ARMxy Embedded Computer

BLIIOT®

MAKE IIOT EASIER

BL310
User Manual

Version: V1.0

Date: 2025-08-15

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 BL310.

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
2025/08/15	V1.0	Initial Release	PH

Table of Contents

1 Introduction

1.1 Overview

BL310 features an NXP i.MX6ULL single-core 32-bit processor with a clock speed of 800 MHz and a single-core Cortex-A7 architecture. The core board's CPU, ROM, RAM, power supply, crystal oscillator, and other components are all industrial-grade. Additionally, the core board has undergone PCB layout optimization and high/low temperature testing to ensure stability and reliability, making it suitable for various industrial applications.

BL310 offers two Ethernet ports, two USB ports, multiple RS485 communication interfaces, as well as optional Wi-Fi and 4G modules, and one power input. It can run Linux and other operating systems, and is compatible with Node-RED, Qt, Python, C++, and other applications. It also supports databases such as MySQL, InfluxDB, and SQLite. The combination of versatile hardware interfaces and strong software compatibility makes the BL310 series suitable for a wide range of industrial applications.

1.2 Appearance

BL310 BL310A BL310B

1.3 Technical Specifications

	Parameter	Description
System	CPU	NXP i.MX6ULL Cortex-A7
	Clock Speed	800MHz
	RAM	256MB/512MB DDR3L
	Storage	1G/8GB eMMC
Power	Input Voltage	DC 12～24V

	Consumption	Normal: 61mA@24V (without 4G module and Y board) 121mA@12V (without 4G module and Y board) Maximum: 164mA@9V (without 4G module and Y board)
	Reverse Polarity	Support
Ethernet	Specification	1–2×RJ45 ports, 2×100Mbps, auto MDI/MDIX.
	Protection	ESD: ±6 kV(contact), ±8 kV(air).
SIM Card	Slot Quantity	1
	Type	Drawer-type interface
Serial port (Optional)	Quantity	1/2/4*RS232/RS485 (optional)
	Baud Rate	300bps to 115200bps (adjustable)
	Data Bits	8
	Parity Bit	None, Even, Odd
	Stop Bit	1, 2
CAN Functionality (Optional)	Baud Rate	10k~1Mbps
	CAN-FD	Not Support
	Multi-device Communication	Support
	Extended Frame	Not Support
	Standard Frame	Support
GPIO Channels (Optional)	Input Voltage	High level: 2.2V$\leqslant V_{OH}<$3.3V Low level: 0V$\leqslant V_{OL}<$1V
	Maximum Allowed Input Current	High level: 9mA Low level: 3mA
	Clamping Voltage	-0.4V~3.6V
USB	Quantity	1xUSB 2.0 HOST, 1xUSB 2.0 OTG
SD Card	Quantity	1
	Type	Support SD, SDHC and SDXC(UHS-I) card
Antenna	Interface	1*Wi-Fi, 1*4G 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	LED*3(Includes two programmable LED indicators)
Environment	Working	-40～85°C/0～70°C, 5～95% RH
	Storage	-40 to 85°C, 5 to 95% RH
Others	Housing	Aluminium housing + stainless steel
	Dimensions	110x83x46mm
	Protection Level	IP30
	Installation	DIN35 rail mounted, wall mounting
	System	Buildroot-2019.02.6(Linux-4.1.15)

		Debian 10

Note: If EFT Level 3 is required, our filter module must be purchased separately.

1.4 Model Selection

1.4.1 Host Model Selection

Model	ETH	USB	X Board I/O Slot	Y Board I/O Slot	Dimensions
BL310	2x10/100M	2	2x5PIN	X	110x83x46mm
BL310A	2x10/100M	2	2x5PIN	1	110x83x46mm
BL310B	2x10/100M	2	2x5PIN	2	110x83x46mm

1.4.2 SOM Selection

ARMxy BL310 Series SOM Selection Table						
Model	**MCU**	**Clock Speed**	**Kernel**	**DDR3L**	**eMMC**	**Temperature level**
SOM311	I.MX6ULL	800MHz	Cortex-A7	256MB	1GByte	Industrial grade -40~85℃
SOM312	I.MX6ULL	800MHz	Cortex-A7	512MB	8GByte	Wide temperature grade -25℃~85℃
SMO313	I.MX6ULL	800MHz	Cortex-A7	512MB	8GByte	Industrial grade -40~85℃

1.4.3 X Series I/O Board Selection

Model	RS485	RS232	CAN	GPIO	PIN
X0	x	x	x	8	2x5PIN
X1	4	x	x	x	2x5PIN
X2	x	4	x	x	2x5PIN
X3	2	2	x	x	2x5PIN
X4	2	x	2	x	2x5PIN

X5	x	2	2	x	2x5PIN
X6	2	x	x	4	2x5PIN
X7	x	2	x	4	2x5PIN
X8	1	1	1	2	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

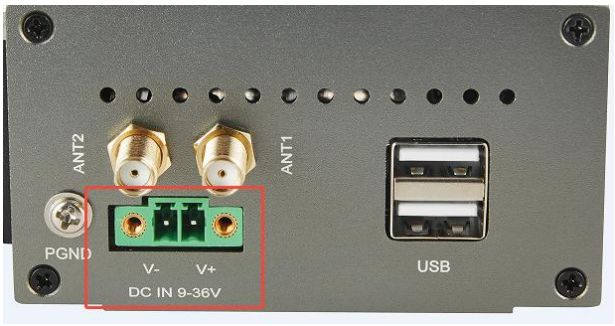
Y01: DI channels support dry contacts or NPN-type wet contact sensors.

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: Note: GND_IOS is the ground common terminal, DI_COM is the common terminal of the wet contact.

2.2.1 X Board Model and Port Definitions

X1(4CH RS485)					
Port Number	1	3	5	7	9
Name	ttymxc1-A	ttymxc2-A	ttymxc4-A	ttymxc5-A	GND
Port Number	2	4	6	8	10
Name	ttymxc1-B	ttymxc2-B	ttymxc4-B	ttymxc5-B	PGND

X2(4CH RS232)					
Port Number	1	3	5	7	9
Name	ttymxc1-TX	ttymxc2-TX	ttymxc4-TX	ttymxc5-TX	GND
Port Number	2	4	6	8	10
Name	ttymxc1-RX	ttymxc2-RX	ttymxc4-RX	ttymxc5-RX	PGND

X3(2CH RS232+2CH RS485)					
Port Number	1	3	5	7	9
Name	ttymxc1-A	ttymxc2-A	ttymxc4-TX	ttymxc5-TX	GND
Port Number	2	4	6	8	10
Name	ttymxc1-B	ttymxc2-B	ttymxc4-RX	ttymxc5-RX	PGND

X4(2CH CAN+2CH RS485)					
Port Number	1	3	5	7	9
Name	CAN0+	CAN1+	ttymxc4-A	ttymxc5-A	PGND
Port Number	2	4	6	8	10
Name	CAN0-	CAN1-	ttymxc4-B	ttymxc5-B	GND

X5(2CH CAN+2CH RS232)					
Port Number	1	3	5	7	9
Name	CAN0+	CAN1+	ttymxc4-TX	ttymxc5-TX	PGND
Port Number	2	4	6	8	10
Name	CAN0-	CAN1-	ttymxc4-RX	ttymxc5-RX	GND

X6(2CH RS485+4CH GPIO)					
Port Number	1	3	5	7	9
Name	ttymxc1-A	ttymxc2-A	gpio2	gpio4	PGND
Port Number	2	4	6	8	10
Name	ttymxc1-B	ttymxc2-B	gpio1	gpio3	GND

X7(2CH RS232+4CH GPIO)					
Port Number	1	3	5	7	9
Name	ttymxc1-TX	ttymxc2-TX	gpio2	gpio4	PGND
Port Number	2	4	6	8	10
Name	ttymxc1-RX	ttymxc2-RX	gpio1	gpio3	GND

X8(1CH RS485, 1CH RS232, 1CH CAN+2CH GPIO)					
Port Number	1	3	5	7	9
Name	CAN0+	ttymxc2-TX	ttymxc4-A	gpio1	PGND
Port Number	2	4	6	8	10

Name	CAN0-	ttymxc2-RX	ttymxc4-B	gpio2	GND

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	DI1	DI2	DI3	DI4	GND_IOS	DI_COM	DO1	DO2	DO3	DO4

Note: GND_IOS is the ground common terminal, and DI_COM is the common terminal for wet contact inputs.

Y02(4CH PNP Type DI+4CH PNP Type DO)										
Port Number	1	2	3	4	5	6	7	8	9	10
Name	DI1	DI2	DI3	DI4	GND_IOS	DI_COM	DO1	DO2	DO3	DO4

Note: GND_IOS is the ground common terminal, and DI_COM is the common terminal for wet contact inputs.

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 wet 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 wet contact inputs.

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 ground common terminal

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 ground common terminal

Y24(4CH Relay Output)										
Port Number	1	2	3	4	5	6	7	8	9	10
Name	AI1+	AI1-	AI2+	AI2-	/	/	AI3+	AI3-	AI4+	AI4-

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

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

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	GND	AI2	GND	/	/	AI3	GND	AI4	GND

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

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

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

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

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

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	ttyWC H0-A	ttyWC H0-B	ttyWC H1-A	ttyWC H1-B	GND	GND	ttyWC H2-A	ttyWC H2-B	ttyWC H3-A	ttyWC H3-B

Note: Once the Y63 module is used in the BL310, other Y-series boards cannot be used, and only one Y63 module can be used.

2.2.3 RS485 Usage

In the BL310, “ttyXXX-A” and “ttyXXX-B” represent a pair of RS485 serial port lines.
When using the RS485 interface, connect the RS485 lines to the port. For example, in the X1 module, there are two ports: ttymxc1-A and ttymxc1-B, with the device file /dev/ttymxc1.
Set the baud rate to 115200, 8N1, no parity bit.

```
stty -F /dev/ttymxc1 ispeed 115200 ospeed 115200 cs8
echo 12345 > /dev/ttymxc1          //Send data through RS485-1 port
cat /dev/ttymxc1                   //Wait to check the received data
```

Press "Ctrl+C" to stop

2.2.4 RS232 Usage

In the BL310, “ttyXXX-RX” and “ttyXXX-TX” represent a pair of RS232 serial port lines.
When using the RS232 interface, connect the RS232 lines to the port. For example, in the X2 module, there are two ports: ttymxc1-TX and ttymxc1-RX, with the device file /dev/ttymxc1.
Set the baud rate to 115200, 8N1, no parity bit.

```
stty -F /dev/ttymxc1 ispeed 115200 ospeed 115200 cs8
echo 12345 > /dev/ttymxc1          //Send data through RS232-1 port
cat /dev/ttymxc1                   //Wait to view the received data
```

Press "Ctrl+C" to stop

2.2.5 CAN Usage

Take CAN0 as an example: configure the CAN bus bitrate to 1Mbps and then start the CAN bus.

```
root@bliiot:~# ifconfig can0 down
root@bliiot:~# ip link set can0 type can bitrate 1000000
root@bliiot:~# ifconfig can0 up
```

CAN0 sending data:

```
root@bliiot:~# cansend can0 123#1122334455667788
```

CAN0 waiting to receive data:

```
root@bliiot:~# candump can0
```

If CAN0 is receiving correctly, the same data sent from another device should be received as:

```
root@bliiot:~# candump can0
  can0  123   [8]  11 22 33 44 55 66 77 88
```

2.2.6 GPIO Usage

To control the GPIO pins on the BL310, execute the following command to enter the configuration folder:

```
cd /sys/class/beilai/
```

Inside the folder, you will find the following files:

```
root@bliiot:/sys/class/beilai# ls
gpio1    gpio2    gpio3    gpio4
```

Output Configuration: Taking the X7 module as an example, if you want to use the gpio1 pin as an output, configure the pull-up resistor and set the output to low level, the configuration should be as follows:

```
echo 1 > gpio1/data
```

Input Configuration: Taking the X7 module as an example, if you want to use the gpio1 pin as an input, configure the pull-down resistor. The configuration should be as follows:

```
echo 0 > gpio1/data
```

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 file location is under the /io folder. Please use the actual file name. Execute chmod +x BEILAI_IOy_IMX6ULL_V1_20250611.bin on BEILAI_IOy_IMX6ULL_V1_20250611.bin. Then install the software.

```
root@bliiot:/# chmod +x BEILAI_IOy_IMX6ULL_V1_20250611.bin
root@bliiot:/# ./BEILAI_IOy_IMX6ULL_V1_20250611.bin
Md5 verify pass!
Restarting iolib:
Stopping iolib: stopped iolib (pid 1568)
OK
Starting iolib: OK
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 4–20mA to 0–20mA, 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.
```

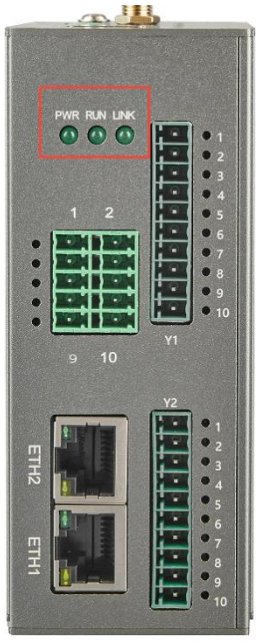
2.2.8 Y63 Module Usage

In the BL310, “ttyWCHX-A” and “ttyWCHX-B” represent a pair of RS485 serial port lines. When using the RS485 interface, connect the RS485 lines to the port. For example, in the Y63 module, there are two ports: ttyWCH0-A and ttyWCH0-B, with the device file /dev/ttyWCH0. Set the baud rate to 115200, 8N1, 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.

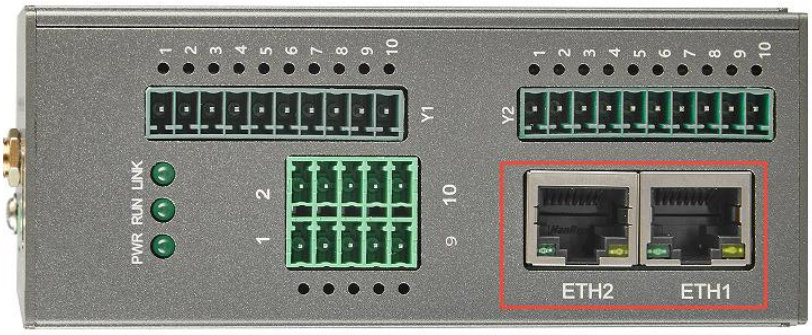
When the LED trigger condition is set to 'none', users can manually control the LED state using commands.

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



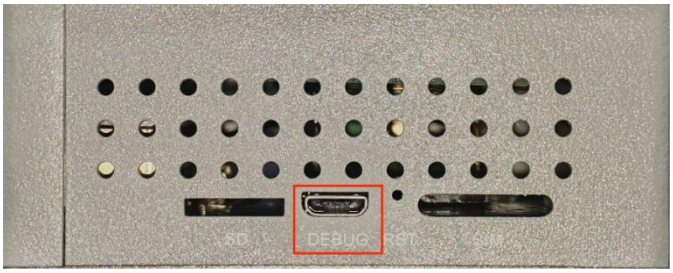
As shown in the figure, the device is equipped with two 100Mbps Ethernet ports: ETH1 and ETH2.

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.

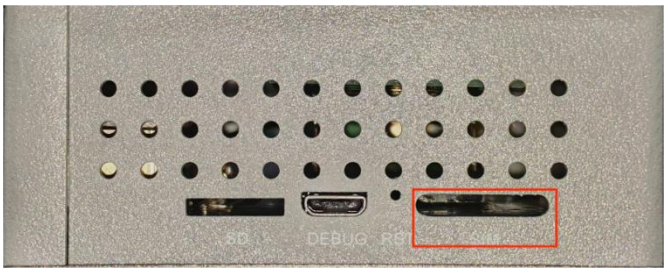
When reading from or writing to a USB drive, please use the sync command to synchronize data and prevent data loss.

2.6 Debugging Serial Port

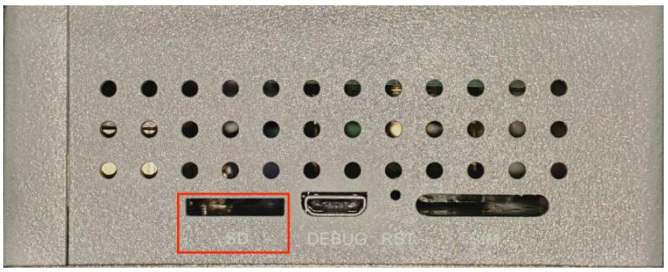


The debugging interface is as shown in the image. You can access the device's system through this port.

2.7 SIM Card Slot

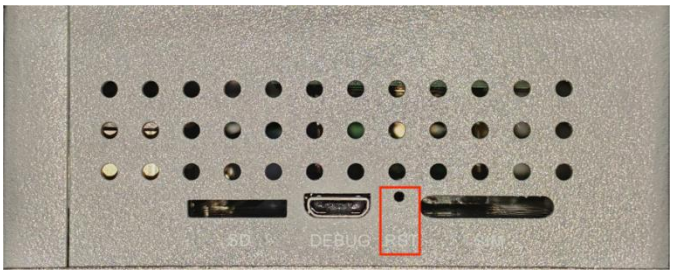


2.8 SD Card Slot



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 Wi-Fi.

2.11 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.

2.12 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.

2.13 Antenna Interface

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

2.14 Hardware Watchdog

Note: The hardware watchdog is enabled by default.
Watchdog control pin: PH4; setting it to 1 disables the hardware watchdog. The hardware watchdog timeout is 10 seconds.

2.15 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.

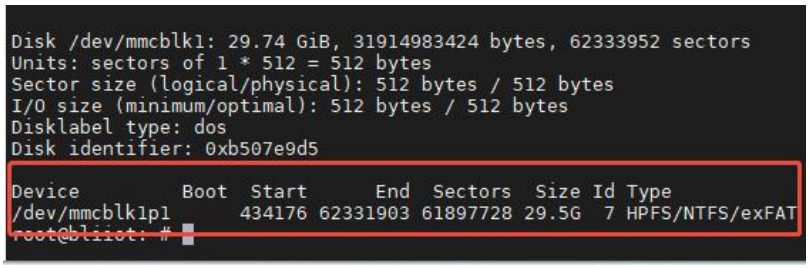
```
root@bliiot: ./rigt102
open unique file failed, create unique file!
random uuid would write rjgt102 : b6275e22-4928-4828-88fb-54a6fd8!
Contrast success
root@bliiot:./rigt102
Contrast success
```

2.16 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
```

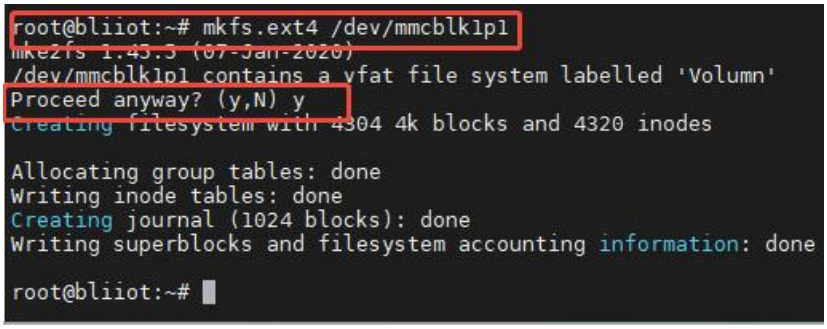
To find an unmounted USB drive or SD card



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
```

2.11 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@bliiot:~# 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

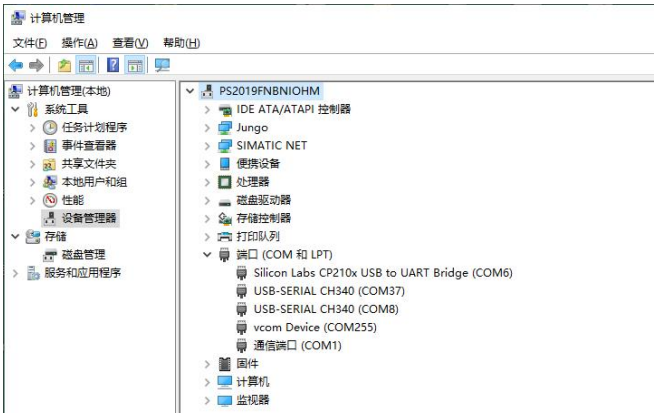
```
root@bliiot:~# hwclock --hctosys -u
```

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:

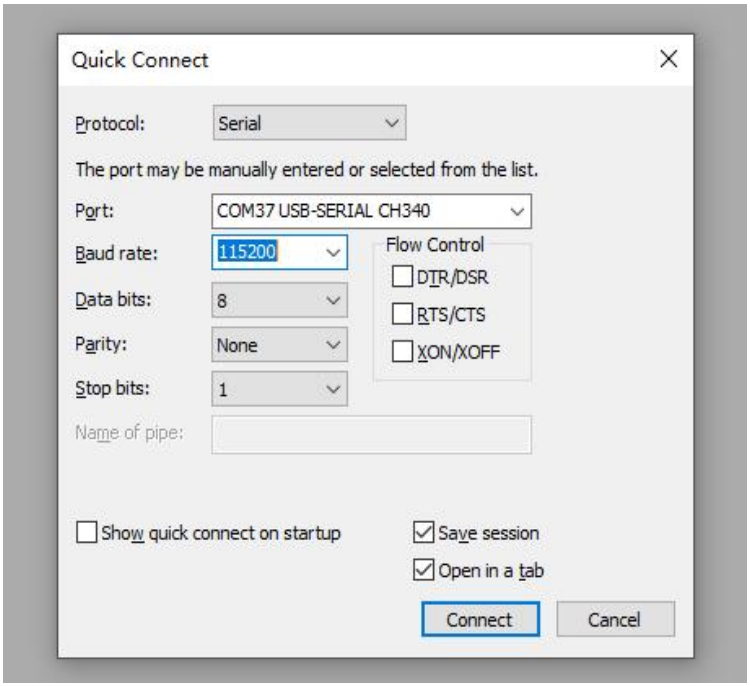
1. Open SecureCRT and create a new connection.
2. Choose "Serial" for the connection type.
3. Select the corresponding port.
4. Set the following parameters:
    1. Baud rate: 115200
    2. Data bits: 8
    3. Parity: None
    4. Stop bits: 1
5. Click "Connect" to access the device.

Linux systems do not have a default login password set.

For Debian system:

Default Login Account: root

No Password

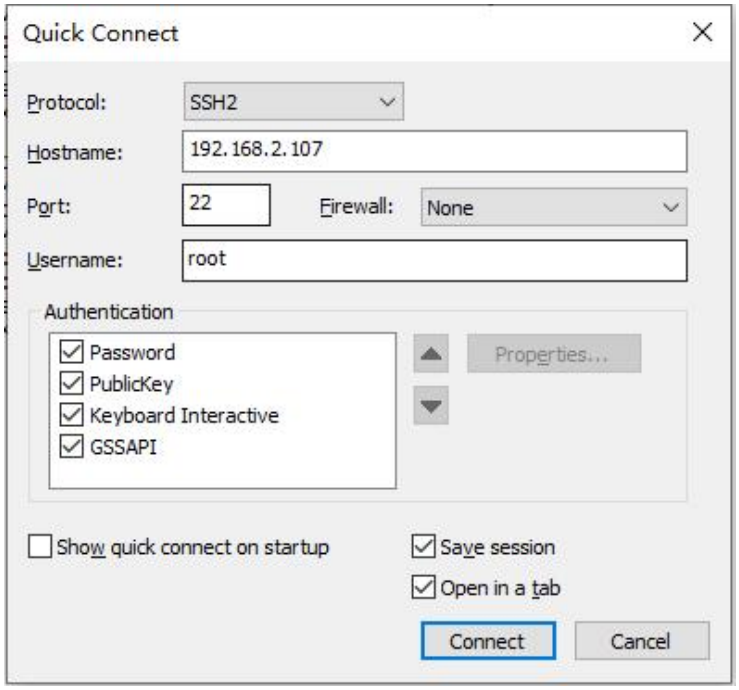


3.2 SSH2 Login

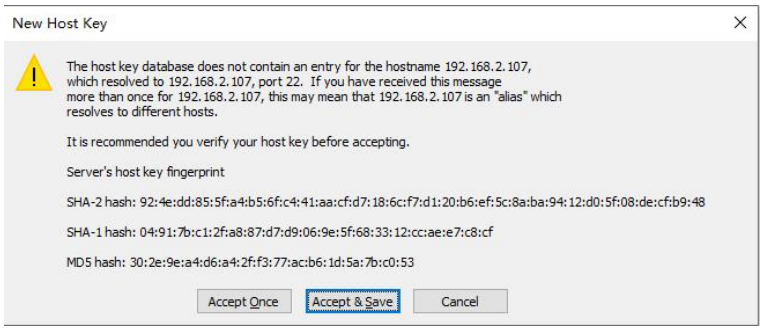
Before logging in via the network port, you need to set the IP address for the corresponding port. For example, ETH2 is connected to the router, and the obtained IP address is 192.168.2.107. The computer IP is on network segment 2.

```
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

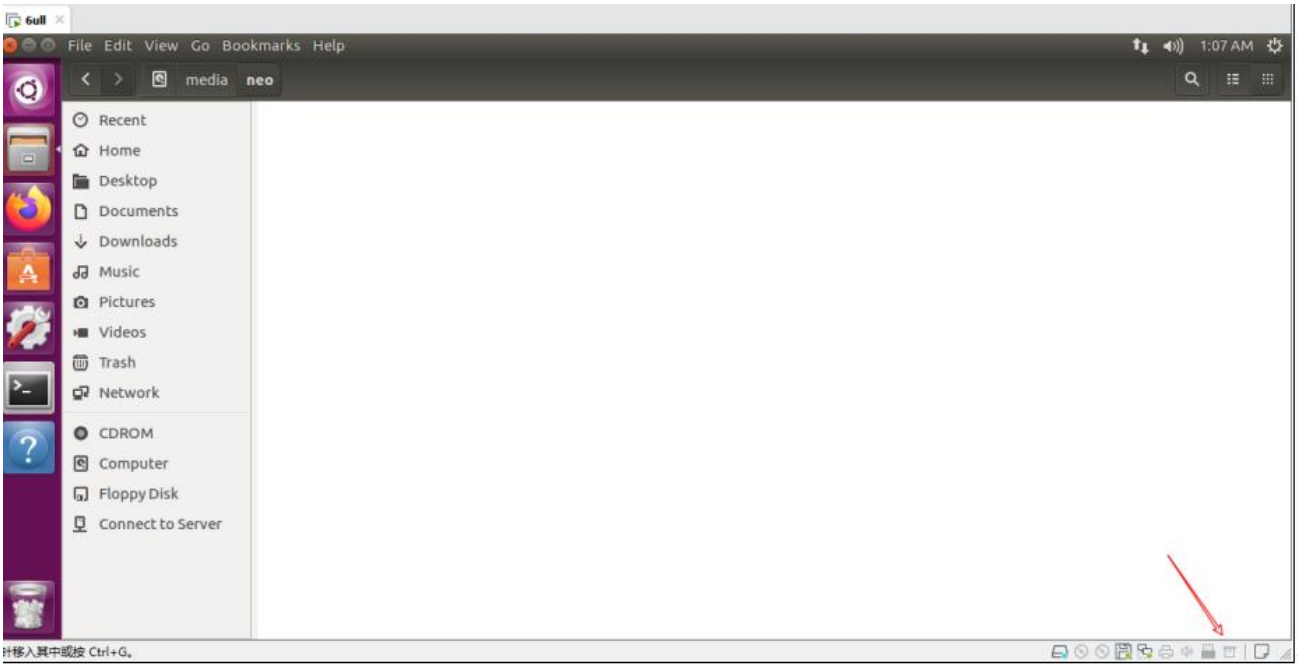
4.1 Micro SD Card Boot

4.1.1 Boot Card Creation

1, Before using the SD card, format it as FAT32 using a formatting tool.

2, After extracting emmc-burnsd.zip, copy it to any directory in the Ubuntu system, for example, /home/forlinx/work.

3, Insert the SD card into the computer's USB port using a USB card reader (for VMware virtual machine users, if the USB drive is not recognized by the VM, you can connect it to the virtual machine using the arrow-pointing icon as shown).



4, Once the virtual machine recognizes the SD card and the directory pops up, proceed with the flashing operation. Enter /home/forlinx/work/nand-burnsd and execute the script:

forlinx@ubuntu:~/work/nand-burnsd$ sudo ./burn.sh

After executing the above command, the terminal will list the computer's hard drives or USB drives. Select your SD card and press Enter. Note: You can identify your USB drive as sda/sdb/sdc based on its capacity. For example, if your USB drive is 8 GB, its size will be approximately 7761920 KB ≈ 8 GB. It is recommended not to insert multiple USB drives at the same time to avoid confusion.

Here, we take our operation as an example:

```
################################################################################
This script will create a bootable SD card from custom or pre-built binaries.
The script must be run with root permissions and from the bin directory of
the SDK
Example:
$ sudo ./6ullsdburn.sh
Formatting can be skipped if the SD card is already formatted and
partitioned properly.
################################################################################
Availible Drives to write images to:
# major minor size name
1: 8 16 7761920 sdb
Enter Device Number: 1 //Select 1 here.
sdb was selected
Checking the device is unmounted
unmounted /dev/sdb1
sdb1 sdb2 sdb3
```

```
7757824
################################################################################
Detected device has 1 partitions already
Re-partitioning will allow the choice of 1 partitions
################################################################################
Would you like to re-partition the drive anyways [y/n] : y
//Enter y, press Enter, and wait for the card creation to complete.
Now partitioning sdb ...
################################################################################
Now making 1 partitions
################################################################################
1+0 records in
1+0 records out
1024 bytes (1.0 kB, 1.0 KiB) copied, 0.0428509 s, 23.9 kB/s
DISK SIZE - 7948206080 bytes
Checking that no-one is using this disk right now ... OK
Disk /dev/sdb: 7.4 GiB, 7948206080 bytes, 15523840 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
>>> Created a new DOS disklabel with disk identifier 0x38224bb5.
Created a new partition 1 of type 'W95 FAT32 (LBA)' and of size 500 MiB.
/dev/sdb2:
New situation:
Device Boot Start End Sectors Size Id Type
/dev/sdb1 20480 1044479 1024000 500M c W95 FAT32 (LBA)
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
################################################################################
Partitioning Boot
################################################################################
mkfs.fat 3.0.28 (2015-05-16)
mkfs.fat: warning - lowercase labels might not work properly with DOS or Windows
Mount the partitions
Emptying partitions
################################################################################
Copying files now... will take minutes
################################################################################
Copying boot partition
```

copy sdrun/ target/ to SD

Buring the u-boot.imx to sdcard

129+0 records in

129+0 records out

132096 bytes (132 kB, 129 KiB) copied, 0.161529 s, 818 kB/s

431+0 records in

431+0 records out

441344 bytes (441 kB, 431 KiB) copied, 0.422838 s, 1.0 MB/s

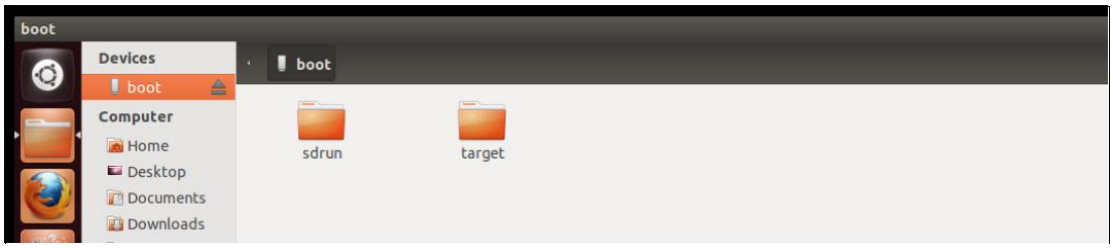
Syncing....

Un-mount the partitions

Remove created temp directories

Operation Finished

6, After the card is created, you can see that the boot partition contains two directories: sdrun and target.



Explanation:

- The contents of the sdrun folder are used to boot the system and write the image; generally, no modifications are needed.
- The contents of the target directory will be written to the flash chip. If the user has modified the image and needs to replace files, simply replace the corresponding files in the target directory, keeping the same filenames, and then perform the system flashing again.

The following provides an introduction to the files inside the target directory of the SD flashing card:

Image Name	Description
u-boot-imx6ull14x14evk_nand.imx	BootLoader image
zimage	Kernel image
okmx6ull-s-nand.dtb	Device tree image
logo.bmp	Boot logo image. To change the boot logo, simply create a new BMP image (refer to “User Data Application Notes” for creation method), name it logo.bmp, and replace the existing file.
rootfs-console.tar.bz2	File system, no Qt interface or Qt library. After creating a new file system, name it rootfs_nogpu.tar.bz2 and replace this file. You can then flash your own file system.
modules.tar.bz2	Module files (extract to file system during flashing)

4.1.2 SD Card Flashing Method

Insert the SD card prepared in the previous section, and set the DIP switches as shown below: switches 3, 5, and 8 set to ON, and switches 1, 2, 4, 6, and 7 set to OFF, as illustrated. At this point, the contents of the target folder on the SD card will be flashed to the eMMC.

The flashing process may take some time. After the system finishes flashing, the serial port will display the following information:

```
./lib/modules/4.1.15-00025-g88c5284/kernel/fs/isofs/isofs.ko
./lib/modules/4.1.15-00025-g88c5284/kernel/fs/configfs/
./lib/modules/4.1.15-00025-g88c5284/kernel/fs/configfs/configfs.ko
./lib/modules/4.1.15-00025-g88c5284/modules.builtin
./lib/modules/4.1.15-00025-g88c5284/modules.dep
./lib/modules/4.1.15-00025-g88c5284/modules.alias
./lib/modules/4.1.15-00025-g88c5284/modules.symbols.bin
./lib/modules/4.1.15-00025-g88c5284/modules.devname
./lib/modules/4.1.15-00025-g88c5284/modules.softdep
./lib/modules/4.1.15-00025-g88c5284/source
./lib/modules/4.1.15-00025-g88c5284/modules.dep.bin
./lib/modules/4.1.15-00025-g88c5284/modules.symbols
Update Complete!!!!!!!
```

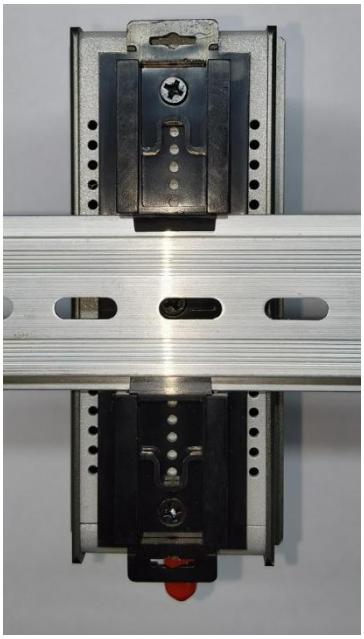
After the flashing is complete, power off the system. Set the DIP switches as follows: switches 3 and 7 to ON, and switches 1, 2, 4, 5, 6, and 8 to OFF. Then power on the system again to boot from the eMMC.

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
- Debian 10

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