ARMxy Embedded Computer

BLIIOT®

MAKE IIOT EASIER

BL335
User Manual

Version: V1.0

Date: 2025-05-09

Shenzhen Beilai Technology Co.,Ltd

Website: https://www.bliiot.com

Preface

Thanks for choosing BLIIOT Embedded Computer. These operating instructions contain all the information you need for operation of BL335.

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.

Revision History

Revision Date	Version	Description	Owner
2025/05/09	V1.0	Initial Release	PH

Table of Contents

1 Introduction

1.1 Overview

BL335 is an industrial-grade ARM controller that allows flexible selection of the SOM board and I/O interfaces.. This embedded computer is powered by the high-performance, cost-effective Allwinner T113-i chip and supports configurable RAM and ROM sizes based on different application needs. Paired with Beilai's custom-designed X-board and Y-board, it is capable of handling complex scenarios with ease.

BL335 is compatible with BLIoTLink industrial protocol conversion software for data collection and transformation, and can seamless integration with various mainstream IoT cloud platforms and industrial SCADA software. Users can leverage the BLRAT for remote access and maintenance of the BL335 embedded computer.

BL335 also comes with the QuickConfig tool, enabling quick configuration, system management. Additionally, it supports AI-assisted application development and "what-you-see-is-what-you-get"programming, making the creation of intelligent industrial solutions faster and more intuitive.

The BL335 series ARM embedded computer has undergone rigorous electrical performance design and high-low temperature testing to ensure stability and reliability. It is designed for DIN35 rail mounting, making it suitable for various industrial application environments.

This embedded computer is widely used in industrial IoT, photovoltaic power generation and energy storage systems, automation control, and transportation and rail applications.

1.2 Appearance

BL335 BL335A BL335B

1.3 Technical Specifications

	Parameter	Description
System	CPU	Allwinner Technology T113-i, 22nm process
	Clock Speed	1.2GHz
	DSP	1x HiFi4, 600MHz
	RISC-V	1x XuanTie C906 RISC-V (64-bit) core, with a clock speed of up to 1008MHz
	RAM	DDR3 256/512MByte, 1GByte
	Storage	eMMC 4/8G
Power	Input Voltage	DC 12～24V
	Consumption	Typical: 167mA@12V (without 4G module and Y-board) Maximum: 500mA@12V (without 4G module and Y-board)
	Reverse Polarity	Reverse Polarity Protection
Ethernet	Specification	RJ-45 ports, 1 to 2 ports: 2×100M, supporting auto MDI/MDIX.
	Protection	ESD ±6kV (contact), ±8kV (air);
SIM Card	Slot Quantity	1
	Type	Drawer-type slot, supports 1.8V/3V SIM/UIM cards (NANO)
	Protection	Built-in 15kV ESD protection
Serial port (Optional)	Quantity	1/2/4*RS232/RS485 (optional)
	Baud Rate	300bps to 115200bps (adjustable)
	Transmission Distance	400 meters (RS485, baud rate ≤ 115200 bps), 10 meters (RS232, baud rate ≤ 1200 bps)
	Data Bits	8
	Parity Bit	None, Even, Odd
	Stop Bit	1, 2
CAN Functionality (Optional)	Quantity	1/2 (optional)
	Baud Rate	10k~1Mbps
	CAN-FD	Not Support
	Multi-device Communication	Support
	Transmission Distance	400 meters (baud rate⩽100 kbps)
	Extended Frame	Not Support
	Standard Frame	Support
GPIO Channels (Optional)	Input Voltage	High level: 2.2V⩽V_{OH}＜3.3V Low level: 0V⩽V_{OL}＜1V

	Maximum Allowed Input Current	High level: 9mA Low level: 3mA
	Clamping Voltage	-0.4V~3.6V
	Fuse Blowout Condition	I_{in}>100mA/channel
USB	Quantity	1*micro USB, 2*USB 2.0 HOST
SD Card	Quantity	1
	Type	Support SD, SDHC and SDXC(UHS-I) card
Antenna	Interface	1xWi-Fi, 1x4G antenna
	Type	SMA
4G Module(Optional)	L-E	GSM/EDGE:900,1800MHz 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
5G(Optional)	Redcap version	5G NR：N1/N3/N5/N8/N28/N41/N78/N79 LTE-FDD：B1/B3/B5/B8 LTE-TD：B34/B38/B39/B40/B41
	N-CN version	NR：N1/28/41/78/79 LTE：FDD B1/3/5/8 LTE：TDD B34/38/39/40/41 WCDMA: B1/8

Wi-Fi (Optional)	Interface	PCIe
	Protocol	IEEE 802.11b/g/n
	Mode	STA, AP
	Frequency	2.4GHz
	Channels	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(Includes two programmable LED indicators)
Environment	Working	-40～85°C/-20～70°C, 5～95% RH
	Storage	-40～85°C/-20～70°C, 5 to 95% RH
Others	Housing	Aluminium housing + stainless steel
	Dimensions	110x83x46mm
	Protection Level	IP30
	Installation	DIN35 rail mounted, wall mounting
	System	Buildroot-201902, Ubuntu 20.04
Certificates	/	CE, RoHS, FCC

1.4 Model Selection

1.4.1 Main Model Selection

Model	ETH	USB	X I/O Board	Y I/O Board	Dimensions
BL335	2x100M	2	2x5PIN	X	110x83x46mm
BL335A	2x100M	2	2x5PIN	1	110x83x46mm
BL335B	2x100M	2	2x5PIN	2	110x83x46mm

1.4.2 SOM Selection

Model	MCU	Clock Speed	Kernel	DDR3	eMMC	Temperature
SOM332	T113-i	1.2GHz	2 x A7	256MByte	4GByte	Industrial grade -40~85℃
SOM333	T113-i	1.2GHz	2 x A7	512MByte	4GByte	Industrial grade -40~85℃
SOM334	T113-i	1.2GHz	2 x A7	512MByte	8GByte	Industrial grade -40~85℃
SOM335	T113-i	1.2GHz	2 x A7	1GByte	8GByte	Industrial grade -40~85℃
SOM336	T113-i	1.2GHz	2 x A7	256MByte	4GByte	Wide-temperature -20~70℃

1.4.3 X Series I/O Board Selection

Model	RS485	RS232	CAN	GPIO	DI	DO	PIN
X0	x	x	x	8	x	x	2x5PIN
X1	4	x	x	x	x	x	2x5PIN
X2	x	4	x	x	x	x	2x5PIN
X3	2	2	x	x	x	x	2x5PIN
X4	2	x	2	x	x	x	2x5PIN
X5	x	2	2	x	x	x	2x5PIN
X6	2	x	x	4	x	x	2x5PIN
X7	x	2	x	4	x	x	2x5PIN
X8	1	1	1	2	x	x	2x5PIN
X9	x	x	x	x	4	4	2x5PIN

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.

Note: When using the Y board, you need to adjust the input power voltage according to the requirements of the Y board.

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: The COM terminal is used for dry contact DI, while the GND terminal is used for wet contact DI.

2.2.1 X Board Model and Port Definitions

X1(4CH RS485)					
Port Number	2	4	6	8	10
Name	ttyAS4-A	ttyAS5-A	ttyAS2-A	ttyAS3-A	GND
Port Number	1	3	5	7	9
Name	ttyAS4-B	ttyAS5-B	ttyAS2-B	ttyAS3-B	PGND

X2(4CH RS232)					
Port Number	2	4	6	8	10
Name	ttyAS4-TX	ttyAS5-TX	ttyAS2-TX	ttyAS3-TX	GND
Port Number	1	3	5	7	9
Name	ttyAS4-RX	ttyAS5-RX	ttyAS2-RX	ttyAS3-RX	PGND

X3(2CH RS232+2CH RS485)					
Port Number	2	4	6	8	10

Name	ttyAS4-A	ttyAS5-A	ttyAS2-TX	ttyAS3-TX	GND
Port Number	1	3	5	7	9
Name	ttyAS4-B	ttyAS5-B	ttyAS2-RX	ttyAS3-RX	PGND

X4(2CH CAN+2CH RS485)					
Port Number	2	4	6	8	10
Name	CAN0+	CAN1+	ttyAS2-A	ttyAS3-A	GND
Port Number	1	3	5	7	9
Name	CAN0-	CAN1-	ttyAS2-B	ttyAS3-B	PGND

X5(2CH CAN+2CH RS232)					
Port Number	2	4	6	8	10
Name	CAN0+	CAN1+	ttyAS2-TX	ttyAS3-TX	GND
Port Number	1	3	5	7	9
Name	CAN0-	CAN1-	ttyAS2-RX	ttyAS3-RX	PGND

X6(2CH RS485+4CH GPIO)					
Port Number	2	4	6	8	10
Name	ttyAS4-A	ttyAS5-A	PB0	PB7	GND
Port Number	1	3	5	7	9
Name	ttyAS4-B	ttyAS5-B	PB1	PB6	PGND

X7(2CH RS232+4CH GPIO)					
Port Number	2	4	6	8	10
Name	ttyAS4-TX	ttyAS5-TX	PB0	PB6	GND
Port Number	1	3	5	7	9
Name	ttyAS4-RX	ttyAS5-RX	PB1	PB7	PGND

X8(1CH RS485, 1CH RS232, 1CH CAN+2CH GPIO)					
Port Number	2	4	6	8	10
Name	CAN0+	ttyAS5-A	ttyAS2-TX	PB6	GND
Port Number	1	3	5	7	9
Name	CAN0-	ttyAS5-B	ttyAS2-RX	PB7	PGND

2.2.2 Y Board Model and Port Definitions

Y01(4CH NPN Type DI+4CH NPN Type DO)										
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 the wet contact of the DO ports, while DI_COM is the common terminal for the dry contact of the DI ports.

Y02(4CH PNP Type DI+4CH PNP Type DO)										
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 the wet contact of the DO ports, and DI_COM is the common terminal for the dry contact of the DI ports.

Y11(8CH PNP Type DI)										
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 the dry contact inputs.

Y12(8CH NPN Type DI)										
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(8CH PNP Type DO)										
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 the wet contact of the DO ports.

Y22(8CH NPN Type DO)										
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 the wet contacts of the DO ports.

Y24(4CH Relay Output)										
Port Number	1	2	3	4	5	6	7	8	9	10
Name	AO1+	AO1-	AO2+	AO2-	/	/	AO3+	AO3-	AO4+	AO4-

Y31(4CH 4~20mA/0~20mA Single-Ended AI)										
Port Number	1	2	3	4	5	6	7	8	9	10
Name	AI1+	AI1-	AI2+	AI2-	GND	GND	AI3+	AI3-	AI4+	AI4-

Y33(4CH 0~5V/0~10V Single-Ended AI)										
Port Number	1	2	3	4	5	6	7	8	9	10
Name	AI1+	AI1-	AI2+	AI2-	/	/	AI3+	AI3-	AI4+	AI4-

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

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

Y41(4CH 4~20mA/0~20mA Single-Ended AO)										
Port Number	1	2	3	4	5	6	7	8	9	10
Name	AO1+	AO1-	AO2+	AO2-	/	/	AO3+	AO3-	AO4+	AO4-

Y43(4CH 0~5V/0~10V Single-Ended AO)										
Port Number	1	2	3	4	5	6	7	8	9	10
Name	AO1+	AO1-	AO2+	AO2-	/	/	AO3+	AO3-	AO4+	AO4-

Y44(4CH 0~5V/0~10V Differential AO)										
Port Number	1	2	3	4	5	6	7	8	9	10
Name	AO1+	AO1-	AO2+	AO2-	/	/	AO3+	AO3-	AO4+	AO4-

Y45(4CH -5~5V/-10~10V Single-Ended AO)										
Port Number	1	2	3	4	5	6	7	8	9	10
Name	AO1+	AO1-	AO2+	AO2-	/	/	AO3+	AO3-	AO4+	AO4-

Y46(4CH -5~5V/-10~10V Single-Ended AO)										
Port Number	1	2	3	4	5	6	7	8	9	10
Name	AO1+	AO1-	AO2+	AO2-	/	/	AO3+	AO3-	AO4+	AO4-

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

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

Y53(2CH RTD 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(2CH RTD 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(4CH TC)										
Port Number	1	2	3	4	5	6	7	8	9	10
Name	T1+	T1-	T2+	T2-	/	/	T3+	T3-	T4+	T4-

Y63										
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

Note: Once the Y63 module is used in the BL335, other Y-series boards cannot be used, and only one Y63 module can be used.

2.2.3 RS485 Usage

In the BL335, “ttyASX-A” and “ttyASX-B” represent a pair of RS485 serial port lines. When using the RS485 serial port, connect the RS485 wires to the corresponding ports. For example, the X1 module includes two ports: ttyAS4-A and ttyAS4-B, and the device file is /dev/ttyAS4. Set the baud rate to 115200, 8N1, with no parity.

```
stty -F /dev/ttyAS4 ispeed 115200 ospeed 115200 cs8
```

```
echo 12345 > /dev/ttyAS4          //Send data through RS485-1 port
cat/dev/ttyAS4                    //Wait to check the received data
```

Press "Ctrl+C" to stop

2.2.4 RS232 Usage

In the BL335, “ttyASX-RX” and “ttyASX-TX” represent a pair of RS232 serial port lines. When using the RS232 serial port, connect the RS232 wires to the corresponding ports. For example, the X2 module includes two ports: ttyAS4-RX and ttyAS4-TX, and the device file is /dev/ttyAS4. Set the baud rate to 115200, 8N1, with no parity.

```
stty -F /dev/ttyAS4 ispeed 115200 ospeed 115200 cs8
echo 12345 > /dev/ttyAS4          //Send data through RS232-1 port
cat /dev/ttyAS4                   //Wait to view the received data
```

Press "Ctrl+C" to stop

2.2.5 CAN Usage

In the BL335, “CANX+” and “CANX-” represent a pair of CAN bus lines. When using the CAN bus, connect the bus lines to the corresponding ports. For example, the X4 module includes two ports: CAN0+ and CAN0-, and the device file is /dev/ttyAS4. Set the baud rate to 1000000.

```
apt install can-utils                  //Download CAN utility tool
ip link set can0 down                  //Ensure CAN0 is disabled; otherwise, it cannot be configured
ip link set can0 type can bitrate 1000000    //Set the baud rate to 1000000
ip link set can0 up                    //Open the CAN0 port
cansend can0 111#2233                  //Send data via the CAN0 port
candump can0                           //Wait to view the data received by the CAN0 port
```

Press "Ctrl+C" to stop

2.2.6 GPIO Usage

To control the pins on the BL335, the debugfs feature is required. Therefore, run the following command to ensure it is enabled:

```
mount -t debugfs debug /proc/sys/debug
```

After installation, navigate to the '/proc/sys/debug/sunxi_pinctrl/' directory. You will find the following files there:

```
cd /proc/sys/debug/sunxi_pinctrl
data                    //Pin level status (0 for low level, 1 for high level)
dlevel                  //Pin drive level (based on the chip datasheet, ranges from 0 to 3, with 0 being the weakest and 3 the strongest)
```

```
function              //Pin function setting (based on the chip datasheet: 0 = input, 1 = output)
platform              //Current platform
pull                  //Pull-up/down configuration (0 for pull-up, 1 for pull-down)
sunxi_pin             //Specify the pin (must follow the exact name as listed in the table)
sunxi_pin_configure   //All configuration information of the pin
```

Output Configuration: Taking the X7 module as an example, if you want to use the PB0 pin as an output pin with a pull-up resistor, drive strength level 1, and output set to low level, the following configuration is required:

```
echo PB0 > sunxi_pin
echo PB0 1 > function
echo PB0 1 > pull
echo PB0 1 > dlevel
echo PB0 0 > data
```

Input Configuration: Taking the X7 module as an example, if you want to use the PB0 pin as an input pin with a pull-down resistor, the following configuration is required:

```
echo PB0 > sunxi_pin
echo PB0 0 > function
echo PB0 0 > pull
cat sunxi_pin_configure
```

2.2.7 Y Board Usage

Note: This method is not applicable to the Y63 module. For Y63 usage, please refer to section 2.2.8.

(1) Software Installation

The corresponding files are located in the /usr/demo/ioy/ directory. Please refer to the actual file names.

Run chmod +x iolib_v1.1.0_install.bin on BEILAI_IOy_T113_V1.0.8_20250321.bin, and then proceed to install the software.

```
root@bliiot:/# cd /usr/demo/ioy/
root@bliiot:/# chmod +x BEILAI_IOy_T113_V1.0.8_20250321.bin
root@bliiot:/# ./BEILAI_IOy_T113_V1.0.8_20250321.bin
Md5 verify pass!
tar: ./iolib: time stamp 2024-05-29 09:16:25 is 1716973791.042311023 s in the future
tar: ./ioy: time stamp 2024-06-25 03:05:22 is 1719284328.036465231 s in the future
tar: ./S90iolib: time stamp 2024-05-27 07:47:16 is 1716795642.036139981 s in the future
Install complete!
Restarting iolib:
Stopping iolib: stopped iolib (pid 1568)
```

```
OK
Starting iolib: OK
```

(2) Y board port usage

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

```
root@bliiot:/# 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
```

Take the DI module as an example. Short DI2 and run ioy show to check 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 value:

```
root@bliiot:/# ioy get 2004      //View by address
address 2004 value 1
root@bliiot:/# ioy get 2.5       //View using <slot>.<channel>
slot 2 channel 5 value 1
```

Take the AO module as an example. Run 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

The mode type is displayed as '4t20', which corresponds to the 4–20 mA current output described in ioy help under config mode.

Use the set command to configure the channel value:

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

```
address 4000 value 10.000000
```

(3) Port Configuration

By using the ioy help command, you can view 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 range from 4–20 mA to 0–20 mA, either of the following commands can be used:

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

Modify the corresponding minimum and maximum values of the range:

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

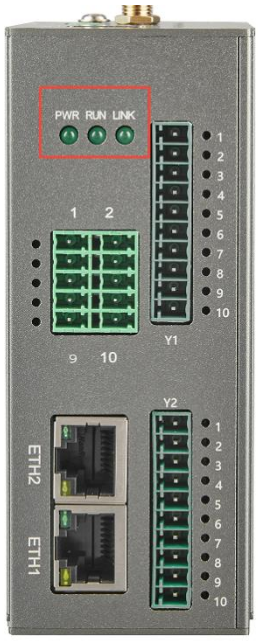
2.2.8 Y63 Module Usage

In the BL335, "ttyWCHX-A" and "ttyWCHX-B" represent a pair of RS485 serial lines. When using the RS485 interface, connect the RS485 wires to the corresponding ports. For example, the Y63 module has two ports: ttyWCH0-A and ttyWCH0-B, and the device file is /dev/ttyWCH0. Set the baud rate to 115200, 8N1, with no parity bit.

```
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 view the received data
```

Press "Ctrl+C" to stop

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 shown in the diagram. From left to right, they are LED2, LED1, and LED0.
LED2 is the POWER indicator – it stays on when the power is on and stable.
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 a wired network, and blinks when using 4G or Wi-Fi.
The control script is located at /etc/beilai_led.sh.
Check the trigger conditions with the command: cat /sys/class/leds/user-led0/trigger

```
root@bliiot:~# cat /sys/class/leds/user-led0/trigger
[none] rc-feedback mmc0 mmc1 mmc2 timer oneshot heartbeat backlight gpio cpu0 cpu1 cpu2 cpu3 default-on transient
```

[none] indicates that the current trigger condition for led0 is set to 'none'. You can change the trigger condition by writing one of the listed strings into the trigger file.

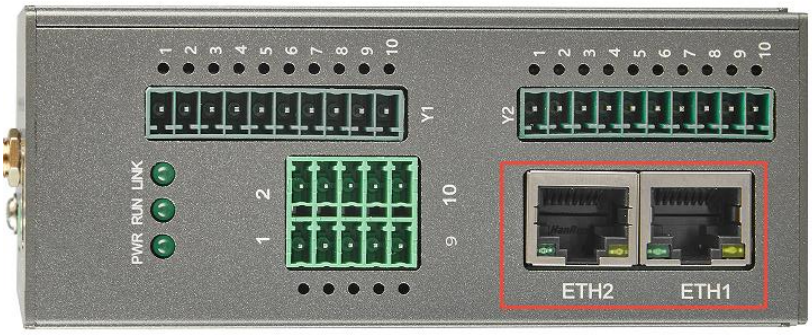
When the LED trigger condition is set to 'none', users can manually control the LED state using commands.

To turn on led0: echo 1 > /sys/class/leds/user-led0/brightness

```
root@bliiot:~# echo none >/sys/class/leds/user-led0/brightness
root@bliiot:~# echo 1 >/sys/class/leds/user-led0/brightness
```

To turn off led1: echo 0 > /sys/class/leds/user-led1/brightness

2.4 Ethernet Port



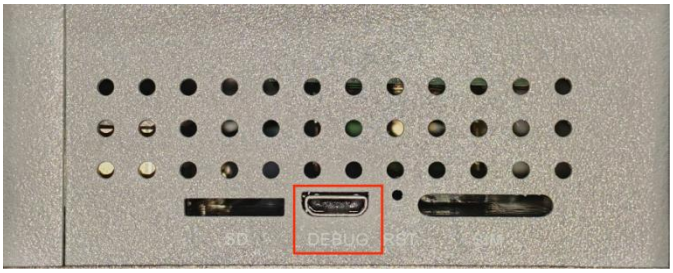
As shown in the figure, the device is equipped with two 100 Mbps Ethernet ports: ETH1 and ETH2.

2.5 USB Port

As shown in the figure, the device has two USB 2.0 HOST ports and supports USB flash drives formatted with FAT32.

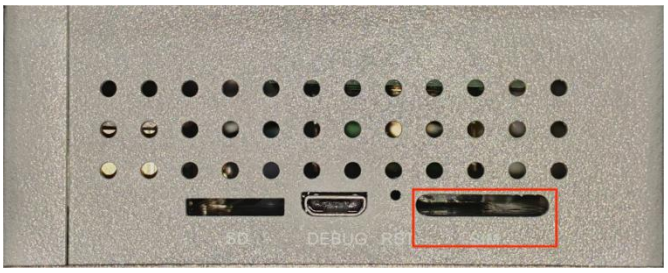
When reading from or writing to a USB drive, please use the sync command to synchronize data and prevent data loss.

2.6 Debugging Serial Port

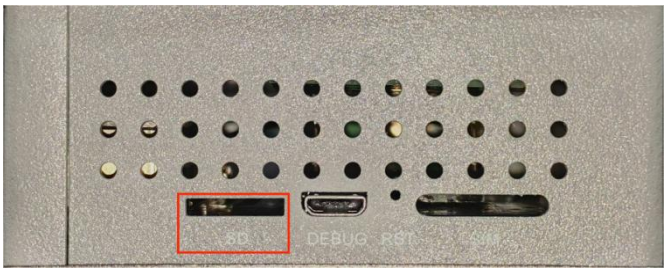


The debugging interface is as shown in the image. You can access the device's system through this port.

2.7 SIM Card Slot

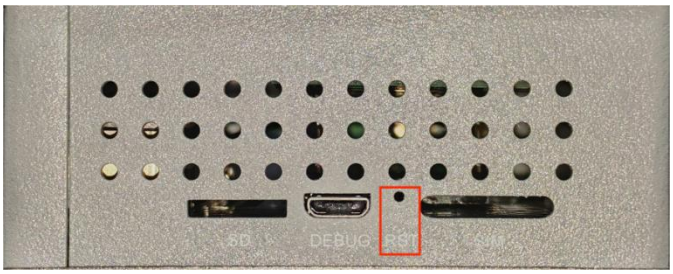


2.8 SD Card Slot



The SD card slot, as shown in the image, supports FAT32 formatted SD cards. After reading or writing data using this slot, use the sync command to ensure data is properly saved and prevent data loss.

2.9 Reset Button



Press the restart button and release it to reboot the device.

2.10 PCIe

The PCIe interface supports both 4G and WiFi.

2.10.1 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
udhcpc -i usb0
ifconfig
```

At this point, a network interface node named usb0 should be generated. If the node does not appear, it is possible that the module has not enabled the network function by default. You can try configuring the 4G module using the following command. (For EC200 series modules, the AT command port is /dev/ttyUSB1.)

```
microcom /dev/ttyUSB2
AT+QCFG="USBNET",1
```

If you are using the EC200 module, you need to add an additional command to enable network connectivity:

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

If the device returns "OK" after execution, it means the configuration was successful. This configuration only needs to be set once. After restarting the device, the usb0 node should be generated. Then, you can re-execute the network disable and enable commands. Once the usb0 node is generated, run the following command to test whether the network function is working properly.

```
ping www.baidu.com -I usb0
```

(2) SMS Functionality

To test the SMS functionality, simply run the test command in the test program directory:

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

Command description:

<device> refers to the device node of the 4G module.

<phonenumber> is the target phone number for sending the SMS.

<text> is the content of the SMS. Note that there must be no spaces between characters in the message content, otherwise an error will be prompted.

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

At this point, the corresponding number should receive an SMS with the content "test".

(3) Call Function

To test the dialing function, simply execute the test command in the test program directory:

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

Command description:

<device> refers to the device node of the 4G module.

<phonenumber> is the target phone number to dial.

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

At this point, the corresponding number should receive an incoming call from the device.

(4) GPS Function

To test the GPS function, simply execute the test command in the test program directory:

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

Command description:

<device> refers to the device node. Use the command ls /dev/ttyUSB* to check the correct node, as it may change after the device is restarted.

<timeout> is the time to wait for the latitude and longitude information to be returned (in seconds).

For example: ./get_location /dev/ttyUSB2 1

It may take a few minutes to acquire the latitude and longitude. If acquisition fails or times out, please check whether the antenna is properly connected and make sure the test is conducted in an open area.

2.10.2 Wi-Fi Module

The Wi-Fi module used here is the BL-R8188EU2 (2.4G frequency band). The test program and drivers are located in the /usr/demo/wifi directory. Make sure to connect the antenna properly. If the wlan0 network interface is not available, you can install the driver by following the steps below.

(1) STA Function

Enter the test program directory, disable other networks, keep only the Wi-Fi network, and load the Wi-Fi driver.

```
ifconfig eth1 down
ifconfig eth2 down
```

```
insmod -f 8188eu.ko                 //Load WiFi driver
ifconfig wlan0 up                   //Based on the name of the network card shown in ifconfig
```

Execute the following command to connect the device to the specified Wi-Fi network. Use -i followed by the Wi-Fi name and -p followed by the Wi-Fi password.

```
./wifi_setup.sh -i bliiot -p bebetter
```

You can check the obtained IP address using ifconfig. Then, execute the following command to test if the network functionality is working correctly.

```
ping www.baidu.com
```

(2) AP Function

After restarting the system, enter the directory where the test program is located, disable other networks, keep only the Wi-Fi network, and load the Wi-Fi driver.

```
ifconfig eth1 down
ifconfig eth2 down
insmod -f 8188eu.ko
Ifconfig wlan0 up
```

Execute the following command to set the Wi-Fi module to AP mode:

```
./ap_setup.sh
```

The default Wi-Fi name is rtl8188eu and the password is 88888888. You can modify these settings in the rtl_hostapd_2G.conf configuration file.

2.11 Antenna Interface

The antenna interface includes one Wi-Fi/4G antenna interface and one GPS antenna interface.

2.12 Hardware Watchdog

Note: The hardware watchdog is enabled by default.
The hardware watchdog timeout is 30ms. If you need to disable the hardware watchdog, you must unload the bl330_watchdog.ko module:

```
root@bliiot: rmmod -f bl330_watchdog.ko
```

If you need to re-enable the hardware watchdog, go to the /usr/demo/watchdog/ directory and reload the bl330_watchdog.ko module:

```
root@bliiot:cd /usr/demo/watchdog/
root@bliiot: insmod -f bl330_watchdog.ko
```

2.13 Encryption Chip

The encryption chip model is RJGT102. It is based on the SHA-256 encryption 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 low-power mode.
The demo for using the encryption chip in the device works by writing the value of /proc/sys/kernel/random/uuid into the encryption chip, while also saving the UUID to /usr/rjgt_unique.json. During usage, the system reads the data from the encryption chip for comparison. If the external data matches the data stored inside the encryption chip, the encryption verification is successful.

Before use, please modify the cross-compiler path in the Makefile as needed, then compile using make; or refer to the RJGT102 Datasheet for details.

When running the sample program rigt102, if the UUID is correct, the following response will appear.

```
root@bliiot: ./rigt102
open unique file failed, create unique file!
random uuid would write rjgt102 : b6275e22-4928-4828-88fb-54a6fd8!
Contrast success
root@bliiot:./rigt102
Contrast success
```

2.14 SD Card and USB Drive Usage

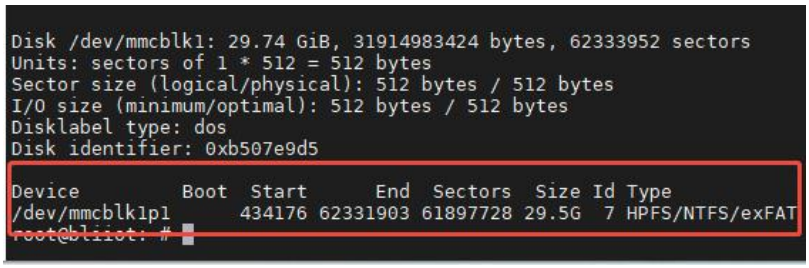
If the USB drive or SD card is not formatted in FAT32, it needs to be reformatted (note: formatting will

erase all data on the USB drive or SD card, so please back up your data in advance).

Check the disk mount status:

```
fdisk -l
```

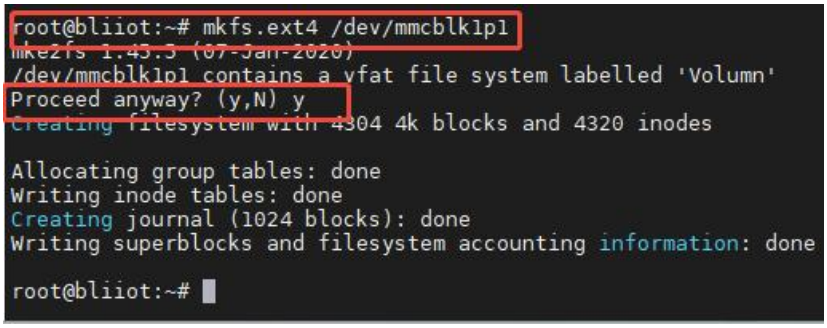
To find an unmounted USB drive or SD card



To format a USB drive or SD card, take the SD card partition mmcblk1p1 as an example. Use the following command to format it:

```
mkfs.ext4 /dev/mmcblk1p1
```

When prompted to confirm the formatting, type y and press Enter to start the formatting process.



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

```
mkdir /mnt/data
mount /dev/mmcblk1p1 /mnt/data
```

After a successful mount, you can use the USB drive or SD card normally.

Once you're done using it, you need to unmount it using the following command:

```
umount /mnt/data
```

2.15 External RTC

This device includes an external RTC clock.

To view the external RTC device node:

```
root@bliiot:~# ls /dev/rtc*
/dev/rtc    /dev/rtc0
```

```
root@BL335:~# dmesg | grep rtc0
[       4.319167] rtc-isl1208 5-006f: rtc core: registered rtc-isl1208 as rtc0
```

To view the system clock:

```
root@bliiot:~# date
Tue Jul 05 01:22:15 UTC 2024
```

To set the system time:

```
root@bliiot:~# date -s "2024-7-05 09:24:00" && hwclock -w -f /dev/rtc0
root@bliiot:~# hwclock -f /dev/rtc0
```

To synchronize the system clock to the RTC:

```
root@bliiot:~# hwclock --systohc -u
root@bliiot:~# hwclock -u
```

To synchronize both the system clock and the RTC

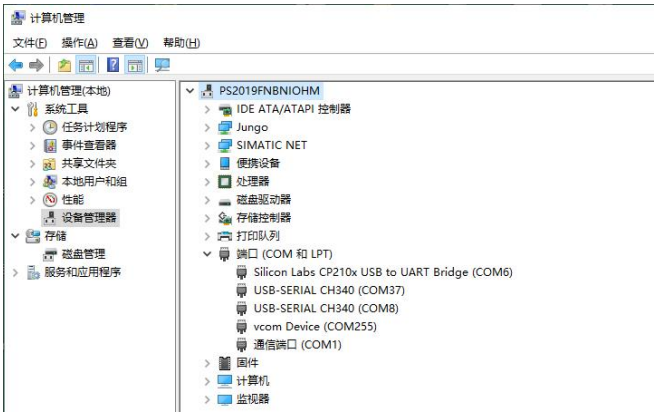
```
root@bliiot:~# hwclock --hctosys -u
```

After executing the command, the system will synchronize the RTC clock to be the system clock.

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

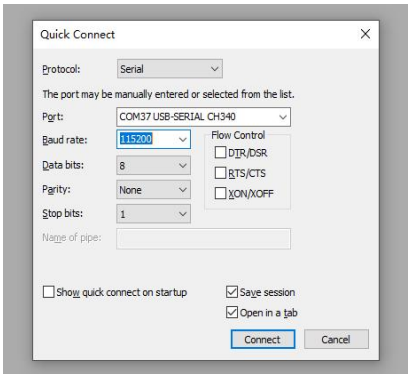
5. Click "Connect" to access the device.

Linux systems do not have a default login password set.

For Ubuntu systems:

Default Login Account: root

Password: root

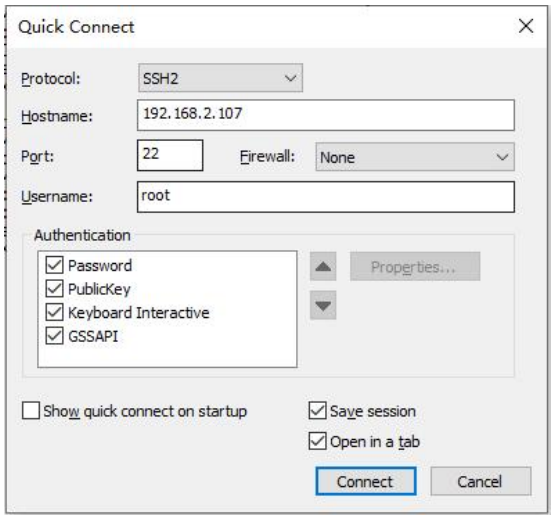


3.2 SSH2 Login

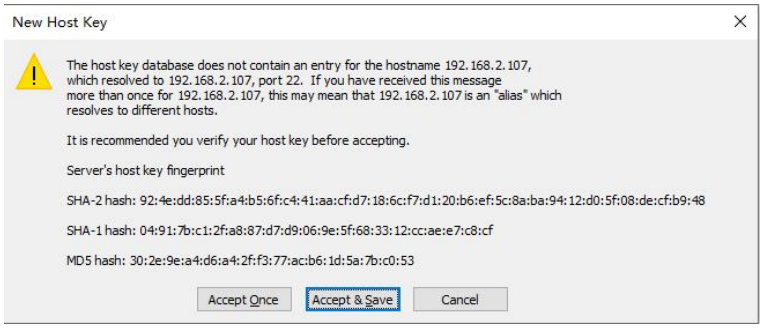
Before logging in via the network port, you need to set the IP address for the corresponding port. For example, ETH2 is connected to the router, and the obtained IP address is 192.168.2.107. The computer IP is on network segment 2.

```
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 as SSH2, enter the hostname as the device IP: 192.168.2.107, set the port to 22, and use the username root. Then, click Connect to establish the connection.



Select Accept for a successful connection.



4 System Programming

4.1 Micro SD Card Boot

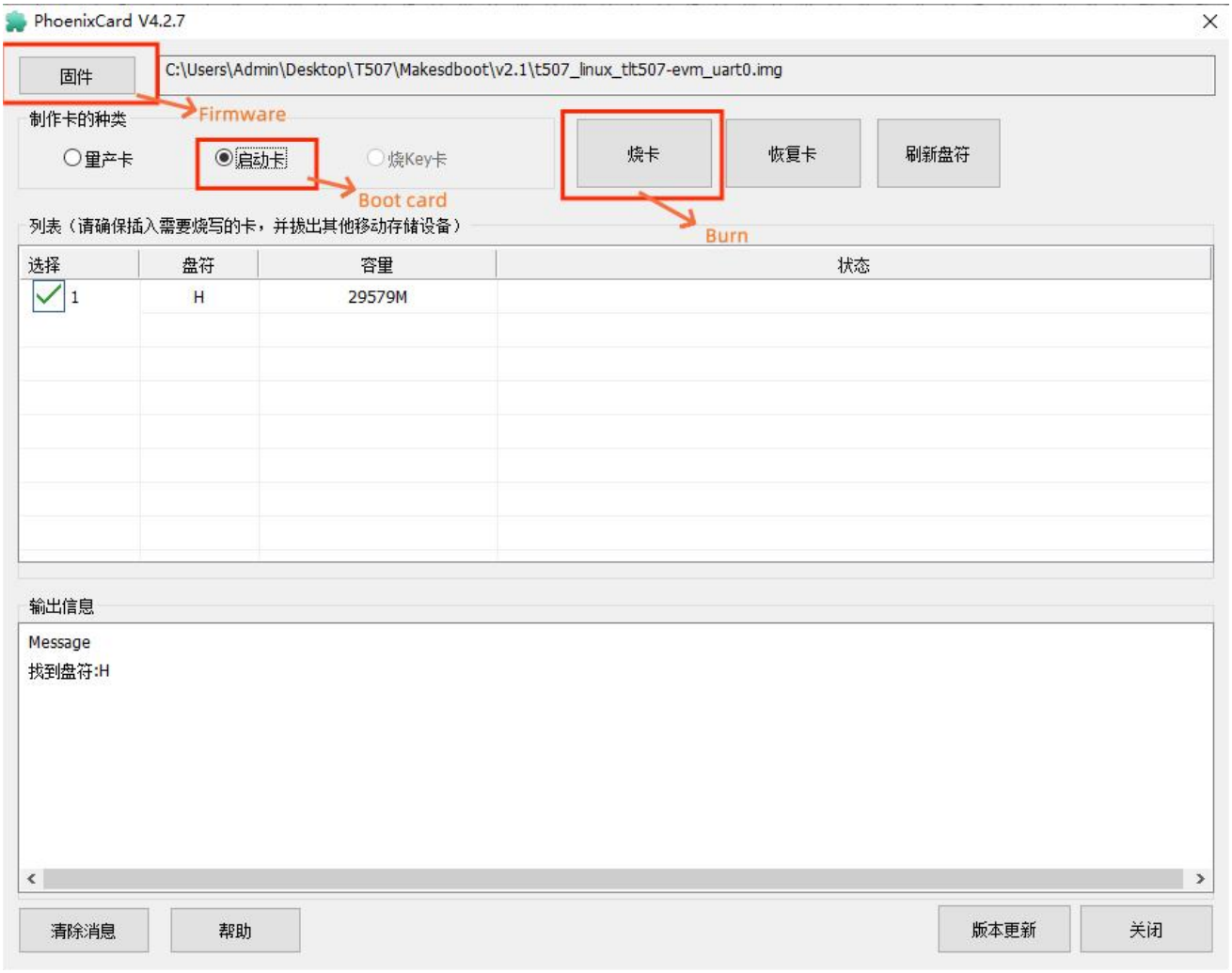
4.1.1 Boot Card Creation

To connect a blank SD card to your computer and open the system burning tool by double-clicking "PhoenixCard.exe".

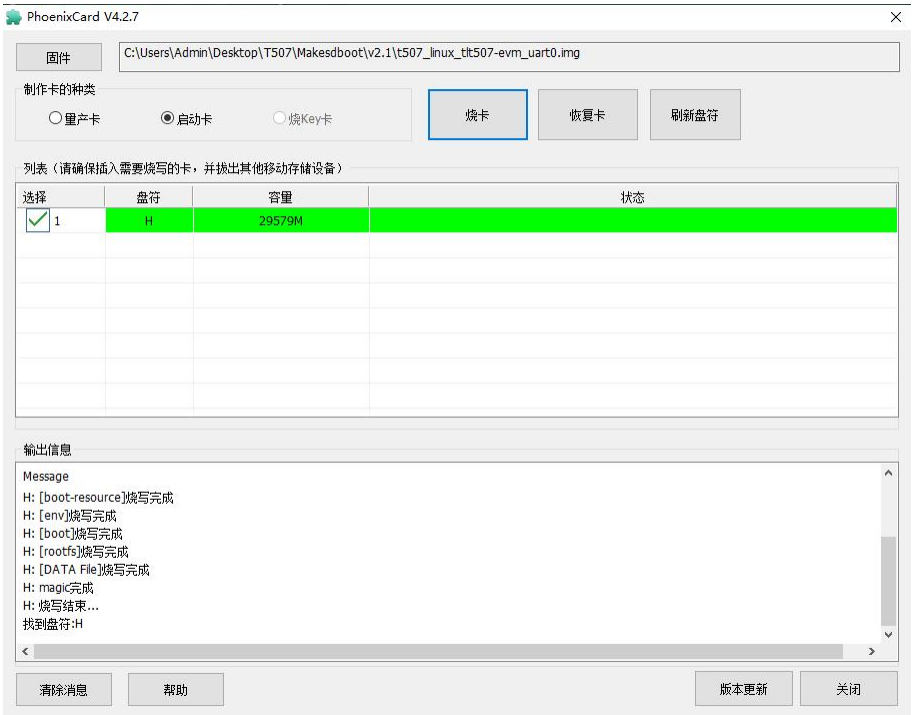
名称	修改日期	类型	大小
ini_fun.lhs	2020/06/15 10:26	LHS 文件	4 KB
IniParasPlg.dll	2020/06/15 10:26	Adobe Acrobat ...	44 KB
Langplg.dll	2020/06/15 10:26	Adobe Acrobat ...	56 KB
LangPlgex.dll	2020/06/15 10:26	Adobe Acrobat ...	24 KB
LangPlgex_1.dll	2020/06/15 10:26	Adobe Acrobat ...	24 KB
lua5.1.dll	2020/06/15 10:26	Adobe Acrobat ...	617 KB
luaBase.dll	2020/06/15 10:26	Adobe Acrobat ...	136 KB
luasocket.dll	2020/06/15 10:26	Adobe Acrobat ...	24 KB
Mbr2Gpt.dll	2021/04/21 15:22	Adobe Acrobat ...	11 KB
option.cfg	2021/07/13 19:52	CFG 文件	1 KB
ParserManager.dll	2020/07/06 18:43	Adobe Acrobat ...	81 KB
PhoenixCard.exe	2021/07/14 16:33	应用程序	1,761 KB
PhoenixCard.lan	2021/07/08 19:20	LAN 文件	4 KB
PhoenixCard_ChangeLog.pdf	2021/07/14 13:54	WPS PDF 文档	67 KB
PhoenixCard_User_Guide.pdf	2021/04/21 15:22	WPS PDF 文档	515 KB
PhoenixCard_使用指南.pdf	2021/04/21 15:22	WPS PDF 文档	4,059 KB
PhoenixCard_使用指南.pdf.10185125	2021/04/21 15:22	10185125 文件	4,059 KB
PhoenixCardFun.dll	2021/07/08 16:57	Adobe Acrobat ...	294 KB
PlgVector.dll	2020/06/15 10:26	Adobe Acrobat ...	70 KB
regbasefun.lhs	2020/06/15 10:26	LHS 文件	1 KB
regsocketfun.lhs	2020/06/15 10:26	LHS 文件	1 KB
reguifun.lhs	2020/06/15 10:26	LHS 文件	1 KB
thisdata.fex	2024/06/12 15:59	FEX 文件	14,460 KB
updatelist.cfg	2020/06/15 10:26	CFG 文件	2 KB
UpdateSelf.ulf	2020/06/15 10:26	ULF 文件	3 KB
UpdateVer.exe	2020/06/15 10:26	应用程序	1,696 KB
updatever.lan	2020/06/15 10:26	LAN 文件	2 KB
UpdateVerEx.exe	2020/06/15 10:26	应用程序	1,707 KB
version.cfg	2020/06/15 10:26	CFG 文件	1 KB
VersionTab.ulf	2020/06/15 10:26	ULF 文件	3 KB

The tool will automatically detect the Micro SD card connected to the PC, as shown in the following image.

To copy the system image to a directory with a non-Chinese path. In PhoenixCard software, click 'Firmware' to select the target image file, then choose 'Boot Card', and finally click 'Burn' to create the 'Boot Card'.



After the system boot card is created, the status bar will display green and print the following information.



4.1.2 Boot from the Boot Card

After inserting the boot card into the device's Micro SD card slot and powering on the device, the system will automatically boot from the boot card and log in as the root user. The serial debug terminal will print similar startup information.

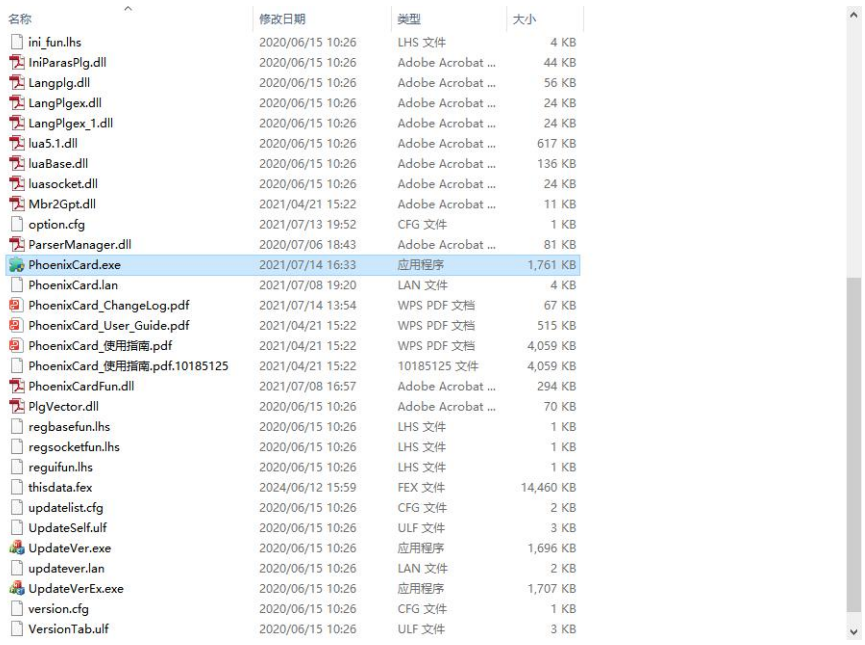
"storage type = 1" indicates Micro SD mode booting, while "storage type = 2" indicates eMMC mode booting.

```
U-Boot 2018.05-g27a1125 (Aug 07 2023 - 09:51:14 +0800) Allwinner Technology

[01.205]CPU:   Allwinner Family
[01.207]Model: TLT507-EVM HDMI
I2C:   ready
[01.212]DRAM:  1 GiB
[01.215]Relocation Offset is: 35f04000
[01.240]secure enable bit: 0
[01.242]pmu_axp152_probe pmic_bus_read fail
[01.247]PMU: AXP858
[01.262][axp][err]: can't find rtcldo_vol from table
[01.266][axp][err]: can't find rtcldo_vol from table
[01.271]CPU=1008 MHz,PLL6=600 Mhz,AHB=200 Mhz, APB1=100Mhz  MBus=400Mhz
[01.278]drv_disp_init
[01.317]drv_disp_init finish
[01.319]gic: sec monitor mode
[01.324]ir_sys_cfg: error:no node for ir boot recovery settings
[01.331]flash init start
[01.333]workmode = 0,storage type = 1
[01.337]MMC:     0
[01.338][mmc]: mmc driver ver uboot2018:2020-5-24 16:54:00-202109221409
```

4.2 EMMC Boot

4.2.1 Programming Card Creation

Connect the blank SD card to your computer, then double-click "PhoenixCard.exe" to open the system firmware tool.

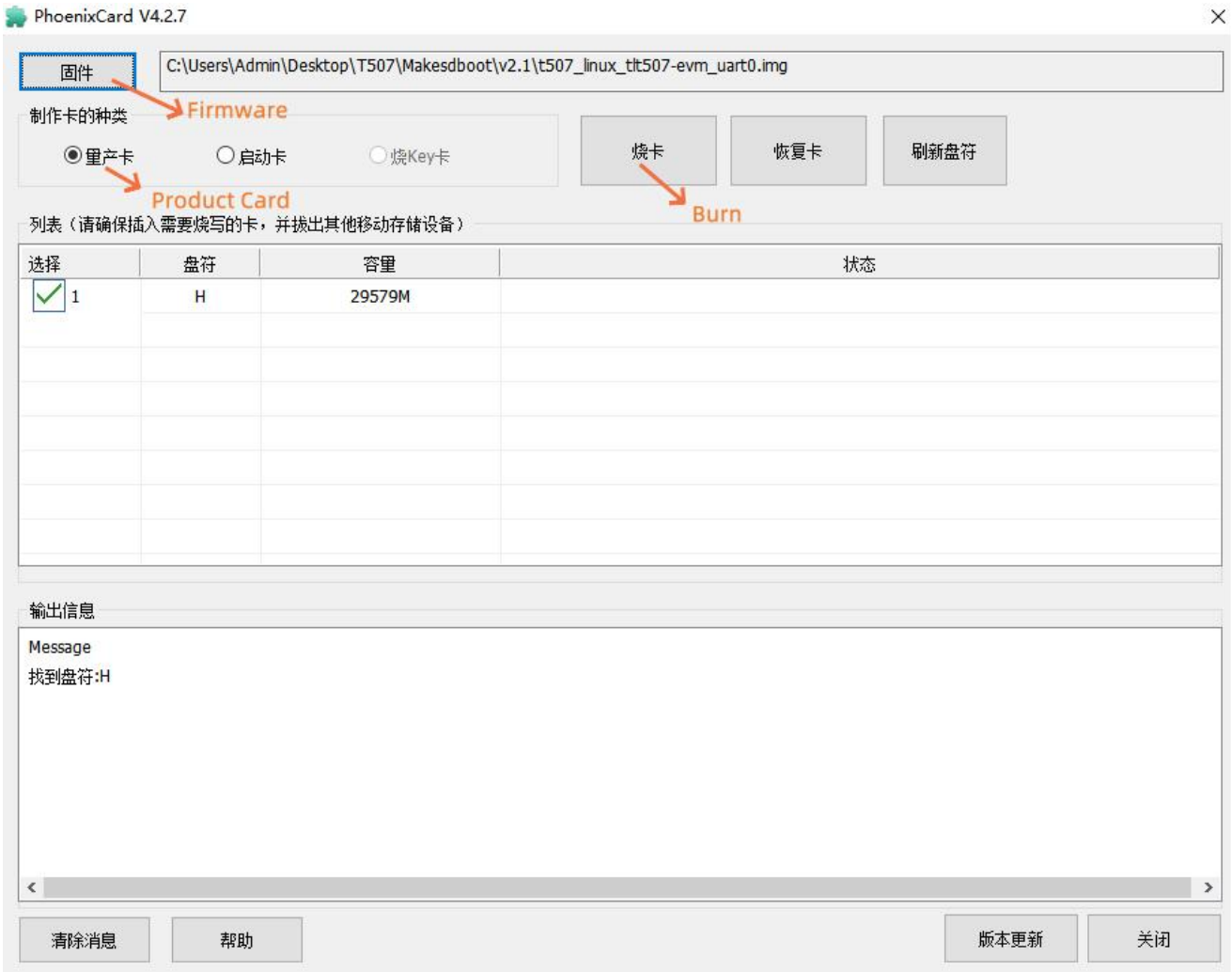


名称	修改日期	类型	大小
ini_fun.lhs	2020/06/15 10:26	LHS 文件	4 KB
IniParasPlg.dll	2020/06/15 10:26	Adobe Acrobat ...	44 KB
Langplg.dll	2020/06/15 10:26	Adobe Acrobat ...	56 KB
LangPlgex.dll	2020/06/15 10:26	Adobe Acrobat ...	24 KB
LangPlgex_1.dll	2020/06/15 10:26	Adobe Acrobat ...	24 KB
lua5.1.dll	2020/06/15 10:26	Adobe Acrobat ...	617 KB
luaBase.dll	2020/06/15 10:26	Adobe Acrobat ...	136 KB
luasocket.dll	2020/06/15 10:26	Adobe Acrobat ...	24 KB
Mbr2Gpt.dll	2021/04/21 15:22	Adobe Acrobat ...	11 KB
option.cfg	2021/07/13 19:52	CFG 文件	1 KB
ParserManager.dll	2020/07/06 18:43	Adobe Acrobat ...	81 KB
PhoenixCard.exe	2021/07/14 16:33	应用程序	1,761 KB
PhoenixCard.lan	2021/07/08 19:20	LAN 文件	4 KB
PhoenixCard_ChangeLog.pdf	2021/07/14 13:54	WPS PDF 文档	67 KB
PhoenixCard_User_Guide.pdf	2021/04/21 15:22	WPS PDF 文档	515 KB
PhoenixCard_使用指南.pdf	2021/04/21 15:22	WPS PDF 文档	4,059 KB
PhoenixCard_使用指南.pdf.10185125	2021/04/21 15:22	10185125 文件	4,059 KB
PhoenixCardFun.dll	2021/07/08 16:57	Adobe Acrobat ...	294 KB
PlgVector.dll	2020/06/15 10:26	Adobe Acrobat ...	70 KB
regbasefun.lhs	2020/06/15 10:26	LHS 文件	1 KB
regsocketfun.lhs	2020/06/15 10:26	LHS 文件	1 KB
reguifun.lhs	2020/06/15 10:26	LHS 文件	1 KB
thisdata.fex	2024/06/12 15:59	FEX 文件	14,460 KB
updatelist.cfg	2020/06/15 10:26	CFG 文件	2 KB
UpdateSelf.ulf	2020/06/15 10:26	ULF 文件	3 KB
UpdateVer.exe	2020/06/15 10:26	应用程序	1,696 KB
updatever.lan	2020/06/15 10:26	LAN 文件	2 KB
UpdateVerEx.exe	2020/06/15 10:26	应用程序	1,707 KB
version.cfg	2020/06/15 10:26	CFG 文件	1 KB
VersionTab.ulf	2020/06/15 10:26	ULF 文件	3 KB

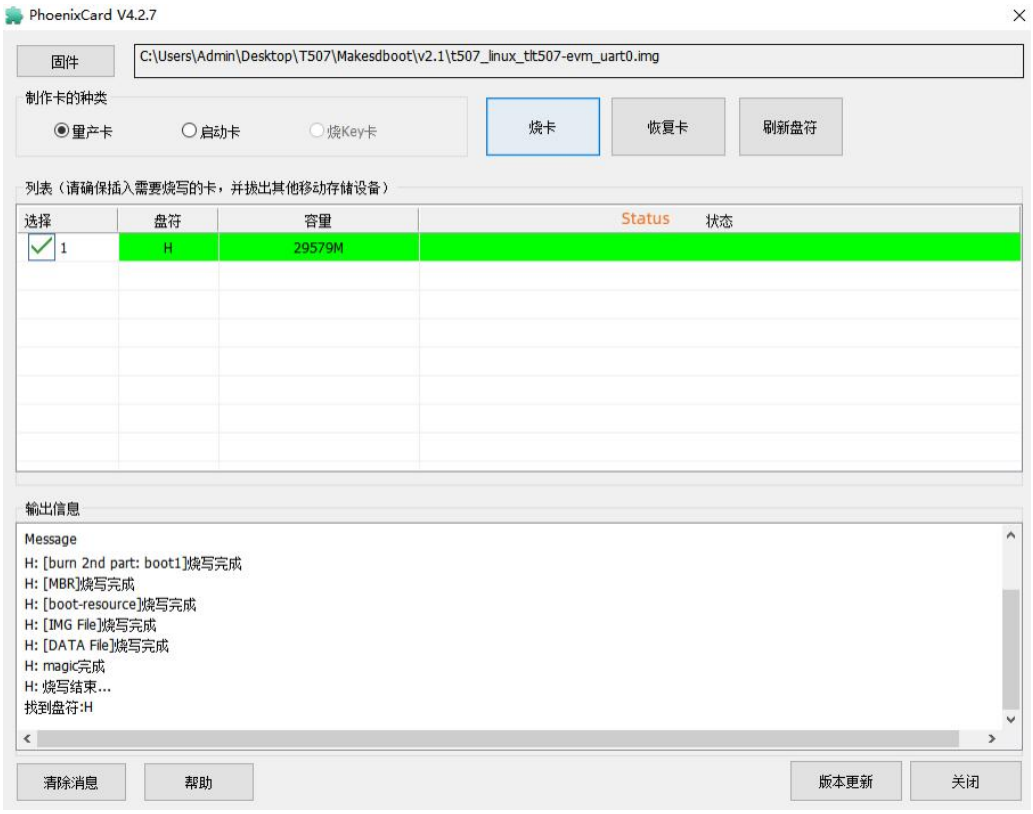
The tool will automatically detect the connected Micro SD card on the PC side.

To flash a system image onto an SD card using PhoenixCard software, follow these steps:

1. Copy the system image to a directory non-Chinese characters in the path.
2. Open PhoenixCard software and click "Firmware" to select the target image file.
3. Select "Product Card" mode.
4. Click "Burn" to create the "Product Card".



After the Product Card creation is completed, the status bar will display the following information.



4.2.2 System Programming

After inserting the prepared SD card into the device's Micro SD card slot and powering it on, the device will boot from the SD card and automatically flash the system to the eMMC. Once the system flashing is complete, the device will shut down automatically. Serial port output will be as follows.

```
dram para[3] = e0e0e0e
dram para[4] = 1919
dram para[5] = dbddaa99
dram para[6] = 60fa
dram para[7] = 4000000
dram para[8] = 520
dram para[9] = 601
dram para[10] = 8
dram para[11] = 0
dram para[12] = 0
dram para[13] = 400
dram para[14] = 813
dram para[15] = 0
dram para[16] = 0
dram para[17] = 0
dram para[18] = 0
dram para[19] = 0
dram para[20] = 0
dram para[21] = 0
dram para[22] = 0
dram para[23] = 80000000
dram para[24] = 0
dram para[25] = 0
dram para[26] = 3380b080
dram para[27] = 402fbb88
dram para[28] = 16131818
dram para[29] = 7070607
dram para[30] = 6c41
dram para[31] = 0
storage type = 2
[105.991]successed in downloading boot0
CARD OK
[105.995]sprite success
sprite_next_work=3
next work 3
SUNXI_UPDATE_NEXT_ACTION_SHUTDOWN
[109.003][mmc]: mmc exit start
[109.021][mmc]: mmc 2 exit ok
```

Please power off the device, remove the SD card, and then power it on again. The device will boot from the eMMC, and upon system startup, it will automatically log in as the root user. The serial debug terminal will print startup information similar to the following:

- "storage type = 1" indicates Micro SD mode boot.
- "storage type = 2" indicates eMMC mode boot.

4.3 System Partition

Device name	Partition	Description
Linux system boot card	/dev/mmcblk1p1	The "boot-resource" partition is used to store resources such as "bootlogo.bmp".
	/dev/mmcblk1p2	The "env" partition stores U-Boot environment variables.
	/dev/mmcblk1p3	The "boot" partition stores the kernel image.
	/dev/mmcblk1p4	The "rootfs" partition stores the root filesystem image.
	/dev/mmcblk1p5	The "private data" partition stores private data.
	/dev/mmcblk1p6	The "user partition" stores user data.
eMMC device	/dev/mmcblk0p1	The "boot-resource" partition stores resources such as "bootlogo.bmp".
	/dev/mmcblk0p2	The "env" partition stores U-Boot environment variables.
	/dev/mmcblk0p3	The "boot" partition stores the kernel image.
	/dev/mmcblk0p4	The "rootfs" partition stores the root filesystem image.
	/dev/mmcblk0p5	The "private data" partition stores private data.

	/dev/mmcblk0p6	The "user partition" stores user data.

After solidifying the Linux system onto the Micro SD or eMMC, the Micro SD or eMMC will be partitioned into 6 partitions. Additionally, a certain amount of space will be reserved in the rootfs partition for quick testing purposes.

To ensure the robustness of the file system, it's generally not recommended to perform frequent data reads and writes on the rootfs partition.

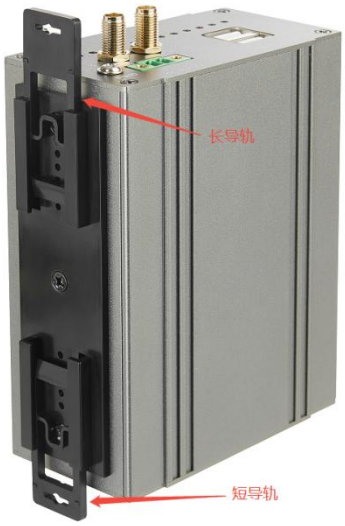
It is recommended to use the user partition "/dev/mmcblk0p6" or "/dev/mmcblk1p6" for daily data storage. Before use, please refer to the following method for formatting. This operation is demonstrated with the partition "/dev/mmcblk1p6" of the boot card of the Linux system as an example.

From the Linux system boot card, enter the device file system and execute the following commands to format the /dev/mmcblk1p6 partition and remount it. After mounting successfully, you can perform file read and write operations in the /mnt/sdcard/mmcblk1p6/ directory.

```
mkfs -t fat /dev/mmcblk1p6
mkdir -p /mnt/sdcard/mmcblk1p6/
mount /dev/mmcblk1p6 /mnt/sdcard/mmcblk1p6/
df -h
```

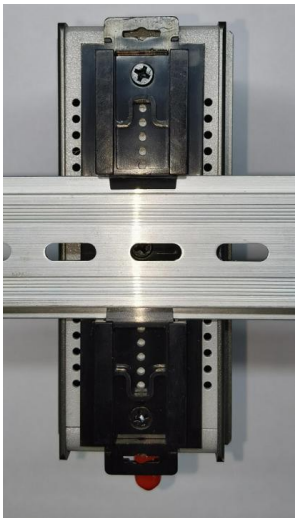
5 DIN Rail Installation

The provided DIN rail clips consist of two parts: upper and lower. The longer clip is installed on the upper part of the device, while the shorter clip is mounted on the lower part. This arrangement facilitates easier screw installation.



Press the clips all the way down to securely attach the device to the DIN rail.

Below is the rear view of the DIN rail installation:



6 Software Support

- BLIoTLink
- BLRAT
- QuickConfig
- Node-red
- Docker
- QT
- Codesys
- IgnitionSCADA
- Ubuntu 20.04

7 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

8 Technical Support

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