

ARMxy Embedded Computer



BL340 User Manual

Version: V1.1

Date: 2024-11-22

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 BL340.

Copyright

This user manual is owned by Shenzhen Beilai Technology Co., Ltd. No one is authorized to copy, distribute or forward any part of this document without written approval of Shenzhen Beilai Technology. Any violation will be subject to legal liability.

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
2024/6/17	V1.0	Initial Release	LKY
2024/11/22	V1.1	Supplemental I/O Port Description	LKY

Table of Contents

1 Introduction	5
1.1 Overview	5
1.2 Appearance	6
1.3 Technical Specifications	7
1.4 Model Selection	9
1.4.1 Host Model Selection	9
1.4.2 SOM Selection	10
1.4.3 X Series I/O Board Selection	10
1.4.4 Y Series I/O Board Selection	11
2 Hardware	12
2.1 Power Interface	12
2.2 I/O Module Port Description	12
2.2.1 RS232&RS485 Module	12
2.2.2 DI Module	13
2.2.3 DO Module	13
2.2.4 AO Module	13
2.2.5 AI Module	14
2.2.6 RTD Module	14
2.2.7 Multiple Port Combinations	14
2.2.8 X board Port Usage	15
2.2.9 Y Board Port Usage	16
2.3 LED	19
2.4 Ethernet Port	20
2.5 USB Port	20
2.6 HDMI	21
2.7 Debugging Serial Port	21
2.8 SIM Card Slot	22
2.9 SD Card Slot	22

2.10 Reset Button	22
2.11 PCIe	22
2.11.1 4G Module	22
2.11.2 Wi-Fi Module	24
2.12 Antenna Interface	25
2.13 Hardware Watchdog	26
2.14 Encryption Chip	26
2.15 External RTC	26
3 Device Login	27
3.1 USB Login	27
3.2 SSH2 Login	28
4 System Programming	29
4.1 Micro SD Card Boot	29
4.1.1 Boot Card Creation	29
4.1.2 Boot from the Boot Card	32
4.2 EMMC Boot	32
4.2.1 Programming Card Creation	32
4.2.2 System Programming	34
4.3 System Partition	35
5 Software	36
6 GPIO	37
7 Warranty Terms	38
8 Technical Support	38

1 Introduction

1.1 Overview

ARMxy BL340 series ARM embedded computer is a low-cost, industrial-grade ARM edge computer with flexible IO and SoM configuration options.

The BL340 series is based on Allwinner Technology's T507-H processor, a quad-core ARM Cortex-A53 industrial controller with a maximum frequency of 1.4 GHz, paired with 8/16 GB eMMC and 1/2 GB DDR4.

BL340 series ARM embedded computer supports Linux-4.9.170, Linux-RT-4.9.170, Ubuntu 20.04, Docker containers, Node-Red, and Qt-5.12.5 for GUI development. It comes preinstalled with:

BLIoTLink, an industrial protocol conversion software for fast industrial data acquisition and integration with major IoT cloud platforms and SCADA systems.

BLRAT, a remote access tool for device monitoring and maintenance.

QuickConfig, a rapid configuration tool for remote device setup and debugging, improving operational efficiency.

It also supports Node-Red, enabling quick IoT application development, and AI-assisted programming for a "what you see is what you get" development experience.

The BL340 series features professional electrical design and has passed high- and low-temperature testing, ensuring stable and reliable performance. It supports DIN35 rail mounting and is suitable for various industrial environments.

Typical applications include industrial IoT, edge computing gateways, photovoltaic power and energy storage systems, automation control, and rail transportation.

1.2 Appearance



Standard Model



C Model



A Model



B Model



Standard Model

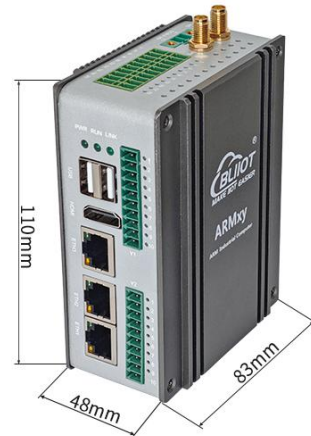


A Model



B Model


Standard Model

A Model

B Model

1.3 Technical Specifications

	Parameter	Description
System	CPU	(AllWinner)T507-H, Quad-core A53
	Clock Speed	1.416GHz
	GPU	G31 MP2, support OpenGL ES 1.0/2.0/3.2, Vulkan1.1, OpenCL 2.0
	RAM	DDR4 1/2G
	Storage	eMMC 8/16G
Power	Input Voltage	DC 12~24V
	Consumption	Normal: 240mA@12V(with 4G module), 220mA@12V (without 4G module) Maximum: 700mA@12V
	Reverse Polarity	Reverse Polarity Protection
Ethernet	Specification	1~3*RJ45, 3x100M, adaptive MDI/MDIX
	Protection	ESD ±2kV (contact), ±8kV (air);
SIM Card	Slot	1
	Type	Drawer interface
Serial Ports (Optional)	Channels	2/4/8xRS485 or RS232
	Baud Rate	300bps-115200bps
	Data Bit	7,8

	Parity Bit	None, Even, Odd
	Stop Bit	1, 2
X board Digital Input (Optional)	Channels	2/4/8/16 channel
	Type	Dry contact or Wet contact
	Dry Contact	Closed: short-circuited Open: circuit open
	Wet Contact	Logic 0: 0-3VDC Logic 1: 10-30VDC
	Isolation Protection	2KVrms
X board Digital Output (Optional)	Channels	2/4/8/16 channel
	Type	SINK
	Output capacity	Single channel 100mA
USB	Quantity	1*micro USB, 2*USB 2.0 HOST
SD Card	Slot	1
	Type	Support SD, SDHC and SDXC(UHS-I) card
HDMI	Quantity	1
Antenna	Interface	1xWi-Fi/4G, 1xGPS 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
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	110x83x42mm or 110x83x48mm
	Protection Level	IP30
	Installation	DIN35 rail mounted, wall mounting
	System	Ubuntu20.04

1.4 Model Selection

1.4.1 Host Model Selection

Model	ETH	USB	HDMI	X I/O Board	Y I/O Board	Dimension
BL340	1×100M	2	×	1×6PIN	×	42×83×110mm
BL340A	1×100M	2	×	1×20PIN	×	42×83×110mm
BL340B	1×100M	2	×	1×20PIN	2	48×83×110mm

BL340C	1×100M	2	×	1×10PIN	×	42×83×110mm
BL341	2×100M	2	×	1×6PIN	×	42×83×110mm
BL341A	2×100M	2	×	1×20PIN	×	42×83×110mm
BL341B	2×100M	2	×	1×20PIN	2	48×83×110mm
BL342	3×100M	2	1	1×6PIN	×	42×83×110mm
BL342A	3×100M	2	1	1×20PIN	×	42×83×110mm
BL342B	3×100M	2	1	1×20PIN	2	48×83×110mm

1.4.2 SOM Selection

ARMxy BL340 Series SOM Selection

Model	MCU	Clock Speed	Kernel	eMMC	DDR4	Temperature
SOM340	T507-H	1.4GHz	4 x A53	8GByte	1GByte	Industrial grade -40~85°C
SOM341	T507-H	1.4GHz	4 x A53	16GByte	2GByte	Industrial grade -40~85°C
SOM342	T507-H	1.4GHz	4 x A53	8GByte	1GByte	Commercial grade 0~70°C
SOM343	T507-H	1.4GHz	4 x A53	16GByte	2GByte	Commercial grade 0~70°C

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.

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

BL340 does not support I/O boards with CAN ports.

Model	RS232/RS485	DI	DO	GPIO	PIN
X10	2	x	x	x	6PIN
X13	x	2	2	x	6PIN
X14	x	4	x	x	6PIN
X15	x	x	4	x	6PIN
X16	x	x	x	4	6PIN
X20	4	x	x	x	10PIN
X23	4	4	4	x	20PIN
X26	2	8	4	x	20PIN
X28	2	12	x	x	20PIN
X30	x	x	x	16	20PIN

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.

Y Series I/O Board Selection

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, $\pm 5V/\pm 10V$	Y95	4xPWM Output(NPN) + 4xPulse Counter Input
Y37	4xIEPE	Y96	4xPWM Output(PNP) + 4xPulse Counter Input

Ordering Notes

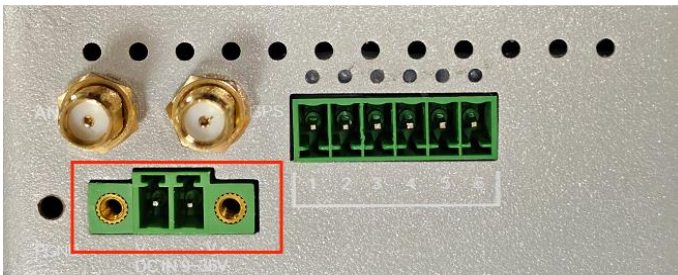
Y01: DI channels support dry contacts or NPN-type wet contact sensors.

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: The ttyS0 port on the X board is also used as a debug serial port. The COM port is for DI dry contact use, while GND is for DI wet contact use.

2.2.1 RS232&RS485 Module

X10 Module					
1	2	3	4	5	6
ttyS1-A	ttyS1-B	GND	ttyS2-A	ttyS2-B	GND

X20 Module

1	3	5	7	9
GND	ttyS0-A	ttyS5-A	ttyS2-A	ttyS1-A
GND	ttyS0-B	ttyS5-B	ttyS2-B	ttyS1-B
2	4	6	8	10

Y63 Module

1	2	3	4	5	6	7	8	9	10
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

2.2.2 DI Module

X14 Module

1	2	3	4	5	6
DI1	DI2	GND	DI3	DI4	COM

Y11/12 Module

1	2	3	4	5	6	7	8	9	10
DI1	DI2	DI3	DI4	COM	COM	DI5	DI6	DI7	DI8

2.2.3 DO Module

X15 Module

1	2	3	4	5	6
DO1	DO2	GND	DO3	DO4	GND

Y21/22 Module

1	2	3	4	5	6	7	8	9	10
DO1	DO2	DO3	DO4	GND	GND	DO5	DO6	DO7	DO8

2.2.4 AO Module

Y41 Module

1	2	3	4	5	6	7	8	9	10
AO1	GND	AO2	GND	/	/	AO3	GND	AO4	GND

Note: Each AO module requires a fixed current of approximately 100mA.

2.2.5 AI Module

Y31 Module

1	2	3	4	5	6	7	8	9	10
AI1	GND	AI2	GND	/	/	AI3	GND	AI4	GND

Y34 Module

1	2	3	4	5	6	7	8	9	10
AI1	GND	AI2	GND	/	/	AI3	GND	AI4	GND

2.2.6 RTD Module

Y51/Y52 Module

1	2	3	4	5	6	7	8	9	10
/	PT+	PT-	PT-	/	/	/	PT+	PT-	PT-

2.2.7 Multiple Port Combinations

X13 Module

1	2	3	4	5	6
DI1	DI2	GND	DO1	DO2	GND

X16 Module

1	2	3	4	5	6
PG3	PG5	GND	PG0	PA11	GND

X23 Module

1	3	5	7	9	11	13	15	17	19
DI4	DI3	DI2	DI1	GND	COM	ttyS0-A	ttyS5-A	ttyS2-A	ttyS1-A
DO4	DO3	DO2	DO1	POWER	GND	ttyS0-B	ttyS5-B	ttyS2-B	ttyS1-B
2	4	6	8	10	12	14	16	18	20

X26 Module

1	3	5	7	9	11	13	15	17	19
DI6	DI5	DI4	DI3	GND	COM	DI2	DI1	ttyS2-A	ttyS1-A
DO4	DO3	DO2	DO1	POWER	GND	DI8	DI7	ttyS2-B	ttyS1-B
2	4	6	8	10	12	14	16	18	20

X28 Module

1	3	5	7	9	11	13	15	17	19
DI6	DI5	DI4	DI3	GND	COM	DI2	DI1	ttyS2-A	ttyS1-A
DI12	DI11	DI10	DI9	POWER	GND	DI8	DI7	ttyS2-B	ttyS1-B
2	4	6	8	10	12	14	16	18	20

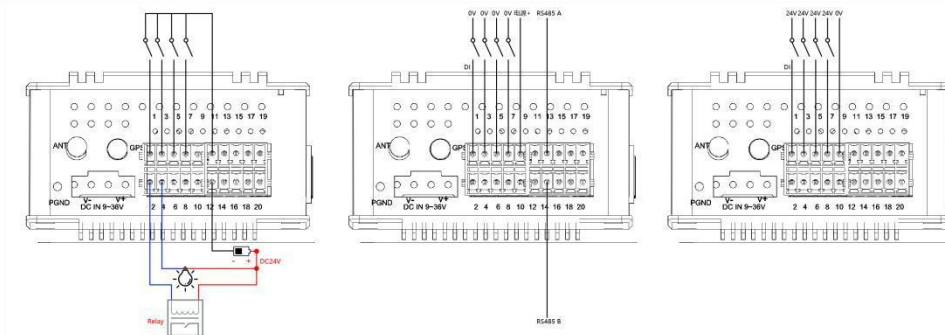
X30 Module

1	3	5	7	9	11	13	15	17	19
PH10	PG1	PG4	PG2	POWER	GND	PG9	PH2	PG15	PG6
PA11	PG0	PG5	PG3	POWER	GND	PG8	PH3	PG16	PG7
2	4	6	8	10	12	14	16	18	20

Note: The POWER port of the X30 module is 3.3V. The input and output of each port are user-defined.

2.2.8 X board Port Usage

The wiring example is shown below, using X23 as an example.



10: The POWER port can be connected to the positive terminal of the power supply to prevent reverse transient current from damaging the circuit when the DO is disconnected.

Navigate to the `/sys/class/beilai/` directory and enter `ls` to view the controllable IO ports.

```
root@bliiot:/sys/class/beilai# ls
DI1 DI2 DI3 DI4 DO1 DO2 DO3 DO4
```

For example, in the X22 module, DI1 (dry contact):

- Status 1 indicates an open state/0~3V.
- Status 0 indicates a closed state/10~30V.

```
root@bliiot:/sys/class/beilai/DI1# ls
cfg  data  device  drv  power  pull  subsystem  uevent
```

By checking the value of data, you can determine that the DI1 status is open.

```
root@bliiot:/sys/class/beilai/DI1# cat data
1
```

For DO (Digital Output), you can change the DO state by modifying the value of data. For example, writing 0 will make DO1 conduct with GND.

```
root@bliiot:/sys/class/beilai/DO1# echo 0 > data
```

RS232/485 Transmission

When using RS232/485 serial ports, connect the RS232/485 wires to the port, such as RS485-1, which has the device file /dev/ttyS1. Set the baud rate to 115200, with 8 data bits, no parity bit (8N1), and no parity checking.

```
stty -F /dev/ttyS1 ispeed 115200 ospeed 115200 cs8
echo 12345 > /dev/ttyS1 //To send data through the RS485-1 port
cat /dev/ttyS1 //To view the received data
```

Press "Ctrl+C" to stop.

2.2.9 Y Board Port Usage

(1) Software Installation

The corresponding file location is /io. Please refer to the specific file names as they appear.

Execute `chmod +x iolib_v1.1.0_install.bin` on `BEILAI_IOY_ARMV8_V1.0.bin`. then install the software.

```
root@bliiot:/# chmod +x BEILAI_IOY_ARMV8_V1.0.bin
root@bliiot:/# ./BEILAI_IOY_ARMV8_V1.0.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) Port Usage

Use `ioy show` to view IO board information, and `ioy help` to check 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,i,e,t,s,r,b,n
```

Take the DI module for example. 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

Channel values can also be obtained with the `get` command:

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

Using the AO module as an example, 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

The mode type is displayed as `4t20`, which corresponds to the current output of 4~20mA in the config mode of `ioy help`.

Set the channel value by set command:

```
root@bliiot:~# ioy set 4000 10 //Output 10mA via address setting
root@bliiot:~# ioy set 2.1 10 //Setting a channel via <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,i,e,t,s,r,b,n
```

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

```
root@bliiot:~# ioy config 4000 mode 0t20
root@bliiot:~# ioy config 2.1 mode 0t20 //change range to 0-20mA
```

To modify the range, adjust the corresponding minimum and maximum values:

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

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:

LED2 (POWER): Stays on when the power is normal after startup.

LED1 (RUN): Blinks when the system is running normally.

LED0 (LINK): Remains on when connected to the internet via a wired network; blinks when using 4G or Wi-Fi.

The configuration file for these settings is `/etc/beilai_led.sh`

To view the trigger conditions: `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
```

Where [none] means the current trigger condition of led0 is none. Write the above string to trigger to modify the trigger condition.

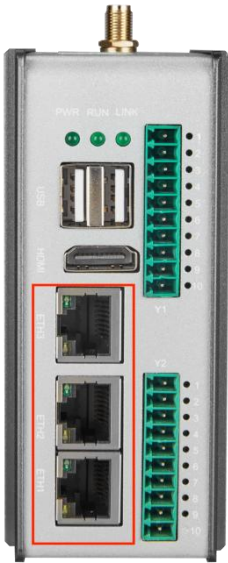
When the led trigger condition is set to none, the user can control the led light on or off by commands

To control LED0 to be on: `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
```

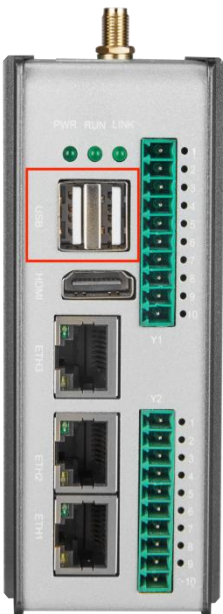
To control LED1 to be off: `echo 0 >/sys/class/leds/user-led1/brightness`

2.4 Ethernet Port



The device is equipped with 3x100M Ethernet ports

2.5 USB Port



The device has 2 USB 2.0 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.

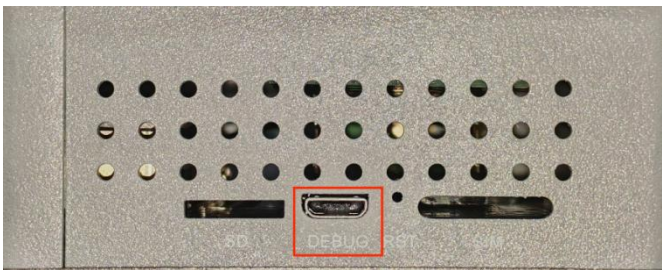
2.6 HDMI



The HDMI interface, as shown in the image, supports both HDMI 1.4 and HDMI 2.0 standards.

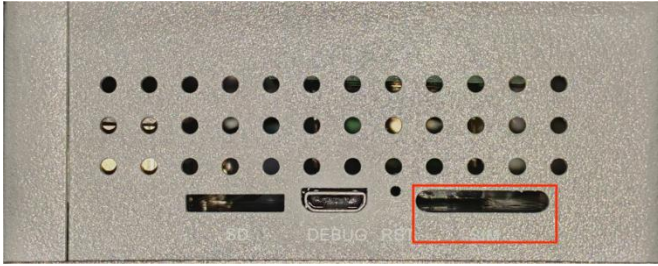
The system default resolution is 1920x1080@60fps. Supports a maximum HDMI display resolution of 4K.

2.7 Debugging Serial Port



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

2.8 SIM Card Slot

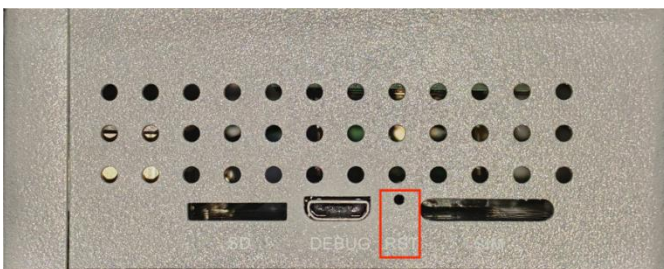


2.9 SD Card Slot



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.

2.10 Reset Button



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

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  
udhcpc -i usb0  
ifconfig
```

At this point, a network node usb0 should be generated. If this node is not present, the module may not have network functionality enabled by default. Try configuring the 4G module with the following commands:

(The AT command port for the EC200 series module is /dev/ttyUSB1.)

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

If using the EC200 module, an additional command may be required to establish a network connection.

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

After executing the command, if the device returns "OK", it indicates that the configuration is successful and this setup needs to be done only once. After rebooting the device, the usb0 node should be created. Then, you can re-run the network disable and enable commands.

Once the usb0 node is generated, use the following commands to test the network.

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

(2) SMS Function

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

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

Command Description:

<device>: The 4G module device node.

<phonenumber>: The target phone number for sending the SMS.

<text>: The content of the SMS. There should be no spaces between characters in the SMS content; otherwise, an error will be prompted.

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

At this time, the corresponding number should receive a text message with the content of "test".

(3) Call Function

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

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

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

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

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

(4) GPS Function

The GPS function can be tested by executing the test command in the test program directory:

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

Command Description:

<device>: The device node, determined by the result of the command `ls /dev/ttyUSB*`. This may change after rebooting the device.

<timeout>: The time to wait for the return of latitude and longitude information (in seconds).

For example: `./get_location /dev/ttyUSB2 1`

Obtaining latitude and longitude information may take a few minutes. If the retrieval fails or times out, check if the antenna is properly connected and ensure you are testing in an open area.

2.11.2 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 card is not available, you can follow the steps below to install the driver.

(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
```

```
ifconfig eth3 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 bliot -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.

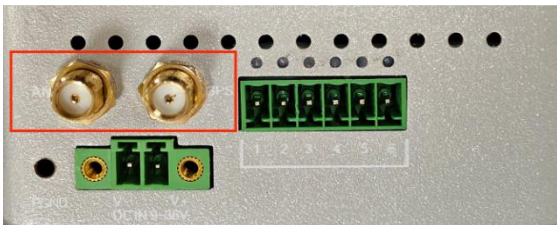
```
ifconfig eth1 down
ifconfig eth2 down
ifconfig eth3 down
insmod -f 8188eu.ko
Ifconfig wlan0 up
```

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

```
./ap_setup.sh
```

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

2.12 Antenna Interface



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

2.13 Hardware Watchdog

Note: The hardware watchdog is disabled by default.

Watchdog Control Pin: PH4. Set this pin to 1 to disable the hardware watchdog.

Watchdog Timeout: The hardware watchdog timeout is set to 30ms.

2.14 Encryption Chip

The encryption chip model RJGT102 is based on the SHA-256 encryption and authentication algorithm. It also provides a configurable watchdog timer and external reset functionality. The chip communicates with the MCU through an I²C-5 serial interface and supports a low-power mode. The device uses the encryption chip's demo by writing `/proc/sys/kernel/random/uuid` to the encryption chip and saving the UUID to `/usr/rjgt_unique.json`. When used, the data from the encryption chip is compared with the external data. If the external data matches the internal data on the encryption chip, the encryption validation is successful.

Please modify the cross-compiler path in the Makefile and then run `make` to compile. Alternatively, refer to the "RJGT102 Data Manual" for more details.

Run the sample programme `rigt102` and if the uuid is correct, the following reply 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
```

2.15 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@BL340:~# 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

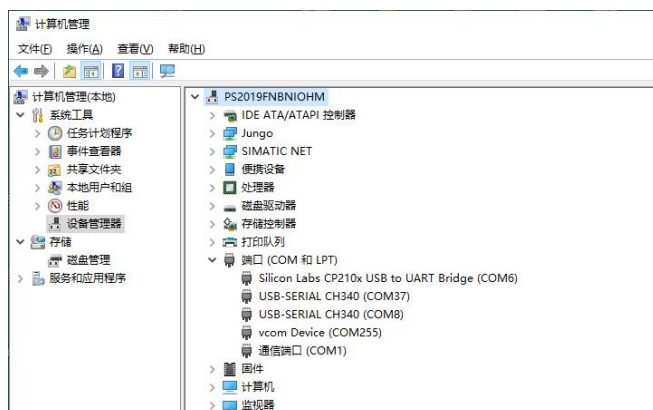
```
root@bliiot:~# hwclock --hctosys -u
```

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:

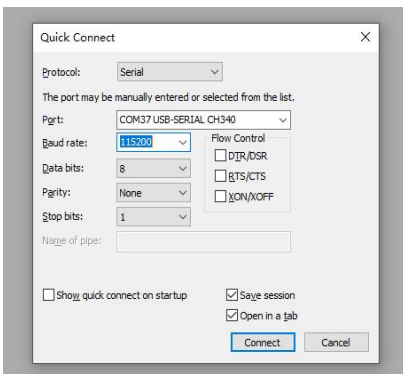
1. Baud rate: 115200
2. Data bits: 8
3. Parity: None
4. Stop bits: 1
5. Click "Connect" to access the device.

Linux systems do not have a default login password set.

For Ubuntu systems:

Default Login Account: root

Password: root



3.2 SSH2 Login

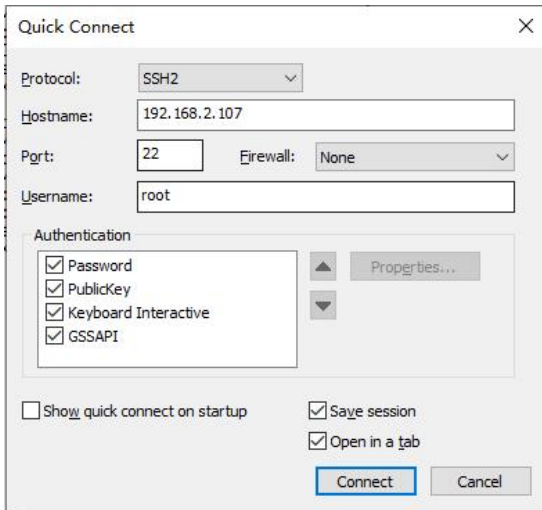
Before logging in via the network port, you need to set the IP address for the corresponding port. For example, ETH2 is connected to the router, and the obtained IP address is 192.168.2.107. The computer IP should be in the same subnet.

```
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)

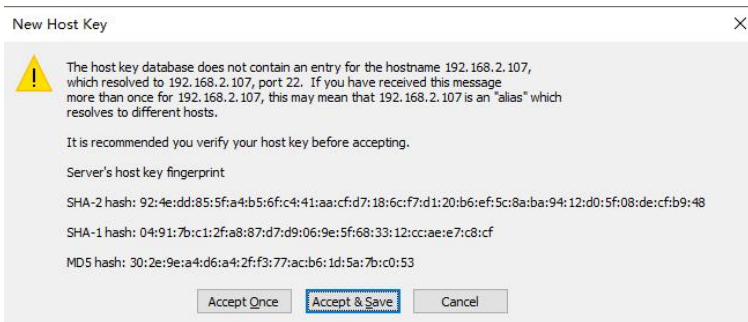
eth2    Link encap:Ethernet  HWaddr 00:E0:99:CD:55:B9
         inet addr:192.168.2.107  Bcast:192.168.2.255  Mask:255.255.255.0
         inet6 addr: fd5f:4184:3ad4:4:66ee:75b9:2b9:476d/64 Scope:Global
         inet6 addr: fe80::5d2c:48eb:826c:7f6/64 Scope:Link
         inet6 addr: fd5f:4184:3ad4:4::74e/128 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

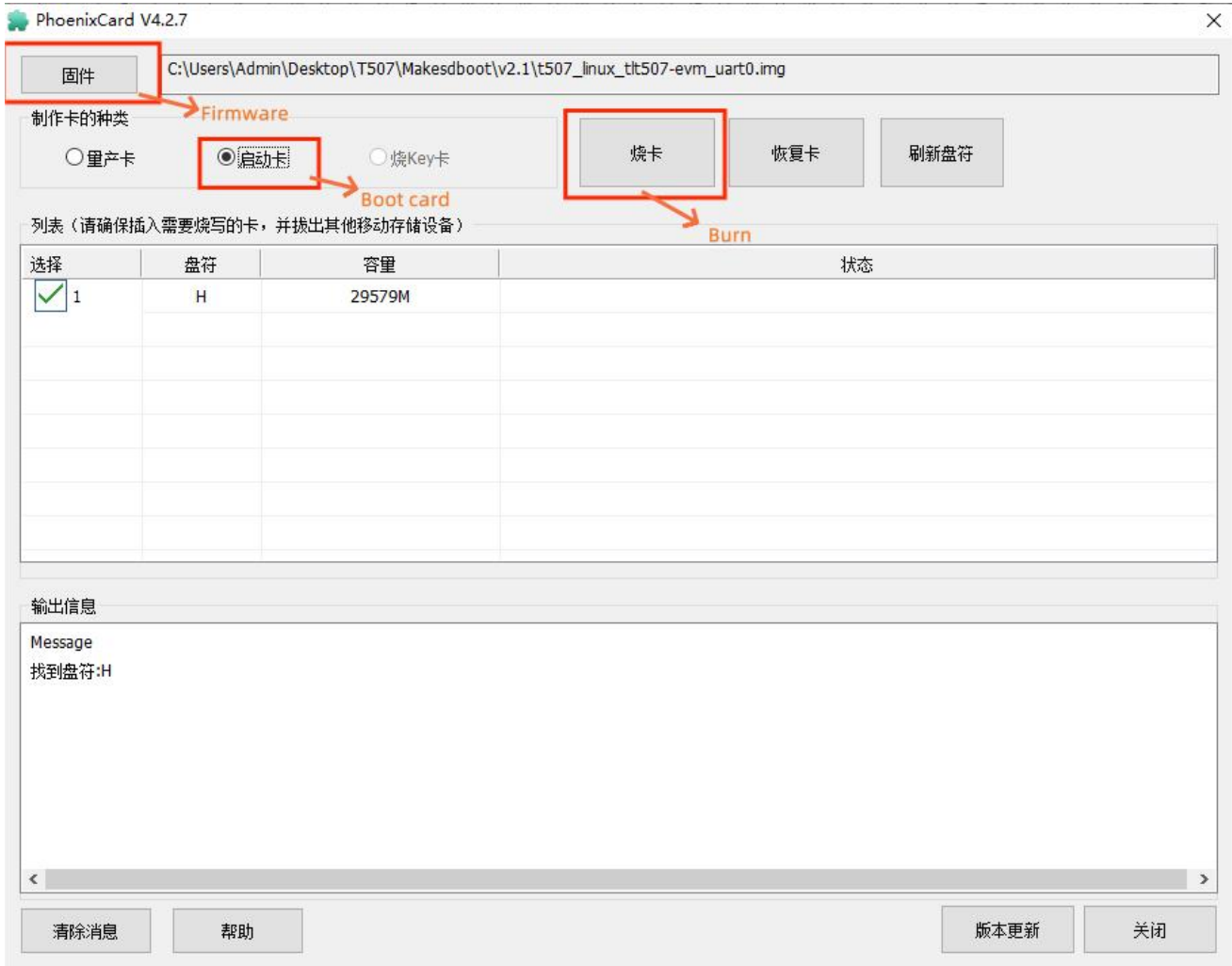
4.1.1 Boot Card Creation

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

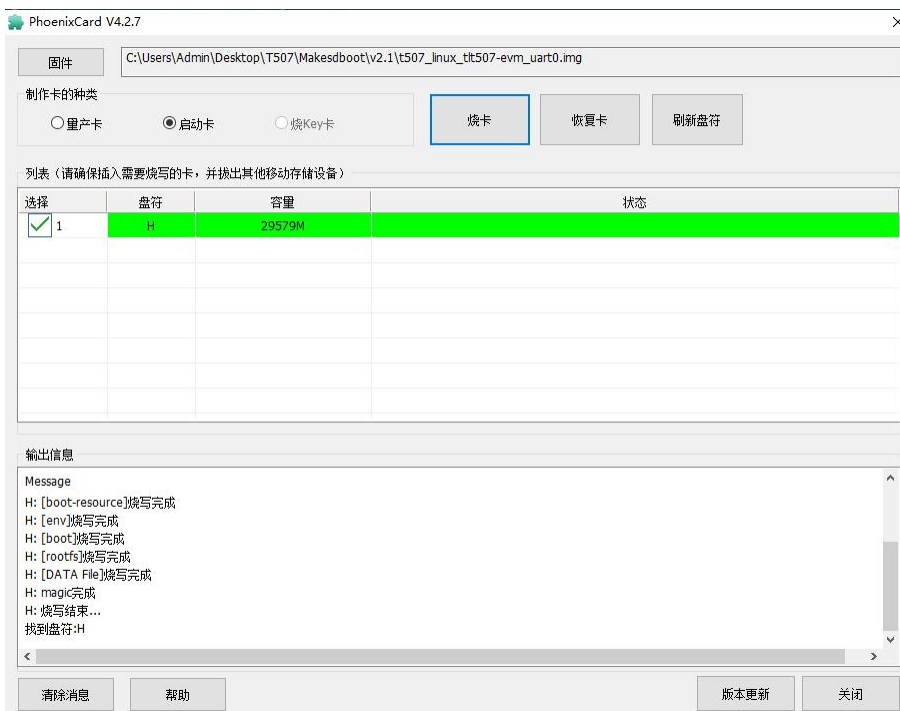
名称	修改日期	类型	大小
ini_fun.lhs	2020/06/15 10:26	LHS 文件	4 KB
IniParasPlg.dll	2020/06/15 10:26	Adobe Acrobat ...	44 KB
Langplg.dll	2020/06/15 10:26	Adobe Acrobat ...	56 KB
LangPlgex.dll	2020/06/15 10:26	Adobe Acrobat ...	24 KB
LangPlgex_1.dll	2020/06/15 10:26	Adobe Acrobat ...	24 KB
lua5.1.dll	2020/06/15 10:26	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

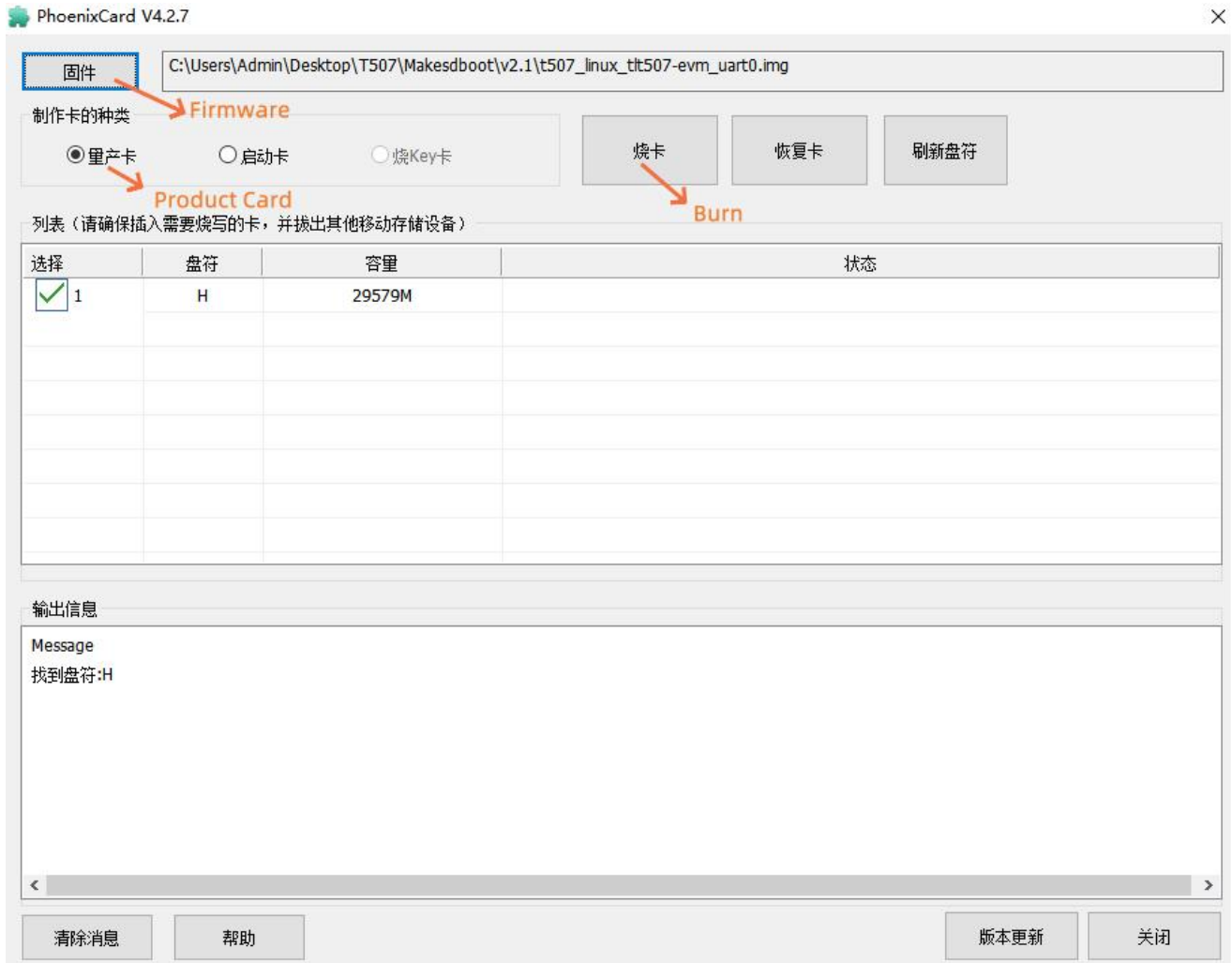
Connect the blank SD card to your computer, then double-click "PhoenixCard.exe" to open the system firmware tool.

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
PhoenixCardFundll	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

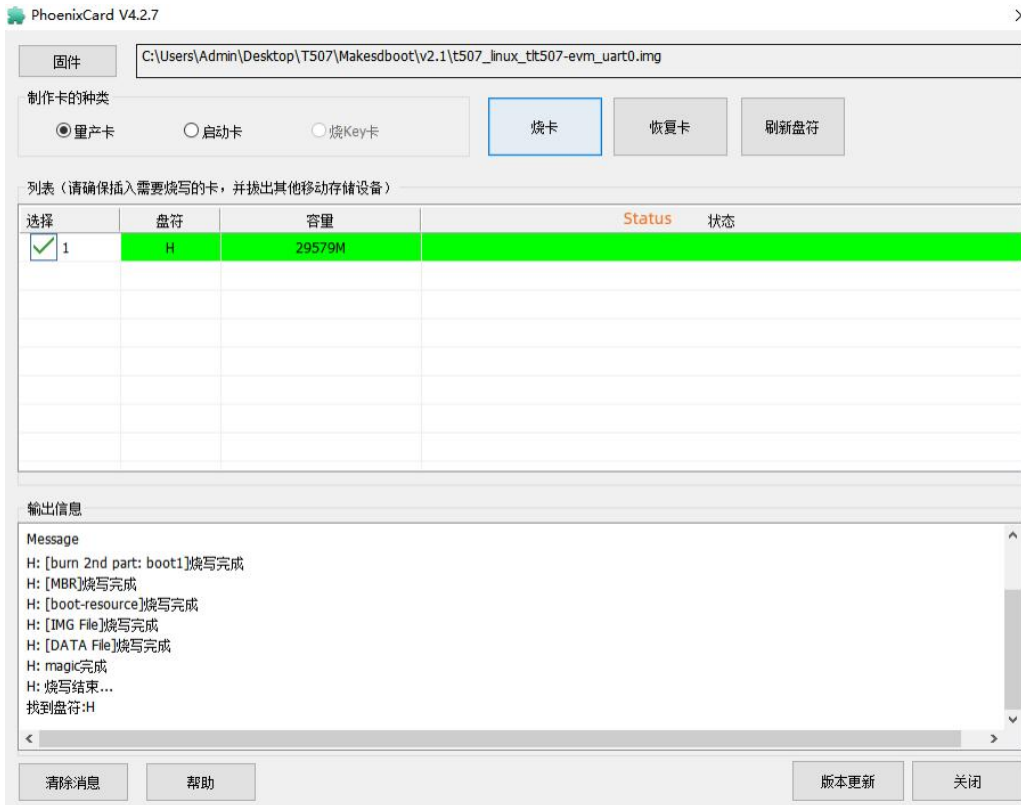
The tool will automatically detect the connected Micro SD card on the PC side.

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] = 00000000
dram para[4] = e0e0e0e
dram para[5] = 1919
dram para[6] = dbddaa99
dram para[7] = 60fa
dram para[8] = 4000000
dram para[9] = 520
dram para[10] = 601
dram para[11] = 8
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]succeeded 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.

```
[260]Loading boot-pkg Succeed(index=0).
[264]Entry_name      = u-boot
[271]Entry_name      = monitor
[274]Entry_name      = dtb
[278]tunning data addr:0x4a0003e8
[281]Jump to second Boot.
NOTICE: BL3-1: v1.0(debug):f0130ea
NOTICE: BL3-1: Built : 10:23:13, 2022-04-14
NOTICE: BL3-1 commit: 8
NOTICE: cpuidle init version v2.0
ERROR: Error initializing runtime service tspd_fast
NOTICE: BL3-1: Preparing for EL3 exit to normal world
NOTICE: BL3-1: Next image address = 0x4a000000
NOTICE: BL3-1: Next image spsr = 0x1d3

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

[00.359]CPU: Allwinner Family
[00.362]Model: TLT507-EVM HDMI
I2C: ready
[00.367]DRAM: 1 GiB
[00.370]Relocation Offset is: 35f04000
[00.395]secure enable bit: 0
[00.397]pmu_axp152_probe pmic_bus_read fail
[00.402]PMU: AXP858
[00.416][axp][err]: can't find rtcldo_vol from table
[00.421][axp][err]: can't find rtcldo_vol from table
[00.426]CPU=1008 MHz,PLL6=600 Mhz,AHB=200 Mhz, APB1=100Mhz MBus=400Mhz
[00.433]drv_disp_init
[00.472]drv_disp_init finish
[00.474]gic: sec monitor mode
[00.479]ir_sys_cfg: error:no node for ir boot recovery settings
[00.486]flash init start
[00.488]workmode = 0, storage type = 2
[00.492]MMC: 2
[00.493][mmc]: mmc driver ver uboot2018:2020-5-24 16:54:00-202109221409
[00.500][mmc]: get sdc_type fail and use default host:tm4.
[00.510][mmc]: SUNXI SDMMC Controller Version:0x40502
[00.535][mmc]: Best spd md: 3-HS200/SDR104, freq: 4-150000000, Bus width: 8
[00.541]sunxi flash init ok
```

4.3 System Partition

Device name	Partition	Description
Linux system boot card	/dev/mmcbk1p1	The "boot-resource" partition is used to store resources such as "bootlogo.bmp".
	/dev/mmcbk1p2	The "env" partition stores U-Boot environment variables.
	/dev/mmcbk1p3	The "boot" partition stores the kernel image.
	/dev/mmcbk1p4	The "rootfs" partition stores the root filesystem image.
	/dev/mmcbk1p5	The "private data" partition stores private data.
	/dev/mmcbk1p6	The "user partition" stores user data.
eMMC device	/dev/mmcbk0p1	The "boot-resource" partition stores resources such as "bootlogo.bmp".
	/dev/mmcbk0p2	The "env" partition stores U-Boot environment variables.
	/dev/mmcbk0p3	The "boot" partition stores the kernel image.
	/dev/mmcbk0p4	The "rootfs" partition stores the root filesystem image.
	/dev/mmcbk0p5	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 Software

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

6 GPIO

Port	Serial Port Device
RS485-1	/dev/ttyS1
RS485-2	/dev/ttyS2
RS485-3	/dev/ttyS5
RS485-4	/dev/ttyS0

Port	Pin Number
DI1	PG2
DI2	PG4
DI3	PG1
DI4	PH10
DO1	PG3
DO2	PG5
DO3	PG0
DO4	PA11
Hardware Watchdog	PH4

Port	IO type	Function 2	Function 3	Function 4	Function5	Function6	
PA11	I/O	TWI3_SDA				PA_EINT11	DO4
PG0	I/O	SDC1_CLK				PG_EINT0	DO3
PG1	I/O	SDC1_CMD				PG_EINT1	DI3
PG2	I/O	SDC1_D0				PG_EINT2	DI1
PG3	I/O	SDC1_D1				PG_EINT3	DO1
PG4	I/O	SDC1_D2				PG_EINT4	DI2
PG5	I/O	SDC1_D3				PG_EINT5	DO2
PG6	I/O	UART1_TX		JTAG_MS		PG_EINT6	RS485
PG7	I/O	UART1_RX		JTAG_CK		PG_EINT7	RS485
PG8	I/O	UART1_RTS	PLL+LOCK	JTAG_DO		PG_EINT8	RS485

			_DBG				
PG9	I/O	UART1_CTS		JTAG_DI		PG_EINT9	RS485
PH0		UART0_TX		PWM3	TWI1_SC K	PH_EINT0	RS485
PH1		UART0_RX		PWM4	TWI1_SD A	PH_EINT1	RS485
PH2		UART5_TX	OWA_MCL K	PWM2	TWI2_SC K	PH_EINT2	RS485
PH3		UART5_RX		PWM1	TWI2_SD A	PH_EINT3	RS485
PG5		UART2_TX			TWI4_SC K	PE_EINT1 6	RS485
PG6		UART2_RX			TWI4_SD A	PE_EINT1 7	RS485
PH10			IR_RX	TCON_TRI G1		PE_EINT1 0	DI4
X23~X26 use PG8 and PG9 as GPIO, while on other X boards, they are used for RS485.							

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>