25 |
 |  11 |  14 |    SCLK |    - | 0 | 23 || 24 | 0 |    - | CE0     |  10 |   8 |
 |     |     |      0v |      |   | 25 || 26 | 0 |    - | CE1     |  11 |   7 |
 |   0 |  30 |   SDA.0 |  OUT | 1 | 27 || 28 | 1 |   IN | SCL.0   |  31 |   1 |
 |   5 |  21 | GPIO.21 |    - | 0 | 29 || 30 |   |      | 0v      |     |     |
 |   6 |  22 | GPIO.22 |  OUT | 0 | 31 || 32 | 0 |    - | GPIO.26 |  26 |  12 |
 |  13 |  23 | GPIO.23 |    - | 0 | 33 || 34 |   |      | 0v      |     |     |
 |  19 |  24 | GPIO.24 |    - | 0 | 35 || 36 | 0 |    - | GPIO.27 |  27 |  16 |
 |  26 |  25 | GPIO.25 |    - | 0 | 37 || 38 | 0 |    - | GPIO.28 |  28 |  20 |
 |     |     |      0v |      |   | 39 || 40 | 0 |    - | GPIO.29 |  29 |  21 |
 +-----+-----+---------+------+---+----++----+---+------+---------+-----+-----+
 | BCM | wPi |   Name  | Mode | V | Physical | V | Mode | Name    | wPi | BCM |
 +-----+-----+---------+------+---+---CM5----+---+------+---------+-----+-----+
```

Here, BCM represents the pin address in the register (i.e., the GPIO pin name on the CM5), and wPi represents the corresponding value used in WiringPi (i.e., the pin name used for GPIO operations).

For example, on the X26 module, to configure DO4 as an output and set its logic value to 1:

```
root@BL460:~# gpio mode 3 out
root@BL460:~#gpio write 3 1
```

For example, on the X16 module, to configure GPIO24 as an input and read its current logic level:

```
root@BL460:~# gpio mode 5 in
root@BL460:~# gpio read 5
```

Note:

GPIO14 and GPIO15 are multiplexed with the motherboard debug port.

GPIO2 is multiplexed with the running indicator LED.

GPIO17 is multiplexed with the network status LED.

GPIO6 and ID_SD are multiplexed with the hardware watchdog.

Modifying these GPIOs may cause the above functions to stop working.

When using the Y board:

Slot 1 uses GPIO8, GPIO9, GPIO10, GPIO11, and ID_SC.

Slot 2 uses multiplexed GPIO1, GPIO18, GPIO19, GPIO20, GPIO21, and GPIO26 to communicate with the motherboard.

2.2.6 Y board Serial Port Usage

(1) Software Installation

The corresponding files are located in the /usr/demo/iolib/ directory. Please use the actual file names as they appear. For the dual Y-board configuration, first execute the command chmod +x BEILAI_IOy_raspberry_V1_20251107.bin, and then install the software.

```
root@BL460:~# cd /usr/demo/iolib/
root@BL460:~# chmod +x BEILAI_IOy_raspberry_V1_20251107.bin
root@BL460:~# ./BEILAI_IOy_raspberry_V1_20251107.bin
Md5 verify pass!
Install complete!
```

For the configuration with the Y63 board, first execute the command chmod +x BEILAI_IOy_raspberryy63_V1_20251107.bin, and then install the software.

```
root@BL460:~# cd /usr/demo/iolib/
root@BL460:~# chmod +x BEILAI_IOy_raspberryy63_V1_20251107.bin
root@BL460:~# ./BEILAI_IOy_raspberryy63_V1_20251107.bin
Md5 verify pass!
Install complete!
```

(2) Usage of Y Board Ports

Note: This usage method does not apply to the Y63 module. For instructions on using the Y63 module, refer to section 2.2.7.

Use ioy show to view IO board information. Use ioy help to see command help.

```
root@BL460:~# ioy help
Usage: ioy <command> [<arguments>]

Commands:
        show
        get             <address>|<slot>.<channel>
        set             <address>|<slot>.<channel> <value>
        config          <address>|<slot>.<channel> mode <mode>,
                        <address>|<slot>.<channel> min <min-value> max <max-value>

config mode:
        ai|ao           4t20(4~20mA),0t20(0~20mA),0t5(0~5V),0t10(0~10V),
                        -5t5(-5~5V),-10t10(-10~10V)
        rtd             pt100-3(pt100 3wire),pt100-4(pt100 4wire),
                        pt1000-3(pt1000 3wire),pt1000-4(pt1000 4wire)
        tc              k,j,e,t,s,r,b,n
```

For example, using the DI module, short DI2 and enter ioy show to view the information.

```
slot    name    channel  address  mode   value      min        max
2       Y12     1        2000     *      0          0.0000     0.0000
2       Y12     2        2001     *      1          0.0000     0.0000
2       Y12     3        2002     *      0          0.0000     0.0000
2       Y12     4        2003     *      0          0.0000     0.0000
2       Y12     5        2004     *      0          0.0000     0.0000
2       Y12     6        2005     *      0          0.0000     0.0000
2       Y12     7        2006     *      0          0.0000     0.0000
2       Y12     8        2007     *      0          0.0000     0.0000
```

You can also use the get command to retrieve the channel values.

```
root@BL460:~# ioy get 2004      //You can view it using the address command.
```

```
address 2004 value 1
root@BL460:~# ioy get 2.5        //You can view it using <slot>.<channel>.
slot 2 channel 5 value 1
```

For example, using the AO module, enter ioy show to view the information.

```
slot    name    channel    address    mode    value       min         max
2       Y41     1          4000       4t20    4.0000      4.0000      20.0000
2       Y41     2          4002       4t20    4.0000      4.0000      20.0000
2       Y41     3          4004       4t20    4.0000      4.0000      20.0000
2       Y41     4          4006       4t20    4.0000      4.0000      20.0000
2       Y41     5          4008       4t20    4.0000      4.0000      20.0000
2       Y41     6          4010       4t20    4.0000      4.0000      20.0000
2       Y41     7          4012       4t20    4.0000      4.0000      20.0000
2       Y41     8          4014       4t20    4.0000      4.0000      20.0000
```

In the mode column, it is shown as 4t20, which corresponds to the 4–20 mA current output in ioy help → config mode. Use the set command to configure the channel value:

```
root@BL460:~# ioy set 4000 10    //Set the output to 10mA using the address command.
root@BL460:~# ioy set 2.1 10     //Set it using <slot>.<channel>.
root@BL460:~# ioy get 4000
address 4000 value 10.000000
```

(3) Port Configuration

By using the ioy help command, you can see the command format for config.

```
Usage: ioy <command> [<arguments>]

Commands:
        show
        get             <address>|<slot>.<channel>
        set             <address>|<slot>.<channel> <value>
        config          <address>|<slot>.<channel> mode <mode>,
                        <address>|<slot>.<channel> min <min-value> max <max-value>

config mode:
        ai|ao           4t20(4~20mA),0t20(0~20mA),0t5(0~5V),0t10(0~10V),
                        -5t5(-5~5V),-10t10(-10~10V)
        rtd             pt100-3(pt100 3wire),pt100-4(pt100 4wire),
                        pt1000-3(pt1000 3wire),pt1000-4(pt1000 4wire)
        tc              k,j,e,t,s,r,b,n
```

To change the 4–20 mA range to 0–20 mA, either of the following two commands can be used:

```
root@BL460:~# ioy config 4000 mode 0t20
root@BL460:~# ioy config 2.1 mode 0t20        //Change the range to 0–20 mA.
```

Set the corresponding minimum and maximum values for the range change:

```
root@BL460:~# ioy config 4000 min 0 max 20 //Set the minimum and maximum values to 0 and 20.
```

2.2.7 Y63 Board Usage

In the BL460, ttyWCHX-A and ttyWCHX-B represent a single RS485 serial line. When using the

RS485 interface, connect the RS485 cable to the corresponding port. For example, the Y63 module has two ports, ttyWCH0-A and ttyWCH0-B, with the device file /dev/ttyWCH0. Configure the serial port with a baud rate of 115200, 8 data bits, no parity, and 1 stop bit (8N1).

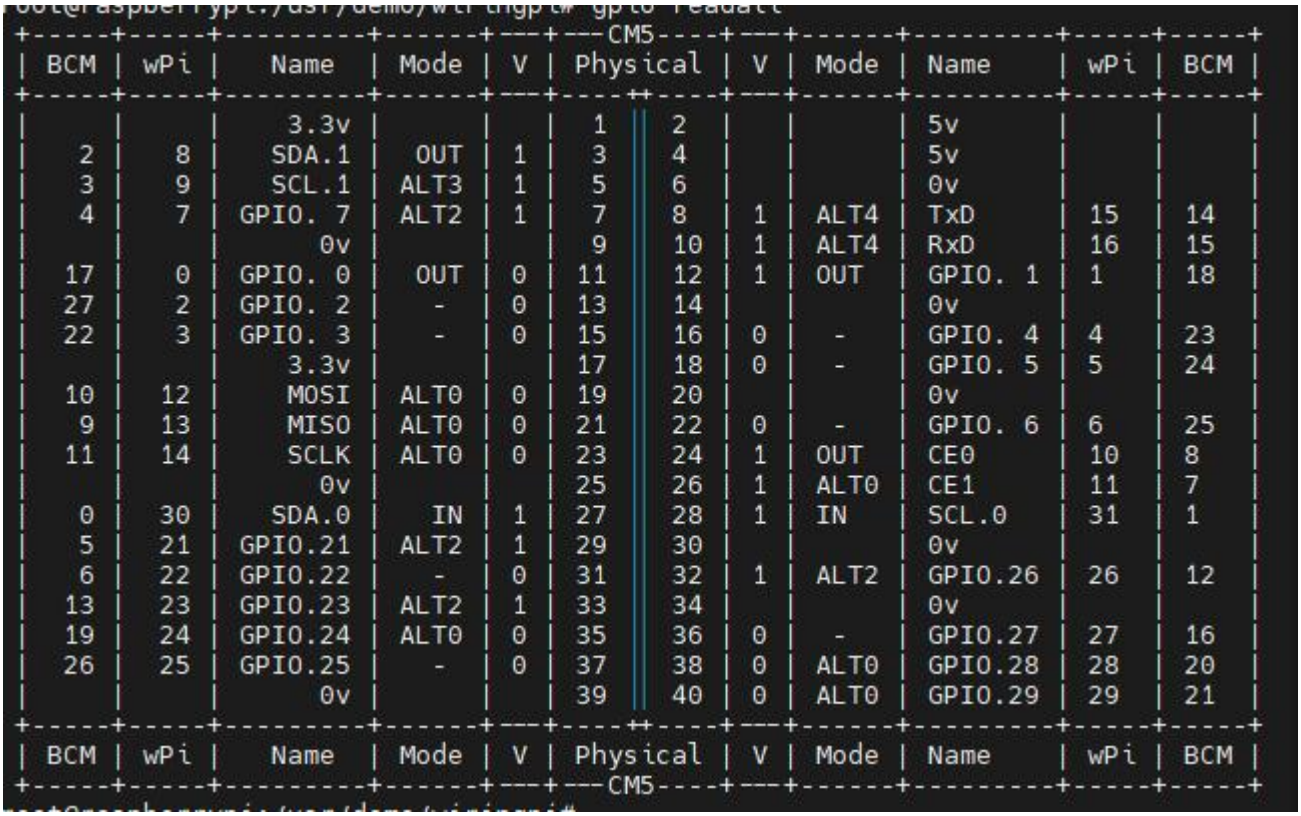
```
stty -F /dev/ttyWCH0 ispeed 115200 ospeed 115200 cs8
echo 12345 > /dev/ttyWCH0          //Send data via the RS485-  1 port.
cat /dev/ttyWCH0                   //Wait to check the received data.
```

Press "Ctrl+C" to stop.

2.2.8 Instructions for using the 40-pin expansion board

Enter gpio readall to view all controllable GPIO pins:

```
root@BL460:~# gpio readall
```



BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM
		3.3v			1 || 2			5v		
2	8	SDA.1	OUT	1	3 || 4			5v		
3	9	SCL.1	ALT3	1	5 || 6			0v		
4	7	GPIO. 7	ALT2	1	7 || 8	1	ALT4	TxD	15	14
		0v			9 || 10	1	ALT4	RxD	16	15
17	0	GPIO. 0	OUT	0	11 || 12	1	OUT	GPIO. 1	1	18
27	2	GPIO. 2	-	0	13 || 14			0v		
22	3	GPIO. 3	-	0	15 || 16	0	-	GPIO. 4	4	23
		3.3v			17 || 18	0	-	GPIO. 5	5	24
10	12	MOSI	ALT0	0	19 || 20			0v		
9	13	MISO	ALT0	0	21 || 22	0	-	GPIO. 6	6	25
11	14	SCLK	ALT0	0	23 || 24	1	OUT	CE0	10	8
		0v			25 || 26	1	ALT0	CE1	11	7
0	30	SDA.0	IN	1	27 || 28	1	IN	SCL.0	31	1
5	21	GPIO.21	ALT2	1	29 || 30			0v		
6	22	GPIO.22	-	0	31 || 32	1	ALT2	GPIO.26	26	12
13	23	GPIO.23	ALT2	1	33 || 34			0v		
19	24	GPIO.24	ALT0	0	35 || 36	0	-	GPIO.27	27	16
26	25	GPIO.25	-	0	37 || 38	0	ALT0	GPIO.28	28	20
		0v			39 || 40	0	ALT0	GPIO.29	29	21
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM

Here, BCM represents the pin address in the register (i.e., the GPIO pin name on the CM5), and wPi represents the corresponding value used in WiringPi (i.e., the pin name used for GPIO operations).

The basic setup commands are as follows:

```
root@BL460:~# gpio mode 3 in          //Set GPIO22 to input mode.
root@BL460:~# gpio mode 3 out         //Set GPIO22 to output mode.
root@BL460:~# gpio write 3 1          //Set the value of GPIO22 to 1.
root@BL460:~# gpio read 3             //Read the value of GPIO22.
```

Note:

GPIO14 and GPIO15 are multiplexed with the motherboard debug port.
GPIO2 is multiplexed with the running indicator LED.
GPIO17 is multiplexed with the network status LED.
GPIO6 and ID_SD are multiplexed with the hardware watchdog.
Modifying these GPIOs may cause the above functions to stop working.
When shared with the X board, GPIOs 3, 4, 5, 7, 12, 13, 16, 22, 23, 24, 25, and 27 are shared with the 40-pin header and may affect each other.
When shared with the Y board:
Slot 1 uses GPIOs 8, 9, 10, 11, and ID_SC.
Slot 2 uses GPIOs 1, 18, 19, 20, 21, and 26.
Modifying these GPIOs may cause the Y board connection to fail.

2.3 LED

LED	Description
PWR	Power LED: It remains constantly on when the power is connected. This LED light cannot be programmed by the user.
RUN	Default Settings: The LED blinks when the CPU usage is below 90% and remains on continuously when the CPU usage exceeds 90% This LED light can be programmed by the user.
LINK	Default Settings: The LED remains on when there is an internet connection and turns off when there is no internet connection. This LED light can be programmed by the user.

The LED indicators are as follows:

The LEDs from left to right are LED2, LED1, and LED0.

LED2 is the POWER indicator; it stays on when the device is powered and the power is normal.

LED1 is the RUN indicator; it blinks when the system is running normally.

LED0 is the LINK indicator; it stays on when connected to the internet via wired network, and blinks when using 4G or Wi-Fi.

The control script is located at /etc/beilai_led.sh.

Users can also control the LEDs using WiringPi.

Enter gpio readall to view all controllable GPIO pins:

```
root@BL460:~#usr/demo/wiringpi# gpio readall
```

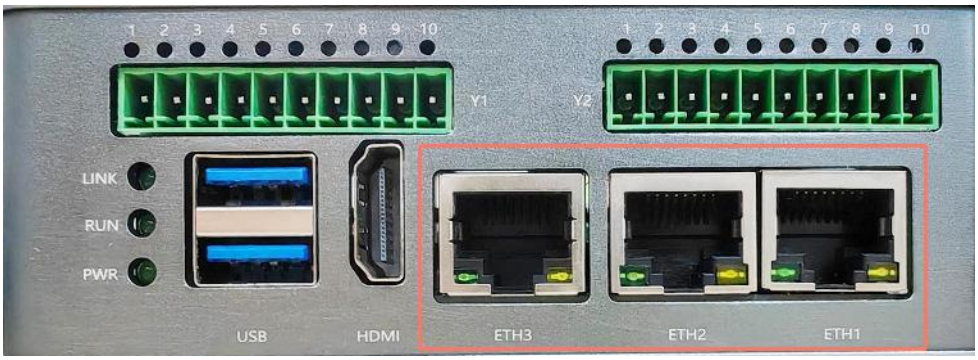
Control LED0 on:

root@BL460:~#usr/demo/wiringpi# gpio write 0 1

Control LED1 off:

root@BL460:~#usr/demo/wiringpi# gpio write 8 0

2.4 Ethernet Port



The device comes with a maximum of 3 network ports.

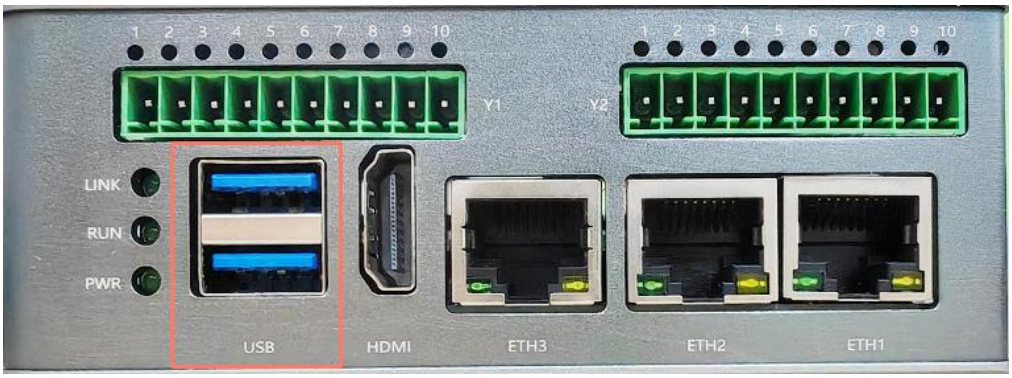
2.4.1 Configure a static IP for ETH1

Execute the following commands in the terminal:

```
sudo nano /etc/NetworkManager/system-connections/static-eth0.nmconnection
//Modify the static IP configuration file.

[connection]
id=static-eth1                          //The network interface connection ID to be modified.
type=ethernet
interface-name=eth1                     //The name of the network interface to be modified.
autoconnect=true

[ipv4]
method=manual                           //Set to manual IP configuration.
addresses1=192.168.1.110/24,192.168.1.1     //The IP address and gateway to be modified.
dns=8.8.8.8;1.1.1.1;
```

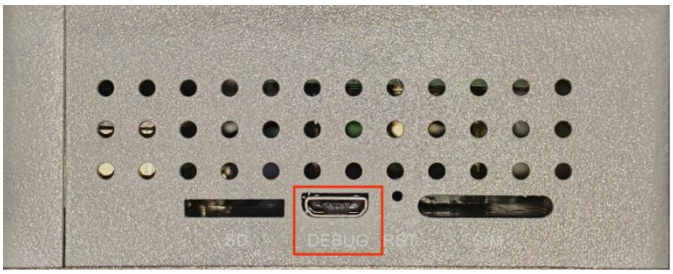
```
[ipv6]
method=auto                    //IPSet to obtain an IP address automatically.
```

2.5 USB Port



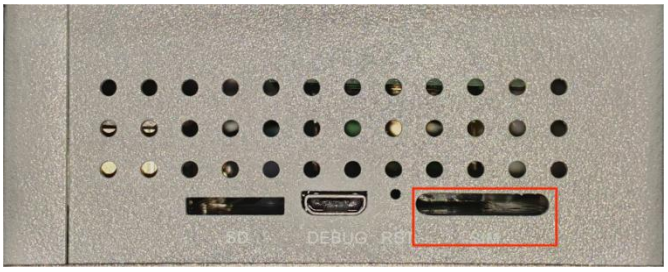
The device has 2 USB 3.0 HOST interfaces, supporting FAT32 formatted USB drives. When reading or writing data to the USB drive, use the sync command to ensure data is properly saved and prevent data loss.

2.6 Debugging Serial Port

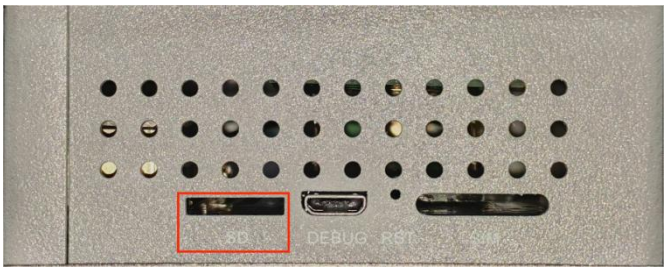


The debugging interface is as shown in the image. You can access the device's system through this port.

2.7 SIM Card Slot

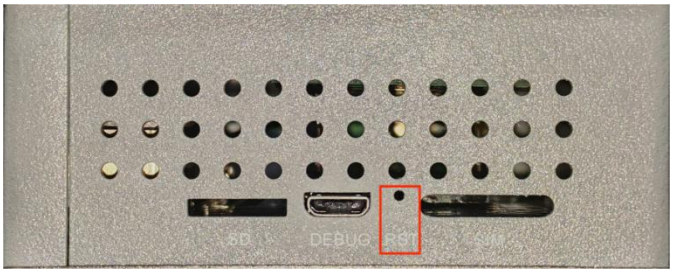


2.8 SD Card Slot



As shown in the figure, the SD card slot is available. Core boards with eMMC do not support using an SD card. Only core boards without eMMC can use an SD card as the system boot device.

2.9 Reset Button



The reset button is shown in the figure. Press and hold for approximately 5 seconds to power off the device. Then, press briefly once to restart the device.

2.10 M.2 Interface

The M.2 interface supports SSDs and Hailo AI accelerator modules, among other peripherals. SSDs can be used immediately upon insertion, and their status can be checked using the lsblk command.

The Hailo AI accelerator module is based on its proprietary AI processor, providing high-performance, low-power AI inference capabilities for edge devices.

Once the Hailo module is connected, the system performs hardware detection, and the PCIe interface is automatically enabled upon hardware connection.

Enter the following command to update the firmware.

```
sudo apt update
sudo rpi-eeprom-update
sudo rpi-eeprom-update -a
```

Enter the following command to install the dependencies required for using the AI Kit online.

```
sudo apt install hailo-all
```

```
sudo reboot                    //After the installation is complete, restart the device.
```

Enter hailortcli fw-control identify to check whether the driver is functioning properly.

```
pi@raspberrypi:~ $ hailortcli fw-control identify
Executing on device: 0000:01:00.0
Identifying board
Control Protocol Version: 2
Firmware Version: 4.17.0 (release,app,extended context switch buffer)
Logger Version: 0
Board Name: Hailo-8
Device Architecture: HAILO8L
Serial Number: HLDDLBB241602672
Part Number: HM21LB1C2LAE
Product Name: HAILO-8L AI ACC M.2 B+M KEY MODULE EXT TMP
```

You can also run dmesg | grep -i hailo to check the logs.

```
pi@raspberrypi:~ $ dmesg | grep -i hailo
[    3.186391] hailo: Init module. driver version 4.17.0
[    3.186521] hailo 0000:01:00.0: Probing on: 1e60:2864...
[    3.186526] hailo 0000:01:00.0: Probing: Allocate memory for device extension, 11600
[    3.186548] hailo 0000:01:00.0: enabling device (0000 -> 0002)
[    3.186554] hailo 0000:01:00.0: Probing: Device enabled
[    3.186575] hailo 0000:01:00.0: Probing: mapped bar 0 - 00000000f40ffb0d 16384
[    3.186583] hailo 0000:01:00.0: Probing: mapped bar 2 - 00000000b69b1e1d 4096
[    3.186587] hailo 0000:01:00.0: Probing: mapped bar 4 - 000000008abfcd4f 16384
[    3.186591] hailo 0000:01:00.0: Probing: Force setting max_desc_page_size to 4096 (recommended value is 16384)
[    3.186601] hailo 0000:01:00.0: Probing: Enabled 64 bit dma
[    3.186604] hailo 0000:01:00.0: Probing: Using userspace allocated vdma buffers
[    3.186608] hailo 0000:01:00.0: Disabling ASPM L0s
[    3.186612] hailo 0000:01:00.0: Successfully disabled ASPM L0s
[    3.415197] hailo 0000:01:00.0: Firmware was loaded successfully
[    3.430033] hailo 0000:01:00.0: Probing: Added board 1e60-2864, /dev/hailo0
```

2.11 PCIe

The PCIe interface supports both 4G and Wi-Fi.

2.11.1 4G Module

Using the Quectel EC20 module as an example, place the SIM card into the module and connect the antenna. The test program can be found in the /usr/demo/4G directory.

(1) Network Function

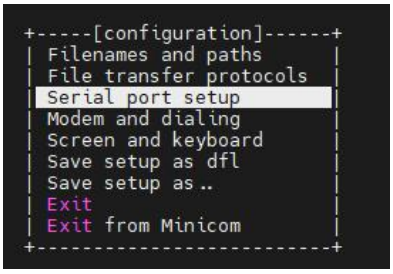
Disable other network connections and keep only the 4G module network active.

```
ifconfig eth1 down

ifconfig eth2 down

ifconfig eth3 down

ifconfig
```

At this point, a network interface eth4 should have been assigned an IP. If no IP is obtained for this interface, the module may not have its network function enabled by default. You can try configuring the 4G module using the following commands. (For EC200 series modules, the AT command port is

/dev/ttyUSB1.)

```
sudo apt-get install minicom          //Install a serial communication tool.
minicom -s
```



After saving, select Exit to quit, then enter the following command (the screen may go blank and not show input; just press Enter to get an OK response):

```
AT+QCFG="USBNET",1
```

If using the EC200 module, an additional command is required to enable network connectivity:

```
AT+QNETDEVCTL=3,1,1
```

Once the command is executed and the device returns "OK", the configuration is successful. This setup only needs to be done once. Press Ctrl + A followed by X to exit. Restart the device to generate the eth4 interface IP, then re-execute the network stop and start commands. After the eth4 IP is generated, run the following commands to test whether the network is functioning properly.

```
ping www.baidu.com -I eth4
```

If ping fails, it may be due to a DNS configuration issue. Add a DNS address using the following command:

```
sudo nano /etc/resolv.conf
Add nameserver 8.8.8.8 to the file.
```

(2) SMS Function

To test the SMS functionality, execute the test command in the program directory.

```
./send_sms <device> <phonenumber> <text>
```

Command explanation: <device> is the 4G module device node. <phonenumber> is the recipient phone number. <text> is the SMS content. There must be no spaces between characters in the message, otherwise an error will occur.

For example: ./send_sms /dev/ttyUSB2 152******** test

The corresponding number should receive an SMS with the content "test".

(3) Call Function

Execute the test command in the test program directory to test the dialing function:

```
./phone_call <device> <phonenumber>
```

Command explanation: <device> is the 4G module device node. <phonenumber> is the target phone number.

For example: ./phone_call /dev/ttyUSB2 152********

The corresponding number should receive a call from the device.

(4) GPS Function

Execute the test command in the test program directory to test the GPS function:

```
./get_location <device> <timeout>
```

Command explanation: <device> is the device node, which can be checked using the command ls /dev/ttyUSB*; it may change after the device is restarted. <timeout> is the time to wait for latitude and longitude information (in seconds).

For example: ./get_location /dev/ttyUSB2 1

Obtaining latitude and longitude may take a few minutes. If it fails or times out, please check that the antenna is properly connected and ensure testing is conducted in an open area.

(5) 4G Module Usage

In some countries, the 4G module may not be able to access the Internet using standard commands. In this case, the 4G module should be switched to USB interface mode, which should generate a network interface wwan0. If this interface does not appear, the module may not have its network function enabled by default. You can try configuring the 4G module using the following commands. (For EC200 series modules, the AT command port is /dev/ttyUSB1.)

```
microcom /dev/ttyUSB2

AT+QCFG="usbnet",0

Then exit
```

When using the 4G module, an APN as well as a username and password are required. This can be configured using the apn_config application in the /usr/demo/apn directory. To set the APN, execute the command in the following format:

```
usage:   ./apn_config <<device>> <<apn>> <<username>> <<passwd>>

  eg:    ./apn_config /dev/ttyUSB2   your_apn 13723634521 *****
```

Note: your_apn is the carrier name, username is the name provided by the carrier, and passwd is the password provided by the carrier.

Before use, ensure that the wwan0 network interface is enabled. The command to enable it is:

```
ifconfig wwan0 up
```

2.11.2 Wi-Fi Module

The Raspberry Pi core board comes with a built-in Wi-Fi module. Once the antenna is connected, it can connect to a Wi-Fi network.
For versions with an HDMI, Wi-Fi can be connected directly via the desktop icon.
By default, the core board uses the onboard antenna for Wi-Fi. If an external antenna is used, it must be added in /boot/firmware/config.txt.

```
dtparam=ant2
#dtparam=ant1                    //Select the onboard antenna.
#dtparam=ant2                    //Select the external antenna.
```

Set the country code for the wireless network.

```
sudo raspi-config nonint do_wifi_country <country>
```

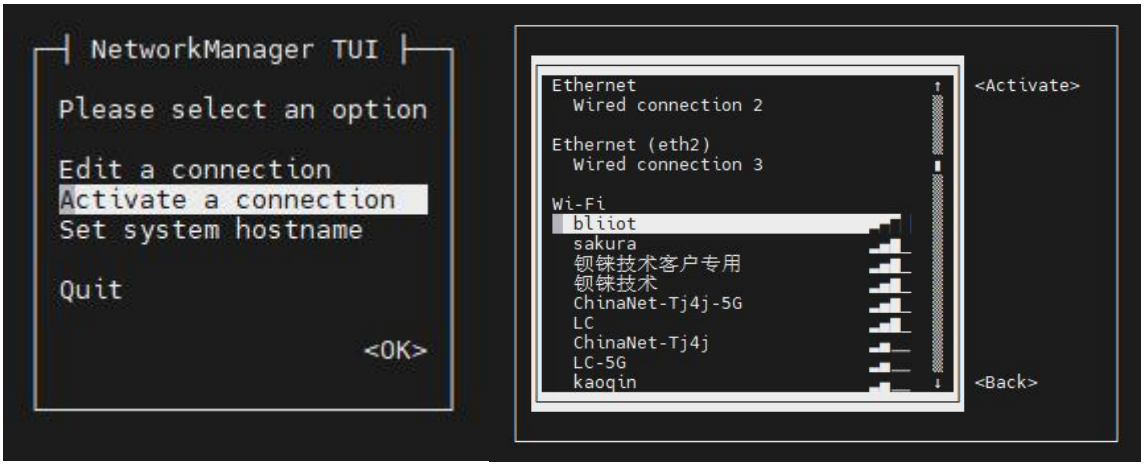
(1) STA function
Connect to the Wi-Fi network.

```
ifconfig eth1 down
ifconfig eth2 down
ifconfig eth3 down                  //Disable other networks.
sudo nmcli radio wifi on            //Enable WiFi
sudo nmcli dev wifi list            //View available WiFi networks.
sudo nmcli dev wifi connect <SSID> password <password>
//Replace <SSID> with your WiFi name and <password> with your WiFi password.
sudo nmcli dev wifi connect <SSID> hidden yes
//Connect to a hidden network.
sudo nmcli connection modify <SSID> connection.autoconnect yes
//Set to connect automatically.
```

You can use ifconfig to check the obtained IP address, and execute the following commands to test whether the network is functioning properly.

```
ping www.baidu.com -I wlan0
```

To connect using the graphical interface, enter the command sudo nmtui to open the interface. Select Activate a connection to enter the Wi-Fi selection screen, then choose the Wi-Fi network and enter the password to connect.



(2) AP function(Currently only supports the Wi-Fi module built into the core board)

Use the following command to quickly create a hotspot.

```
sudo nmcli device wifi hotspot ssid hotspot password 88888888

// The hotspot name follows ssid, and the hotspot password follows password
```

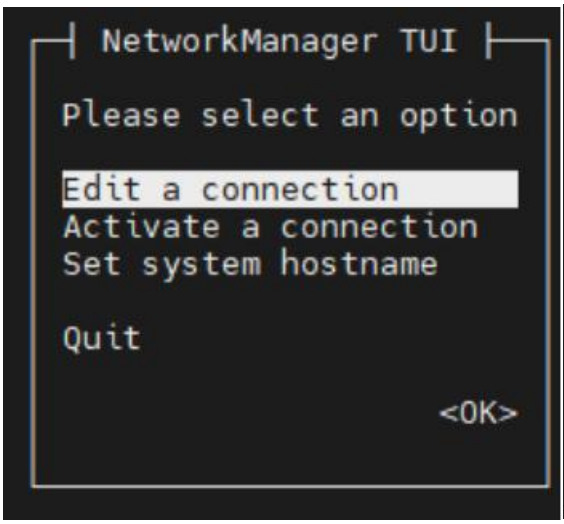
You can also create a standalone hotspot connection configuration.

```
nmcli connection add type wifi ifname wlan0 con-name my-hotspot autoconnect no ssid “Myhotspot” mode ap

//Create a hotspot connection configuration (con-name is the connection name, and ssid is the Wi-Fi name).

nmcli connection modify my-hotspot 802-11-wireless-security.key-mgmt wpa-psk 802-11-wireless-security.psk “88888888”

//Configure hotspot security parameters, where 88888888 is the Wi-Fi password.

nmcli connection modify my-hotspot ipv4.method shared

//Configure IP and network sharing.

nmcli connection up my-hotspot

//Start the network.
```

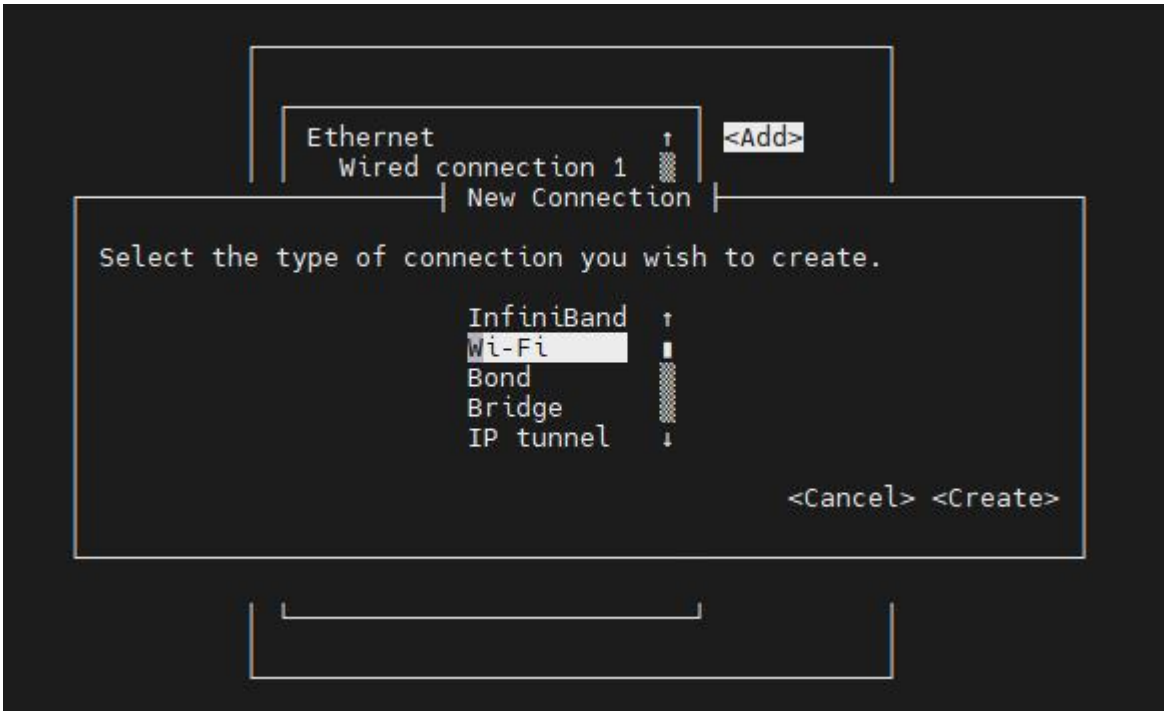
You can now find your Wi-Fi hotspot using your computer or mobile phone.

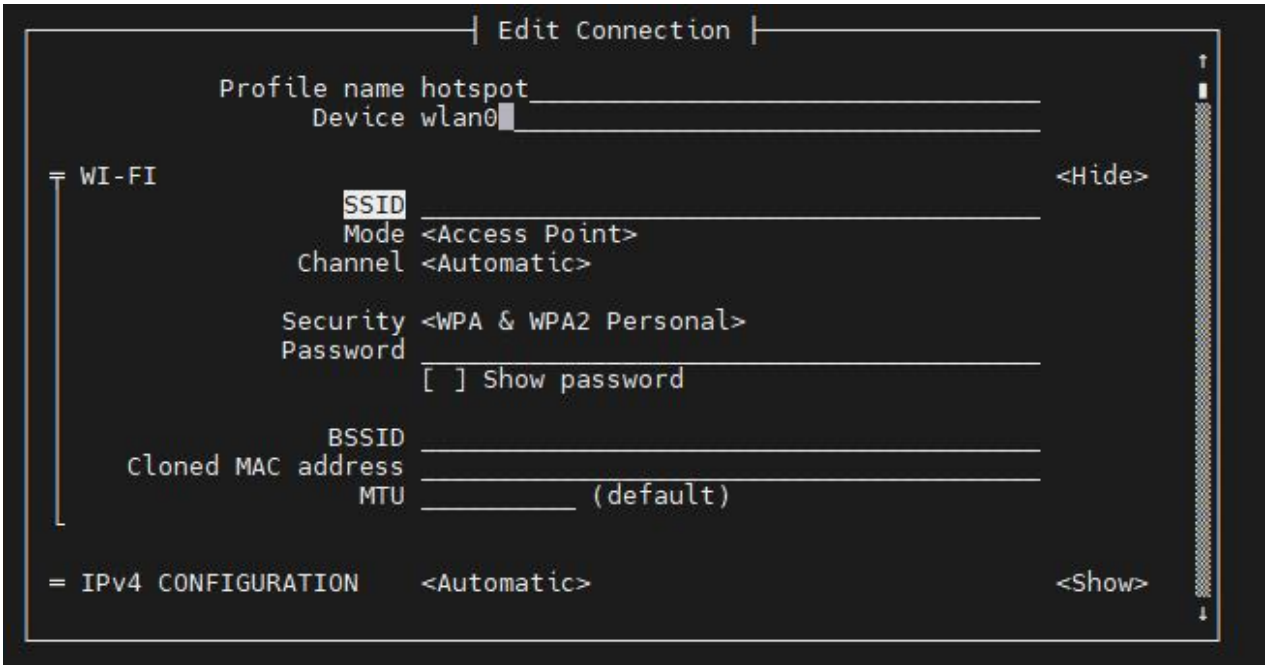
You can also use the graphical user interface to set up the hotspot configuration.

Enter the command sudo nmtui to access the GUI, select Edit a connection, then choose Add to enter the network creation interface.



Select Wi-Fi to enter the network configuration settings. In Mode, choose Access Point for hotspot mode, configure the settings as needed, and click OK to save. Then use the command nmcli connection up <Profile name> (the connection name you set) to start the hotspot.

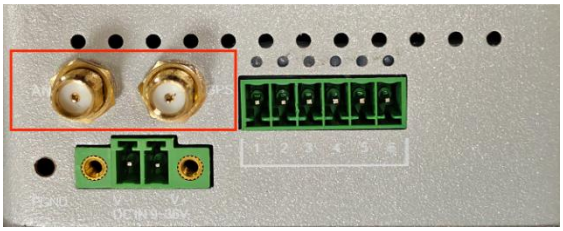




Use the following command to disable the hotspot.

```
sudo nmcli device disconnect wlan0          //Disable the hotspot.
sudo nmcli device up wlan0                  //Reconnect to WiFi
```

2.12 Antenna Interface



The antenna interfaces include one Wi-Fi/cellular antenna interface and one GPS antenna interface.

Note: When both Wi-Fi and 4G modules are present, the GPS port connects to the Wi-Fi signal antenna; if there is no Wi-Fi, it can connect to the GPS antenna. The ANT interface is used to connect the 4G signal antenna.

2.13 Hardware Watchdog

The hardware watchdog has a default timeout of 30ms. To disable the hardware watchdog:

```
root@BL460:~#gpio write 22 1
```

If the hardware watchdog module fails to start properly, you need to reload the w_dog.ko module from the directory:

/lib/modules/6.6.78-v8-16k/kernel/drivers/

```
root@BL460:~#sudo insmod /lib/modules/6.6.78-v8-16k/kernel/drivers/w_dog/w_dog.ko.xz
```

2.14 Encryption Chip

The encryption chip model is RJGT102. It uses a SHA-256–based encryption and authentication algorithm, and also provides a configurable watchdog timer and external reset function. It communicates with the MCU via the I²C-5 serial interface and supports a low-power mode.
The device uses a demo of the encryption chip by writing /proc/sys/kernel/random/uuid into the chip while also saving the UUID to /usr/rjgt_unique.json. During use, the data from the encryption chip is retrieved and compared; if the external data matches the internal data in the chip, the encryption verification passes.
Before use, modify the cross-compiler path in the Makefile, then compile with make; alternatively, refer to the RJGT102 Data Manual.
Run the example program rjgt102; if the UUID is correct, the following response will appear.

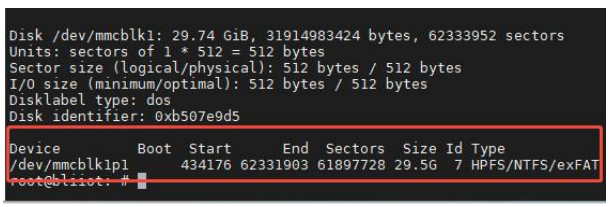
```
root@BL460:~#usr/demo/rjgt# ./rigt102

open unique file failed, create unique file!

random uuid would write rjgt102 : b6275e22-4928-4828-88fb-54a6fd8!

Contrast success

root@BL460:~#usr/demo/rjgt#./rigt102

Contrast success
```

2.15 SD Card, Solid-State Drive and USB Drive Usage

The eMMC version of the core board does not support using an SD card.
The SD card is only used as a system boot card.
If the storage format of the USB drive, SSD, or SD card is not FAT32, you need to format the USB drive, SSD, or SD card (formatting will erase all data on the device, so make sure to back up your data first).
Check the hard drive mounting status.

```
fdisk -l
```

Locate any unmounted USB drives, SSDs, and SD cards.

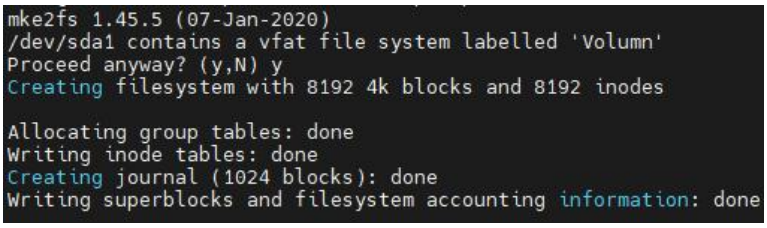


Format the USB drive, SSD, or SD card. Here, using USB drive sda1 as an example, format sda1 as

follows:

```
mkfs.ext4 /dev/sda1
```

You will be prompted to confirm the format. Enter y to start formatting.



After formatting is complete, you can normally mount the USB drive, SSD, or SD card. Use the following command to mount them:

```
mkdir /mnt/data
mount /dev/sda1 /mnt/data
```

Once mounted successfully, the USB drive, SSD, or SD card can be used normally.

After use, you need to unmount the USB drive, SSD, or SD card by entering the following command:

```
umount /mnt/data
```

If you need to mount the device permanently, follow these steps:

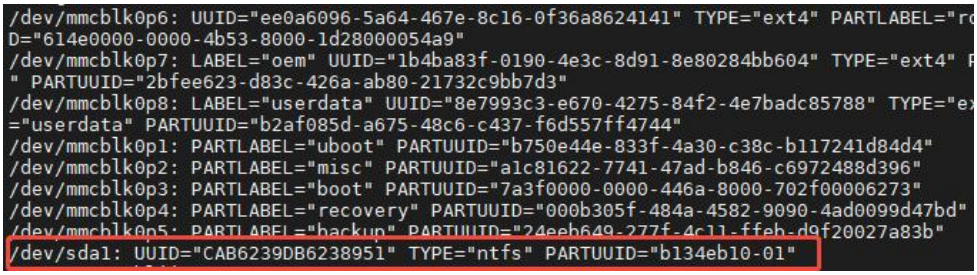
1, Identify the device: Insert the device and check it using the following commands:

sudo fdisk -l or lsblk -f

Record the device's UUID and file system type (e.g., ext4, ntfs, vfat).

2, Create a mount point: For example, to mount to /mnt/usb, run: sudo mkdir -p /mnt/usb

3, Obtain the UUID: Use the blkid command or check the output of lsblk -f to find the UUID of the corresponding device.



4, Edit /etc/fstab: sudo nano /etc/fstab and add a line at the end of the file in the following format: UUID=<device UUID> <mount point> <file system type> <mount options> <dump backup setting> <file system check order>. For example, for an NTFS USB drive: UUID=1234-5678 /mnt/usb ntfs-3g defaults,uid=1000,gid=1000,umask=022 0 0. For a commonly used Linux ext4 file system: UUID=abcd1234 /mnt/ssd ext4 defaults 0 2. For FAT32 (vfat) format: UUID=1234-5678 /mnt/sd vfat defaults,uid=1000,gid=1000,umask=000 0 0. Note: uid and gid set the user and group, and umask sets the permission mask (for example, 000 means read/write access for all users).

```
/swapfile none swap sw 0 0
UUID=1234-5678 /mnt/usb ntfs-3g defaults,uid=1000,gid=1000,umask=022 0 0
~
~
~
~
~
~
~
~
```

5. If you do not want to specify a user, you can omit options like uid, gid, and umask, and just use defaults.
6. For NTFS, you need to install ntfs-3g (usually already installed) and use ntfs-3g as the file system type, or simply write ntfs (on some systems it may be a symbolic link to ntfs-3g).
7. For exFAT, you need to install exfat-fuse and exfat-utils, and then use exfat as the file system type.
8. Test: After saving /etc/fstab, run the following command to check for errors: sudo mount -a. If no error messages appear, the mount is successful, and you can see the files at the mount point.
9. Permission settings: If the permissions after mounting do not meet your requirements, you can adjust them in the mount options or use the chown and chmod commands after mounting.

Important: If /etc/fstab is configured incorrectly, it may prevent the system from booting. Therefore, always test with sudo mount -a before rebooting. If the test fails, correct /etc/fstab according to the error messages.

For removable devices (such as USB drives or SD cards), if you want them to mount only when inserted and allow users to safely unmount them, you can use udisks2 and udev rules for automatic mounting (this is plug-and-play, not permanent).

Using /etc/fstab mounts the device at every boot, which is suitable for permanently connected devices (like internal SSDs). Therefore, for USB drives and SD cards that are not always connected, permanent mounting via /etc/fstab may not be appropriate, as the system will report an error and enter emergency mode if the device is missing at startup.

To avoid this, you can add the nofail option in the mount settings, so the system will continue to boot even if the device is not present.

Example:

UUID=1234-5678 /mnt/usb ntfs-3g defaults,nofail 0 0

With this option, the system startup will not be affected even if the device is not connected.

2.16 External RTC

This device includes an external RTC clock.

To view the external RTC device node:

```
root@BL460:~# ls /dev/rtc*
/dev/rtc     /dev/rtc0
root@BL460:~# dmesg | grep rtc0
[      4.319167] rtc-isl1208 5-006f: rtc core: registered rtc-isl1208 as rtc0
```

To view the system clock:

```
root@BL460:~# date
Tue 21 Oct 17:46:04 BST 2025
```

To set the system time:

```
root@BL460:~# sudo hwclock --set --date="2025-10-21 17:18:00"
root@BL460:~# sudo hwclock -r
```

To synchronize the system clock to the RTC:

```
root@BL460:~# sudo hwclock -s
```

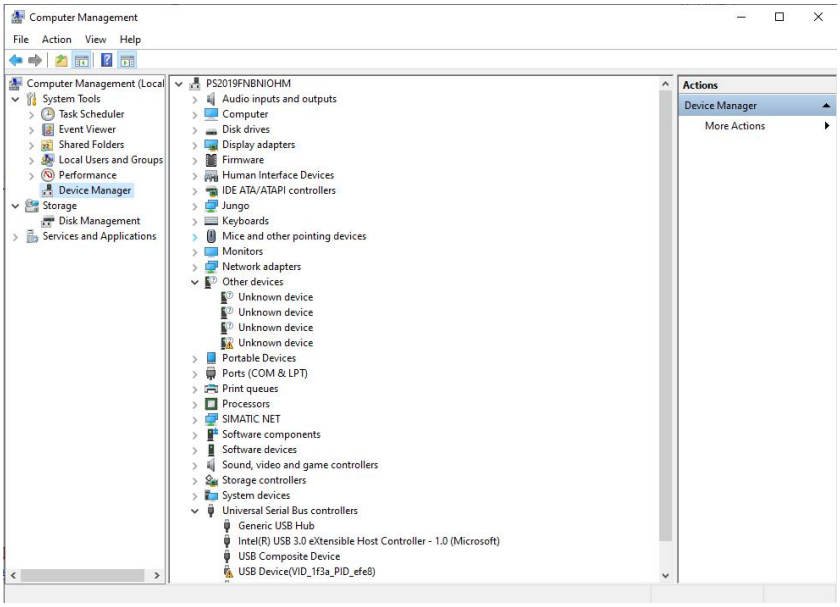
To synchronize both the system clock and the RTC

```
root@BL460:~# sudo hwclock -w
```

3 Device Login

3.1 USB Login

To access this on a computer, navigate to "This PC" → "Manage" → "Device Manager". Open the Ports section, then insert the USB cable. The refreshed port indicates the connected device port.

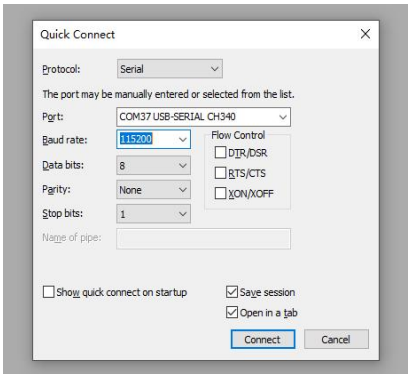


Here's an example using SecureCRT:

1. Open SecureCRT and create a new connection.
2. Choose "Serial" for the connection type.
3. Select the corresponding port.
4. Set the following parameters:
    1. Baud rate: 115200
    2. Data bits: 8
    3. Parity: None
    4. Stop bits: 1
5. Click "Connect" to access the device.

Linux systems do not have a default login password set.

Raspberry Pi default login credentials: Username: root Password: root



You can change the password of the current user account using the following command line:

```
root@BL460:~#sudo raspi-config
```

To add a new user, enter the following command:

```
root@BL460:~#sudo adduser <username>
```

To delete a user, run the following command:

```
root@BL460:~#sudo deluser -remove-home <username>
```

To create a root user, run the following command:

```
root@BL460:~#sudo su                          //Switch to the root user
root@BL460:~#sudo passwd root                 //Set the root password
root@BL460:~#sudo passwd -u root              //Enable the root user
```

3.2 SSH2 Login

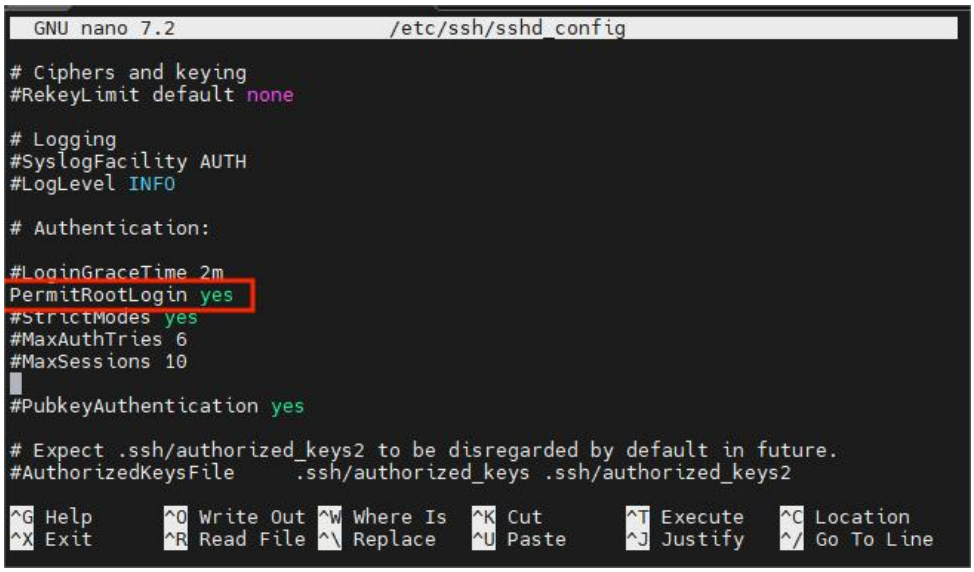
The SSH function of the BL460 may be disabled. You can enable it using the following command:

```
root@BL460:~#sudo raspi-config nonint do_ssh 0/1    //0 means enabled, 1 means disabled.
```

In the new version, SSH disables root login by default. You can modify the SSH configuration file as follows:

```
root@BL460:~#sudo nano /etc/ssh/sshd_config
```

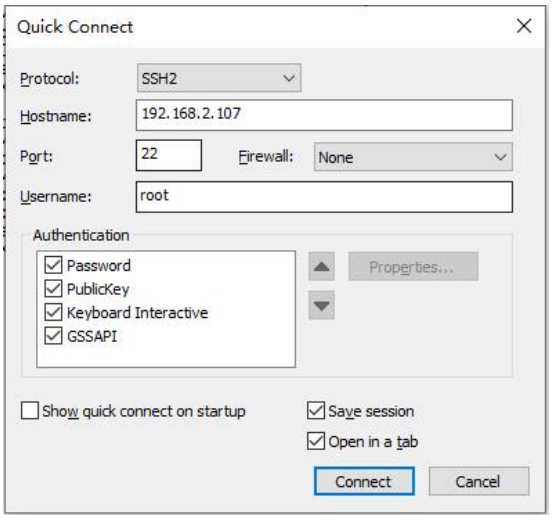
Change #PermitRootLogin without-password to PermitRootLogin yes.



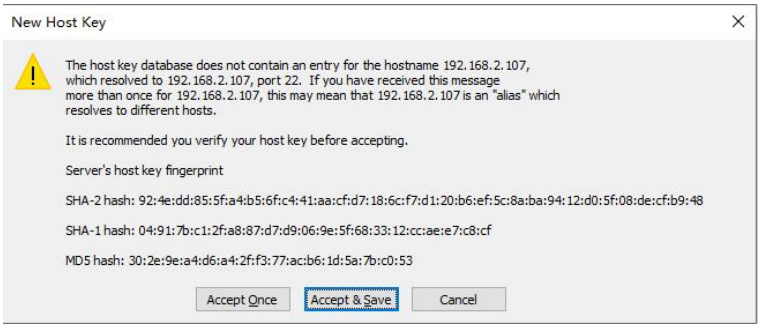
Before logging in via the Ethernet port, you need to set the IP for the corresponding port. Here, ETH2 is used as an example. ETH2 is connected to a router and has obtained the IP 192.168.2.107. The computer's IP is in the same 192.168.2.x network segment.

```
docker0   Link encap:Ethernet  HWaddr 02:42:3D:E2:6F:88
          inet addr:172.17.0.1  Bcast:172.17.255.255  Mask:255.255.0.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

eth2      Link encap:Ethernet  HWaddr 00:E0:99:CD:55:B9
          inet addr:192.168.2.107  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fd5f:4184:3ad4:4:66ee:75b9:2b9:476d/64 Scope:Global
          inet6 addr: fe80::5d2c:48eb:826c:7f6/64 Scope:Link
          inet6 addr: fd5f:4184:3ad4:4::74e/128 Scope:Global
          inet6 addr: fd2f:fd7:7cda::74e/128 Scope:Global
          inet6 addr: fd2f:fd7:7cda:0:15d3:ffef:f05a:3ebf/64 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:240 errors:0 dropped:18 overruns:0 frame:0
          TX packets:80 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:24541 (23.9 KiB)  TX bytes:8407 (8.2 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:146 errors:0 dropped:0 overruns:0 frame:0
          TX packets:146 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:10796 (10.5 KiB)  TX bytes:10796 (10.5 KiB)
```

Click Create Connection, select the protocol SSH2, enter the device IP 192.168.2.107 as the hostname, set the port to 22, and the username to root. Then click Connect to establish the connection.



Select Accept, and the connection will be successful.

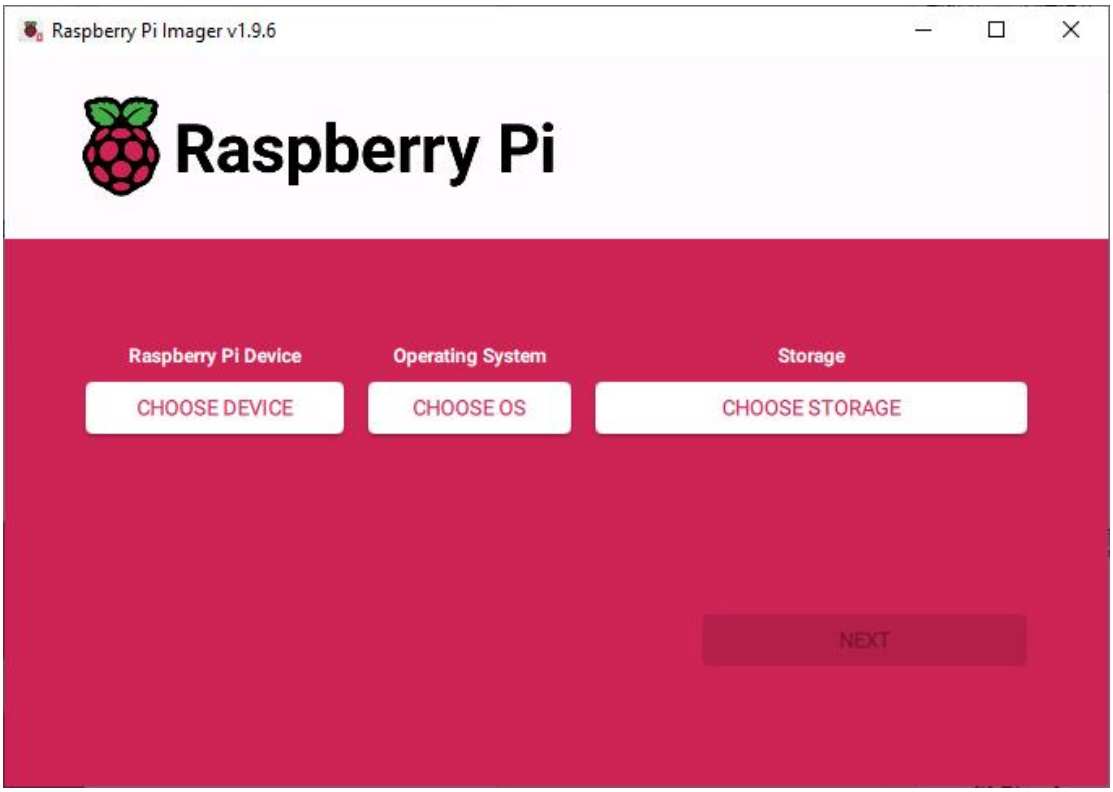


4 System Programming

4.1 Micro SD Card Boot

4.1.1 Boot Card Creation

Insert a blank Micro SD card into your computer, then download and open the flashing tool Raspberry Pi Imager.

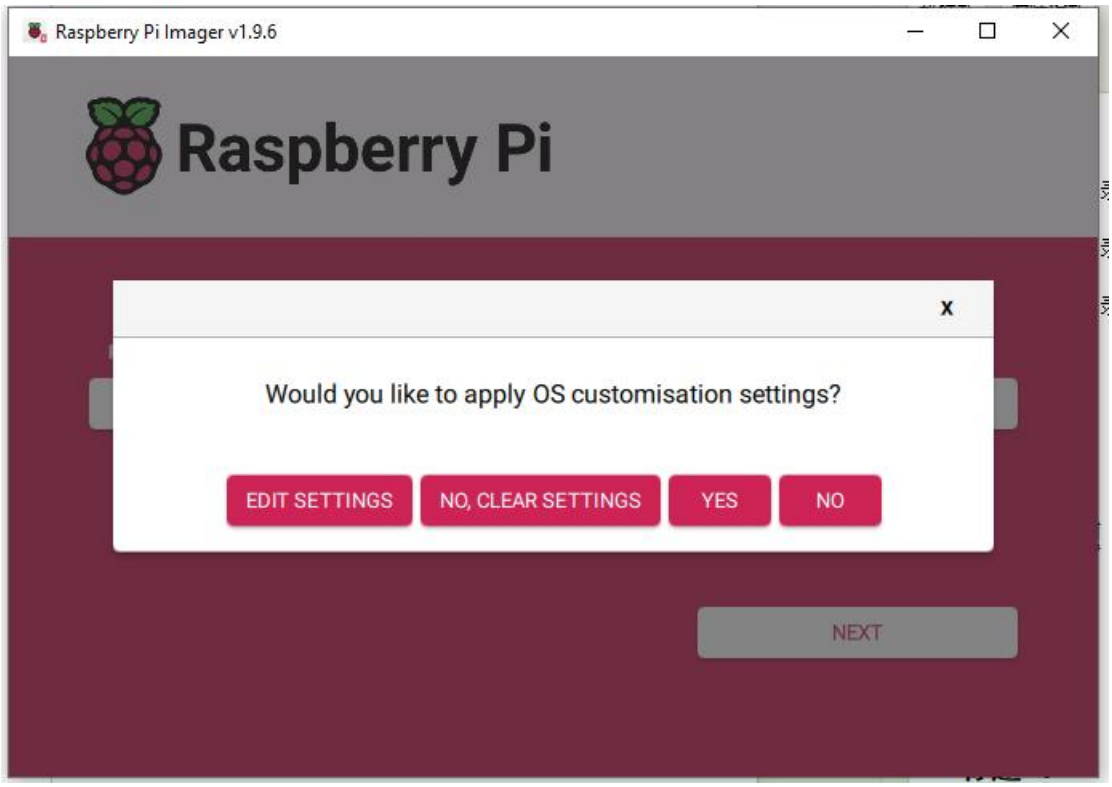


1, Select the hardware to flash; for CM5, choose PI5.

2, Select the image. If multiple versions are available, choose the one that suits you (you can also choose to format or select a previously backed-up system).

3, Select the target drive. Only removable drives (USB or SD card) are supported—choose your SD card.

4, Once everything is selected, click NEXT. If you choose Raspberry Pi OS, a window will appear. If you do not need to personalize settings, click No. If you have previously saved configurations, you can click Yes. For first-time use, if you want to set options, click Edit Settings.



5, With a brand-new system, the default is not to configure a username and password. Users can set the username and password directly in the configuration interface. If not configured, you will need to connect a keyboard and mouse after booting to set the username and password.

4.1.2 Boot from the Boot Card

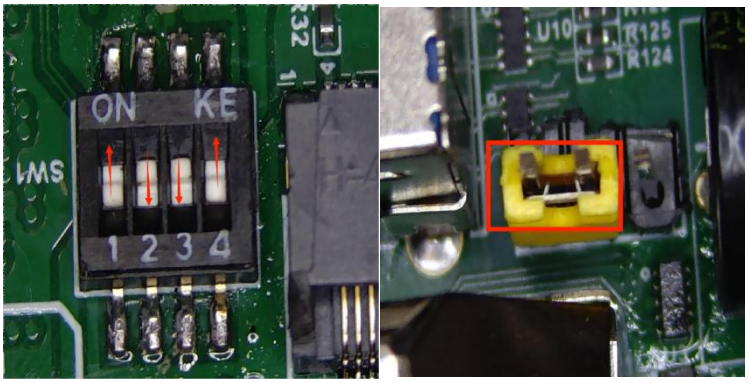
After flashing is complete, remove the Micro SD card. Insert the Micro SD card into the Micro SD slot on the baseboard, connect the CM5 core, and power it on.
Note: During the first system boot after flashing, the device may restart twice; this is normal.

4.2 EMMC Boot

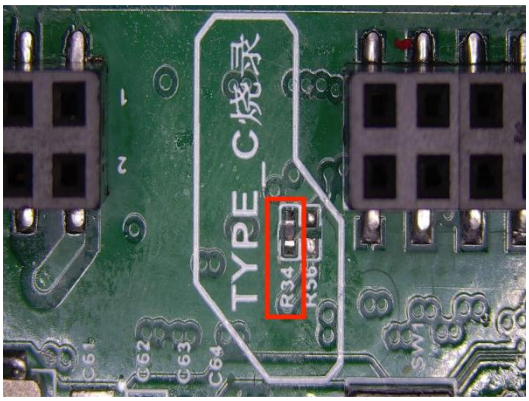
4.2.1 Programming Card Creation

1, First, remove the four screws on the top and bottom of the case, open the top cover, and then remove the upper side cover. Remove the Y board and X board (if present). When removing the X/Y boards, first unplug the pin connectors, then gently push the side cover outward to remove the Y1 board.

2, Toggle the DIP switch: set switches 1 and 4 to the ON position, and switches 2 and 3 in the opposite direction. Move the jumper caps to the positions shown in the diagram.



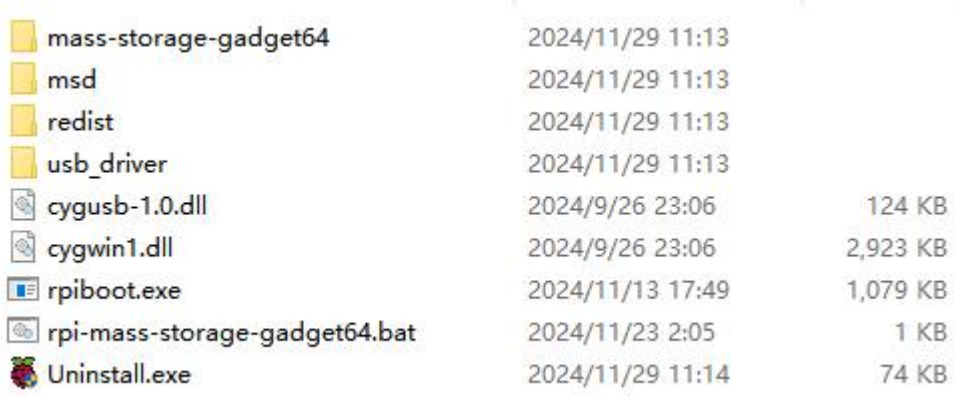
3, Disconnect the 0 Ω grounding resistor R34 on the USB_OTG port.



4, Use the Type-C port on the top of the baseboard for the flashing connection.

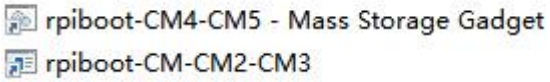
4.2.2 Recognize the core board eMMC as a removable drive.

Download and open the rpiboot software with administrator privileges. Disable any antivirus software to install the drivers and boot tool. After successful installation, rpiboot.exe will be located in the installation directory.

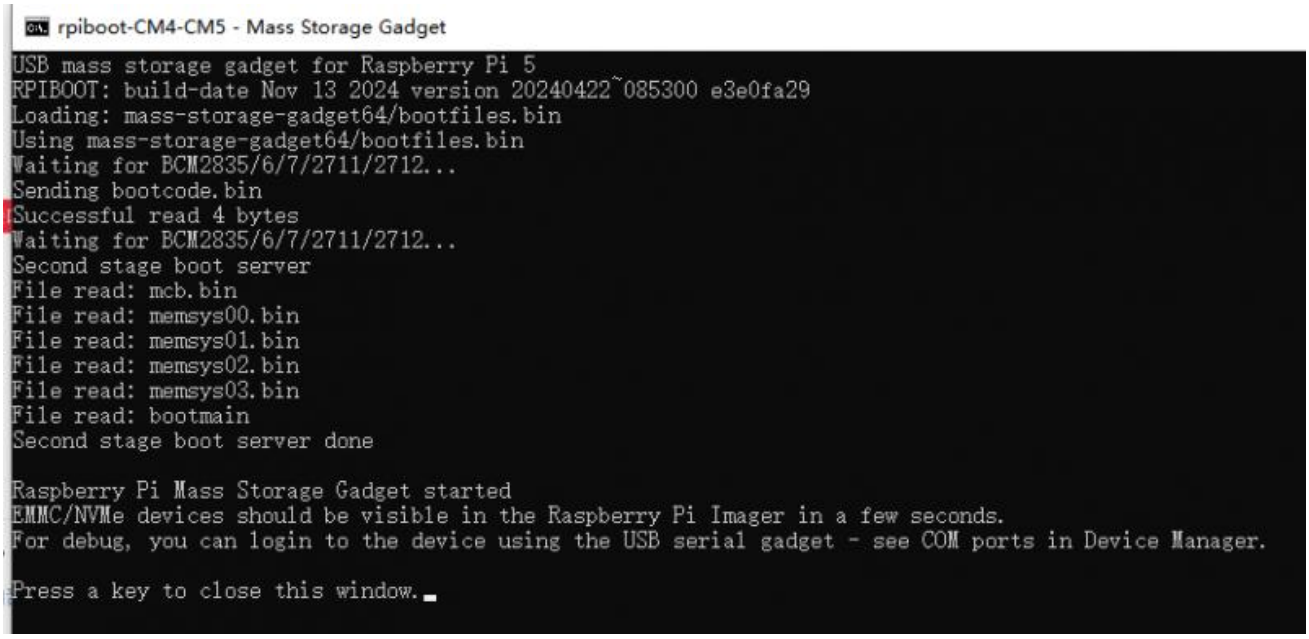


1, Connect the board to the computer via the Type-C port (SLAVE port) using a USB cable.
2, With the board powered on and connected, the computer's Device Manager will recognize a BCMxxx device (CM5 shows BCM2712, CM4 shows BCM2711). If the driver was not installed successfully, the device will appear under "Other Devices."

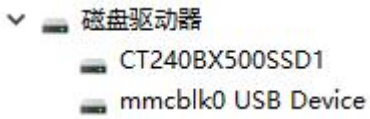
Then run rpiboot (do not run as administrator) and select the corresponding executable or script.



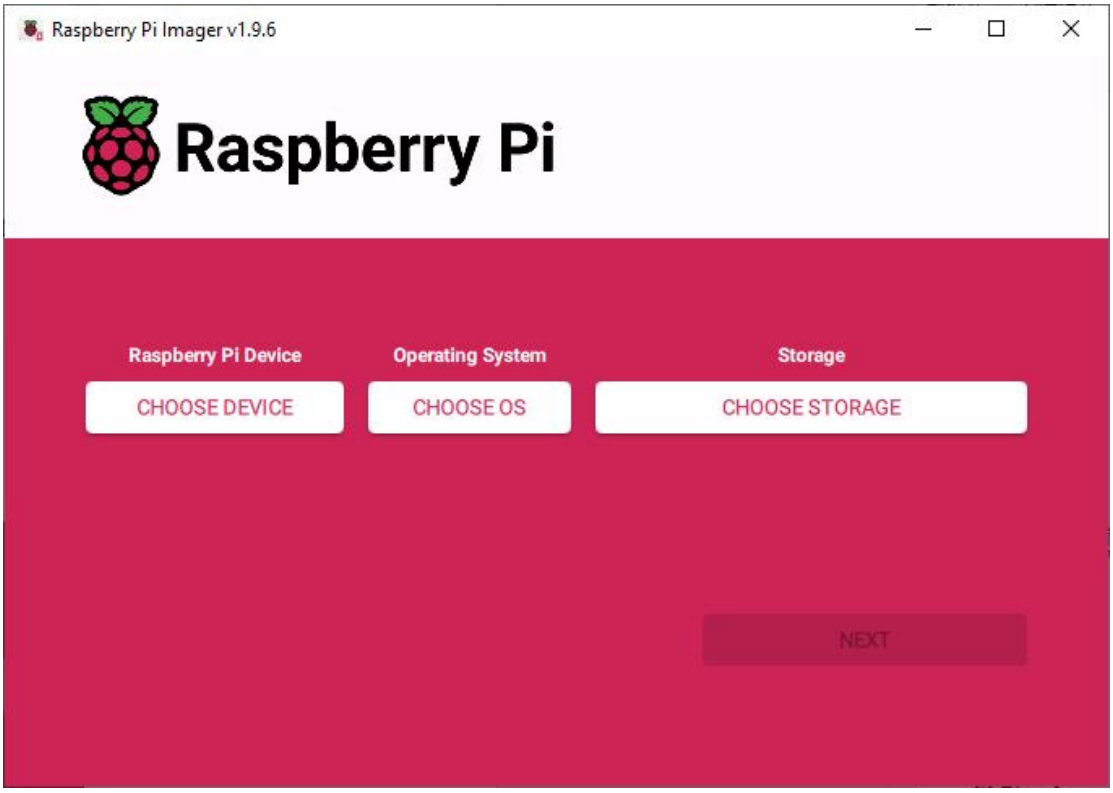
For example, if the connected board is a CM5.



3, Wait for the process to complete. A new drive, mmcblk0, will appear in My Computer.



4.2.3 System Programming

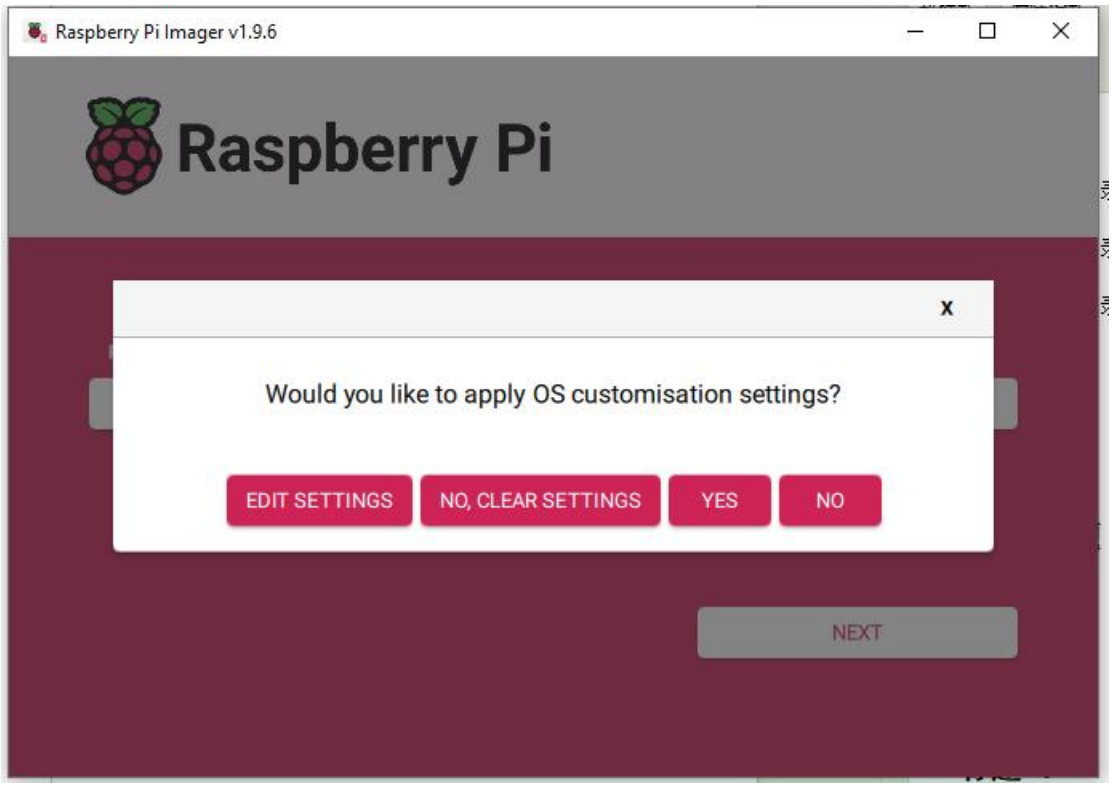


1, Open Raspberry Pi Imager and select the hardware to flash (downloaded in the previous steps); for CM5, choose PI5.

2, Select the image. If multiple versions are available, choose the one that suits you (you can also choose to format or select a previously backed-up system).

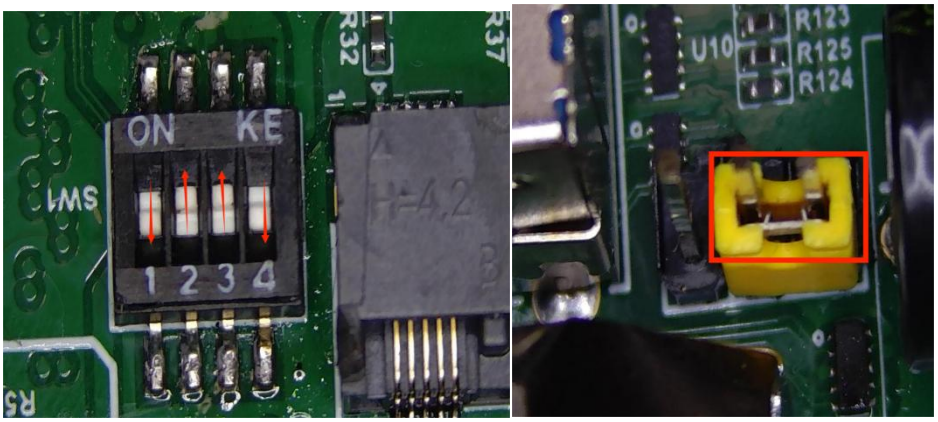
3, Select the target drive, which is the removable drive generated in step 4.2.1 (USB or SD card).

4, Once everything is selected, click NEXT. If you choose Raspberry Pi OS, a personalization configuration interface will appear. Click No if no personalization is needed, Yes if you have a previously saved configuration, or Edit Settings if it's the first time and you want to configure options.



5, After flashing is complete, power off the board and disconnect the cable from the computer.

6, Toggle the DIP switches: set switches 2 and 3 to ON, and switches 1 and 4 in the opposite direction. Then move the jumper caps to the positions shown in the diagram.



7, Reconnect the 0 Ω grounding resistor R34 on the USB_OTG port. If frequent flashing is needed, you can leave it disconnected temporarily; this only affects the host/slave recognition of USB devices.

8, Reassemble the case and power on the device.

Note: During the first system boot after flashing, the device may restart twice; this is normal.

5 Software Ecosystem

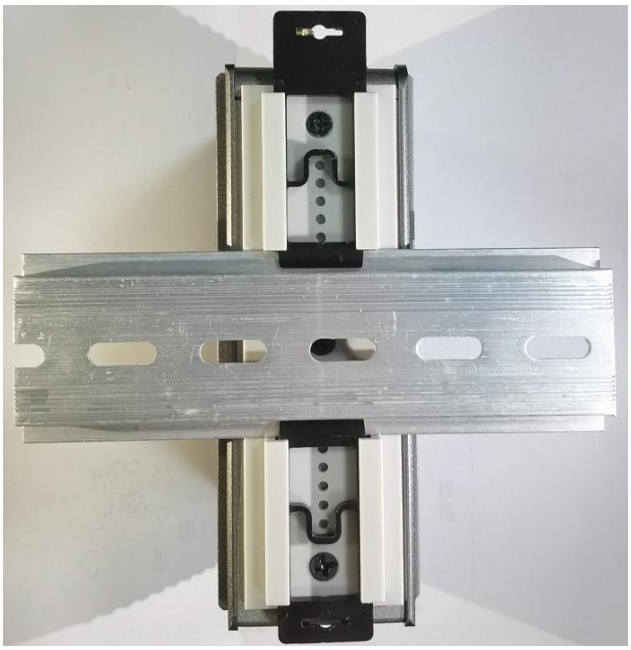
Category	Software	Type	Highlights
Industrial Communication & Protocols	IGH EtherCAT Master	Open Source	Supports real-time EtherCATmasterfor high-precision motion control and synchronized I/O.
Data Acquisition Edge Processing	BLIoTLink	Proprietary	Data acquisition and protocol conversion, supporting multiple protocols and API-based secondarydevelopment.
	Node-RED	Open Source	Low-code logic orchestration tool, supporting visual flow design and custom nodes.
	Vnode	Open Source	Lightweight edge computing node, suitable for high-efficiency data pipeline processing.
Industrial Control & Execution	OpenPLC	Open Source	Open-source PLC, suitable for simple logic control and local automation.
	CODESYS Runtime	Licensed	Industrial control platform, supporting full IEC61131-3 programming and motion control.
	Beremiz	Open Source	Open-source IEC61131-3 compliant PLC integrated development environment for machine automation, providing tools to create HMI.
	NexPLC	Proprietary	Next-generation industrial control and operation & maintenance integrated platform, supporting cloud-based collaboration.
Visualization & Monitoring	FUXA	Open Source	Lightweight web-based SCADA, suitable for rapid configuration and small to medium monitoring projects.
	Ignition	Open Source	Enterprise-level industrial platform, supporting integrated SCADA, MES, and IoT deployment.
	Grafana	Open Source	Professional time-series data visualization and analytic dashboards, supporting multiple data sources.

Communication &Middleware	Nginx/Apache	Open Source	Web portal for exposing and securely managing edge services.
AI / Machine Vision	YOLOv5/8 OpenCV	Open Source	Complete edge AI vision stack, supporting object detection and image preprocessing.
	TensorFlow Lite, PyTorch Mobile	Open Source	Lightweight AI model inference frameworks, supporting edge-side intelligent analysis.
Remote Operation & Maintenance Management	BLRAT	Proprietary	Secure remote operation&maintenance channel, supporting remote device debugging and maintenance.
	QuickConfig	Proprietary	Graphical gateway configuration and management tool, supporting one-click deployment and monitoring.
Development & Support Environment	Python, C/C++,Node.js, Java	Open Source	Multi-language development support, suitable for diverse development scenarios and performance requirements.
	Python 3, Node.js	Open Source	Provides standard runtime, supporting scripting and containerized applications.
	Docker, Kubernetes(K3s)	Open Source	Supports application containerization and cluster management, enabling micro services architecture.
	API Documentation, Deployment Guides, Sample Projects	Proprietary / Open Source	Provides comprehensive technical documentation and typical scenario examples.
System & Security	OpenSSL	Open Source	Provides communication encryption and secure tunneling to ensure data transmission security.
	iptables	Open Source	Kernel-level firewall for network protection.
	Encryption Chip Demo	Proprietary	Encapsulates SHA-256 encryption and authentication algorithms.
	Wireshark, tcpdump	Open Source	Network protocol analysis for security monitoring.
	Prometheus + Grafana	Open Source	System resource monitoring and alerting, supporting visualized operation & maintenance.

6 DIN Rail Mounting

The provided DIN rail clips are divided into two parts, upper and lower. The upper part, which is longer, is installed first, while the shorter lower part is mounted on the rail. This arrangement makes it easier to install screws.

Press the clip downward, and it can be mounted onto the DIN rail. The following is a rear view of the DIN rail installation:



7 Electromagnetic Compatibility Testing

Test	Item	Standard	Level	Condition	Result	Remarks
Electromagnetic Emission	Conducted Emission	GB/T 9254 Class A/ CISPR 32 Class A	Class A	150 kHz - 30 MHz	PASS	Complies with limits for general industrial environments
	Radiated Emission	GB/T 9254 Class A/ CISPR 32 Class A	Class A	30 MHz - 1 GHz	PASS	Complies with limits for general industrial environments

Immunity Testing	ESD	GB/T 17626.2/IEC 61000-4-2	Level III	Contact discharge: ±4 kV; Air discharge: ±8 kV	PASS	—
	Radiated RF Immunity	GB/T 17626.3/ IEC 61000-4-3	Level III	Field strength: 10 V/m, 80 MHz – 1 GHz	PASS	—
	EFT	GB/T 17626.4/ IEC 61000-4-4	Level III	Power lines: 2 kV; Signal lines: 1 kV	PASS	—
	Surge	GB/T 17626.5/ IEC 61000-4-5	Level III	Differential mode: 2 kV; Common mode: 4 kV	PASS	—
	Voltage Dips and Interruptions	GB/T 17626.11/ IEC 61000-4-11	Level III	Voltage dip: 70% for 500 ms; Complete interruption: 10 ms	PASS	—
	Power Frequency Magnetic Field Immunity	GB/T 17626.8/ IEC 61000-4-8	Level III	Test intensity: 30 A/m, 50 Hz	PASS	—

Note: If the Electrical Fast Transient (EFT) standard needs to reach Level 3, filter module must be purchased separately.

8 Appendix

X13 module ports corresponding to CM5 GPIO pins.						
Port	1	2	3	4	5	6
Name	GPIO16	GPIO27	GND	GPIO7	GPIO3	COM

X14 module ports corresponding to CM5 GPIO pins.						
Port	1	2	3	4	5	6
Name	GPIO16	GPIO27	GND	GPIO25	GPIO22	COM

X15 module ports corresponding to CM5 GPIO pins.						
Port	1	2	3	4	5	6
Name	GPIO24	GPIO23	GND	GPIO7	GPIO3	GND

20-pin X board ports corresponding to CM5 GPIO pins.										
Port	1	3	5	7	9	11	13	15	17	19
Name	GPIO22	GPIO25	GPIO27	GPIO16	POWER	GND	GPIO4	GPIO12	/	/
Port	2	4	6	8	10	12	14	16	18	20
Name	GPIO3	GPIO7	GPIO23	GPIO24	POWER	GND	GPIO5	GPIO13	/	/

9 Warranty Terms

1) This equipment will be repaired free of charge for any material or quality problems within one year from the date of purchase.
2) This one-year warranty does not cover any product failure caused by man-made damage, improper operation, etc

10 Technical Support

Shenzhen Beilai Technology Co., Ltd
Website: https://www.bliiot.com