Embedded ARM Computer BL301 BL302

BL301/302 User Manual

Version: V1.2

Date: 2023-6-8

Shenzhen Beilai Technology Co.,Ltd

Website: https://www.bliiot.com

Preface

Thanks for choosing BLIIoT Embedded ARM Computer BL301 BL302. These operating instructions contain all the information you need for operation of a device in the EdgeCOM BL30 family.

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
December 27, 2022	V1.0	Initial Release	LKY
February 28, 2023	V1.1	Add Login instructions	LKY
June 8, 2023	V1.2	Added root file system installation	LKY

Table of Contents

1 Introduction

1.1 Overview

The BL301/BL302 series Embedded ARM Computer use NXP I.MX6ULL processor, with advanced ARM Cortex-A7 architecture, running speed up to 800MHz. BL302 comes with 4 RS485 or RS232, 1 CAN port, 2 Ethernet ports, 2 DI, 2 PWM output and 1 USB port, 1 power input/output port, 1 HDMI, 1 Mini PCIe expansion slot for a wireless module. The computer supports LINUX, Ubuntu, Debian and other OS; Node-Red, QT, Python, C++; MySQL, InfluxDB, SQLite and other databases. This tiny embedded computer is widely applicable to a variety of industrial solutions.

1.2 Features

- NXP I.MX6ULL processor, ARM Cortex-A7 architecture
- Dual 10/100 Mbps Ethernet ports; RS485 or RS232 serial ports
- 1 Mini PCIe expansion slot for 4G/5G/WiFi module
- Supports LINUX, Ubuntu, Debian; Node-Red, QT, Python, C++; MySQL, InfluxDB, SQLite
- Automatic frequency reduction or restart when the chip is overheated
- Chip frequency can be adjusted manually
- Multiple sleep modes, with timing wake-up function
- IP30 protection; metal shell and system are safely isolated; DIN rail installation

1.3 Application scenarios

BL301/BL302 series Embedded ARM Computer are widely applicable to IoT, Industrial IoT, digital factories, industrial automation, energy monitoring, smart security, rail transit, telecommunications, smart EV charging, human-computer interaction and other fields.

1.4 Technical Specifications

Item	Parameter	Description
System	Processor	i.MX6ULL 800MHz
	RAM	256/512MB
	Flash	256MB/8GeMMC
Power	Input Voltage	DC 9～36V
	Power Consumption	Normal: 170mA@12V, max 340mA@12V
	Wiring	Anti- Inverse Connection Protection
Ethernet Port	Interface Spec	2 x RJ45, 10/100Mbps, adaptive MDI/MDIX
	Protection	ESD ±16kV (contact), ±18kV (air), EFT 40A (5/50ns), Lightening 6A (8/20μs)
Serial Port	QTY	4 x RS485/ RS232
	Baud Rate	300bps-115200bps
	Data Bit	7, 8
	Parity Bit	None, Even, Odd
	Stop Bit	1, 2
	Protection	ESD ±8kV (contact), ±15kV (air) EFT 2KV, 40A (5/50ns)
CAN Port	QTY	1
	MAX Speed	1Mbps
SIM Card	QTY	2 SIM Card Slot
	Spec	Drawer type slot, support 1.8V/3V SIM/UIM card (NANO)
	Protection	Built-in 15KV ESD Protection
Digital Input	QTY	2
	Input Type	Both Dry contact and Wet contact(NPN)
	Dry Contact	Close: Short circuit Open: Open circuit
	Wet Contact	Logic 0: 0-3VDC Logic 1: 3-30VDC
	Isolation protection	2KVrms
Digital Output	QTY	2
	Output Type	PWM
USB Port	QTY	1xmicro USB, 1x USB2.0

	Protection	Over Current Protection
SD Card Slot	QTY	1
	Spec	Supports SD, SDHC and SDXC (UHS-I) cards
HDMI	QTY	1
Antenna	QTY	1x Cellular antenna, 1xWiFi Antenna
	Type	SMA Hole Type
4G Module (Optional)	L-E version	GSM/EDGE:900,1800MHz WCDMA:B1,B5,B8 FDD-LTE:B1,B3,B5,B7,B8,B20 TDD-LTE:B38,B40,B41
	L-CE version	GSM/EDGE:900,1800MHz WCDMA:B1,B8 TD-SCDMA:B34,B39 FDD-LTE:B1,B3,B8 TDD-LTE:B38,B39,B40,B41
	L-A version	WCDMA:B2,B4,B5 FDD-LTE:B2,B4,B12
	L-AU version	GSM/EDGE:850,900,1800MHz WCDMA:B1,B2,B5,B8 FDD-LTE:B1,B3,B4,B5,B7,B8,B28 TDD-LTE:B40
	L-AF version	WCDMA:B2,B4,B5 FDD-LTE:B2,B4,B5,B12,B13,B14,B66,B71
	CAT-1 version	GSM:900,1800 FDD-LTE:B1,B3,B5,B8 TDD-LTE:B34,B38,B39,B40,B41
5G Module (Optional)	Interface	PCIe
	5G NR	n1/n28/n41/n77/n78/n79
	LTE-FDD	B1/B3/B5/B8
	LTE-TDD	B34/B38/B39/B40/B41
	WCDMA	B1/B5/B8
WiFi(Optional)	Interface	PCIe
	Protocol	IEEE 802.11b/g/n
	Mode	STA, AP
	Frequency	2.4GHz
	Channel	Ch1 ~ Ch13
	Security	Open, WPA, WPA2
	Encryption	AES, TKIP, TKIPAES
	Connection	8(Max)

	Speed Rate	150Mbps(Max)
	Transmission distance	Outdoor/Open area, up to 20 meters
	SSID	Support
Indicator	QTY	LEDx8
Safety Certification	MTBF	≥100,000 hours
	EMC	EN 55022: 2006/A1: 2007 (CE &RE) Class B
		IEC 61000-4-2 (ESD) Level 4
		IEC 61000-4-3 (RS) Level 4
		IEC 61000-4-4 (EFT) Level 4
		IEC 61000-4-5 (Surge)Level 3
		IEC 61000-4-6 (CS)Level 4
		IEC 61000-4-8 (M/S) Level 4
	Other	CE, FCC
Environment	Working	-40～80℃, 5～95% RH
	Storage	-40～85℃，5～95% RH
Others	Case	Metal Case
	Size	BL302: 81mm×45mm×93mm(L*W*H) BL301: 81mm×30mm×93mm(L*W*H)
	Protection	IP30
	Mounting	DIN-Rail Mounting

1.5 Model Selection

Model	BL301	BL301T	BL302	BL302T
Feature	Without DI DO MINIPCIE		With DI DO MINIPCIE	
Processor	i.MX6ULL	i.MX6ULL	i.MX6ULL	i.MX6ULL
CPU Frequency	800MHz	800MHz	800MHz	800MHz
RAM	512MB	256MB	512MB	256MB
Flash	8G eMMC	256MB	8G eMMC	256MB
ETH	2 x 100M	2x 100M	2 x 100M	2x 100M
USB	1	1	1	1
RS232/RS485	2	2	4	4
CAN	x	x	1	1

SD slot	1	1	1	1
MINI-PCIe	x	x	1	1
4G(GPS)/WIFI(BLE)	x	x	√	√
SIM slot	x	x	2	2
HDMI	1	1	1	1
Audio	x	x	x	x
DI	x	x	2	2
DO	x	x	2	2
House	Metal	Metal	Metal	Metal
Temperature（°C）	-25 ~ 85	-40 ~ 85	-25 ~ 85	-40 ~ 85

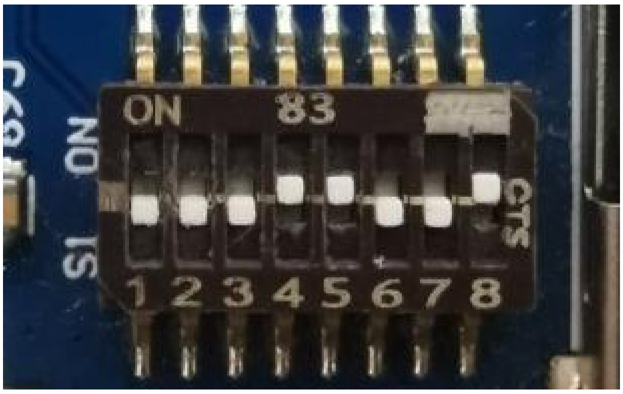
2 Development

2.1 Development Environment

- Operating System: Ubuntu20.04 64-bit
- Cross Toolchain: arm-linux-gnueabihf-gcc 4.9.0
- Bootloader version: u-boot-2016.03
- Linux kernel version: Linux-4.1.15
- Migrate QT version: QT5.6.2

2.2 System Programming

The computer supports USB OTG and SD card programming, supports eMMC startup, and uses DIP switch (S1) to distinguish different operation methods (NAND startup as shown in the figure below)



Switch / Mode	1	2	3	4	5	6	7	8
SD Card Programming	OFF	OFF	ON	OFF	ON	OFF	OFF	ON
NAND startup	OFF	OFF	OFF	ON	ON	OFF	OFF	ON
USB OTG Programming	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF
eMMC startup	OFF	OFF	ON	OFF	OFF	OFF	OFF	OFF

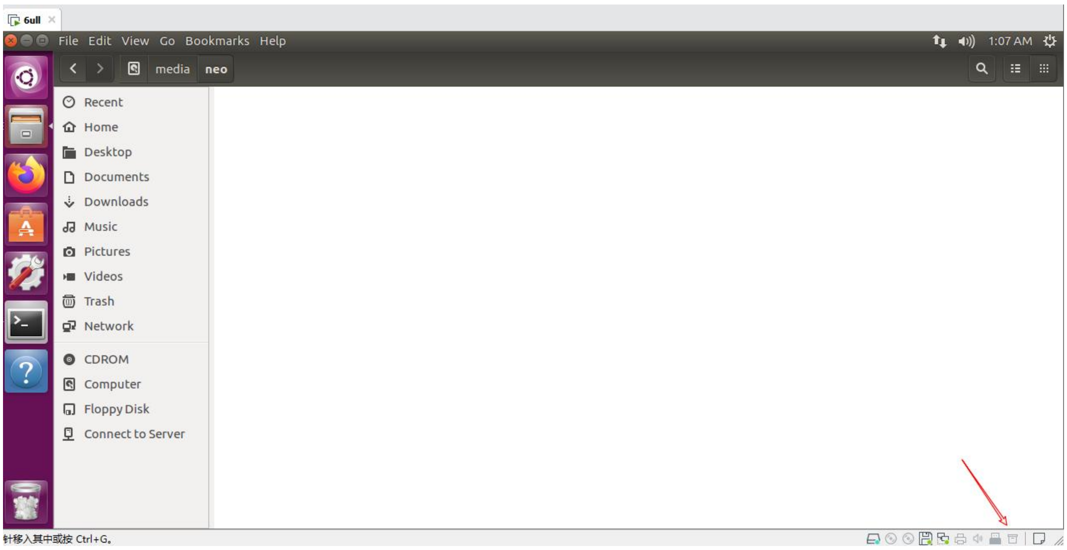
2.2.1 Programming via SD card

Copy nand-burnsd to any directory of the Ubuntu system, such as /home/beilai/work
emmc-sdburn.tar.bz2 is the 4G/8G eMMC file system

1) Format the SD card into FAT32 format before using the SD card.

2) After decompressing emmc-sdburn.tar.bz2, copy it to any directory under the ubuntu system. For example, /home/beilai/work.

3) Use a USB card reader to insert the SD card into the USB port of the computer (For VMware virtual machine users, if the USB flash drive is not recognized by the virtual machine, you can use the arrow pointing icon to connect the USB flash drive to the virtual machine).



4) After the virtual machine recognizes the SD card, and the directory pop up, and then perform the following programming operation. Enter /home/beilai/work/emmc-burnsd directory, execute the script:

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

After executing the above command, the terminal will list the computer's hard disk or U disk, and choose your SD card, enter.

Note: To determine whether your U disk is sda/sdb/sdc, it can be judged according to the capacity, for example, if the capacity of your USB flash drive is 8G, its size is 7761920 KB≈8G. Please do not insert multiple USB flash drives at the same time to avoid confusion.

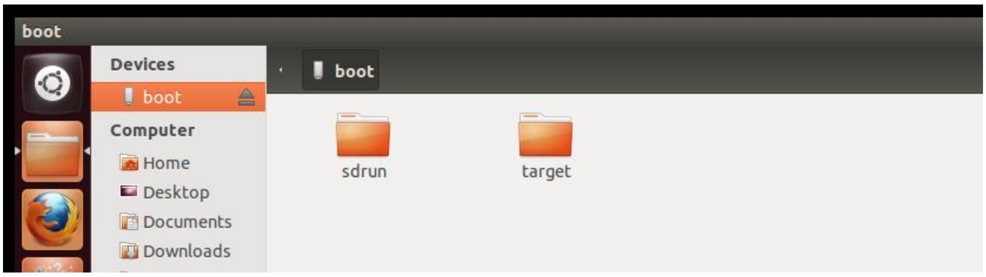
Here is an example:

```
################################################################################
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.
################################################################################
Available Drives to write images to:
# major minor size name
1: 8 16 7761920 sdb
Enter Device Number: 1 //select 1 in 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 //input y，enter，Wait for the card 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
```

5) When the operation is complete, you will see that the boot partition contains two directories: sdrun and target



The content of the sdrun folder is used to programming the system, it does not need to be modified;

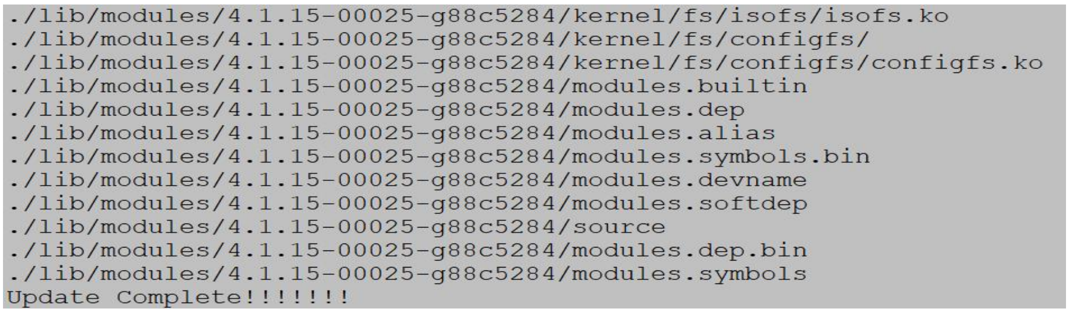
The contents of the target directory will be written to the flash chip, if user has modified the mirror image and needs to replace the mirror image file, just replace the corresponding file in the target directory and keep the same name, then re-program the system.

The following is an introduction to the files in the target of the NAND SD programming:

u-boot-imx6ull14x14evk_nand.imx	BootLoader mirror image
zImage	kernel mirror image
okmx6ull-s-emmc.dtb	device Tree Mirroring
logo.bmp	Boot logo image Users only need to make a bmp format picture to replace the boot logo image (reference method: User Profile\Application Notes), replace the file with the name **logo.bmp**.
rootfs-console.tar.bz2	File system, no QT interface and QT library. After the user creates a new file system, name it as **rootfs_nogpu.tar.bz2** and replace this file, you can program file system of your own.

modules.tar.bz2	Module file (unzip to the file system when programming)

6) Insert the finished SD card in the previous section, and set the DIP switch as shown in the figure below. 3, 5, 8 are ON, 1, 2, 4, 6, 7 are OFF, at this time, the content of the target in the SD card will be programmed into eMMC. It takes a long time to programming. After the system programmed, the serial port prints data:



```
./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!!!!!!!
```

7) At the same time LED1 on the backplane flashes.

8) When programming completed, turn off the power, turn the DIP switch to 3, 7 (ON), 1, 2, 4, 5, 6 and 8 (OFF), power on again, and NAND starts.

2.2.2 Programming via OTG

The eMMC core board defaults to the qt version of the file system. If you use the console version of the file system, you can rename **rootfs-console.tar.bz2** under the **mfgtools\Profiles\Linux\OS Firmware\files/linux** path to **rootfs -qt.tar.bz2**

The OTG programming uses the board firmware programming tool mfgtools developed by NXP, which can programming uboot, image, dtb, rootfs and other mirror images.

Following is a brief introduction to the files that users may use in the programming process.

The following paths start with: user profile\Linux\programming tools\OTG programming\mfgtools.

File name	Path	Description
mx6ull-4gemmc-512mddr-qt5.6.vbs	mfgtools	For programming eMMC core board-related mirror images
mx6ull-256mnand-256mddr-cmd.vbs	mfgtools	For programming 256M Nand core board related mirror images
ucl2.xml	Mfgtools\Profiles\Linux\OS Firmware	Defines the specific operation steps and operation content of the programming process, users can view this file for

		instructions related to single-step update
Boot the relevant image	Mfgtools\Profiles\Linux\OS Firmwarefirmware	The content of the folder is used to guide the system to programming, and generally does not need to be modified
Mirror images programming to flash	Mfgtools\Profiles\Linux\OS Firmware\files\linux	The content of the folder is used to programming into the flash. After the user modifies the mirror image, rename it to the same name and replace it, which can be used to burn your own mirror image

OTG programming method

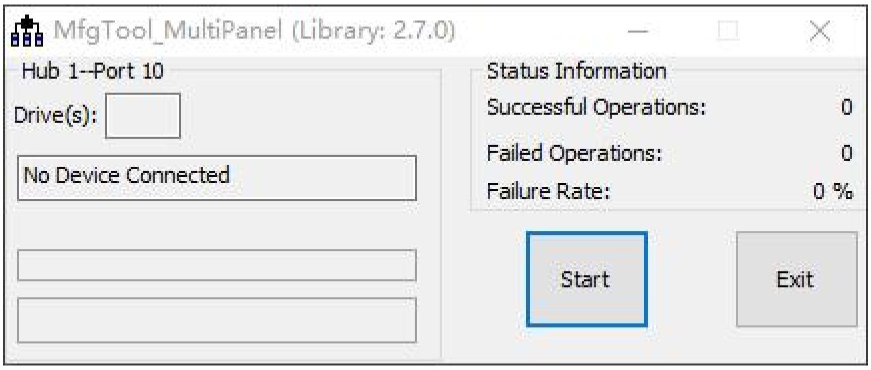
Note: When using OTG programming, the SD card cannot be inserted.

mx6ull-4gemmc-512mddr-qt5.6.vbs programming 4GeMMC+512MDDR system

1) Copy the programming tool Mfg tool to windows and decompress it. The path of the programming tool is as follows: User Profile\Linux\Programming Tools\ mfgtools.zip

2) Turn the DIP switch to 1, 2 is ON, other states are arbitrary, try to be OFF

3) Double-click "**mx6ull-4gemmc-512mddr-qt5.6.vbs**" (the script has been written and programmed directly), as shown below:



MfgTool_MultiPanel (Library: 2.7.0)

Hub 1--Port 10

Drive(s):

No Device Connected

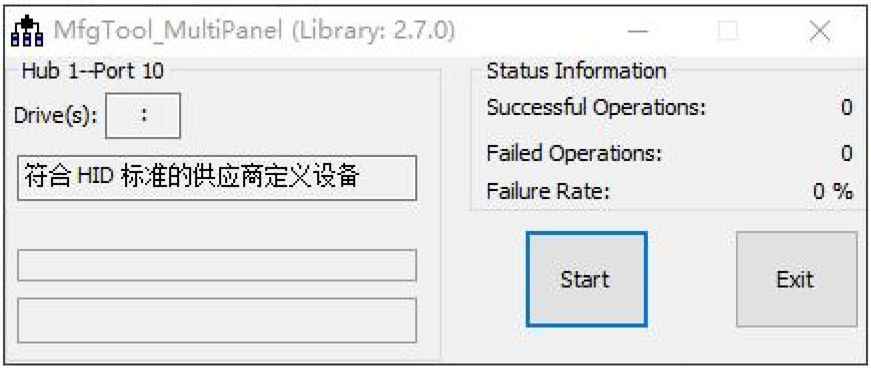
Status Information

Successful Operations: 0

Failed Operations: 0

Failure Rate: 0 %

Insert USB OTG, it will be automatically identified as HID, as shown in the figure:



MfgTool_MultiPanel (Library: 2.7.0)

Hub 1--Port 10

Drive(s): :

符合 HID 标准的供应商定义设备

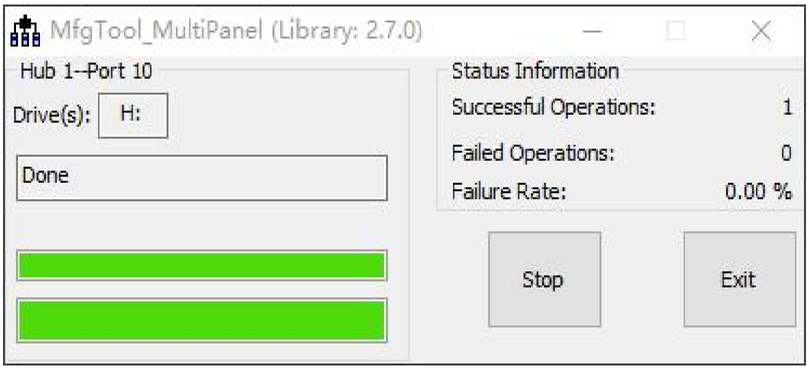
Status Information

Successful Operations: 0

Failed Operations: 0

Failure Rate: 0 %

Click start to start system programming, a formatting dialog box pops up, click the "Cancel" formatting option, or leave it alone until the programming is complete. As shown in the picture:



MfgTool_MultiPanel (Library: 2.7.0)

Hub 1--Port 10

Drive(s): H:

Done

Status Information

Successful Operations: 1

Failed Operations: 0

Failure Rate: 0.00 %

After the system programming is completed, "Done" will appear, and then click "stop" to stop. Then click "Exit" to close the programming tool. Power off, turn the DIP switch to 3, 7 are ON, 1, 2, 4, 5, 6, 8 are OFF, power on again, eMMC starts.

2.3 Root File System Installation

Copy the file fsl-imx-x11-glibc-x86_64-meta-toolchain-qt5-cortexa7hf-neon-toolchain-4.1.15-2.0.0.sh to any directory like /home/bliiot/ and execute it there:

```
~$ ./fsl-imx-x11-glibc-x86_64-meta-toolchain-qt5-cortexa7hf-neon-toolchain-4.1.15-2.0.0.sh
```

The command line will prompt: Enter target directory for SDK (default: /opt/fsl-imx-x11/4.1.15-2.0.0) Press Enter twice in a row, the program will automatically install the cross-compilation toolchain (the cross-compilation toolchain can be installed once, you don't need to reinstall it if you change terminals or reboot the system). During the installation process, make sure the network connection is working fine, Ubuntu system can access the external network. You can determine whether the installation is successful by printing out the information.

The main purpose of setting up the compilation environment is to specify the target architecture and the cross-compilation toolchain, as well as the paths of some libraries used in the compilation process, etc. Use the following command to configure the environment variables. Use the following command to configure environment variables (. followed by a space):

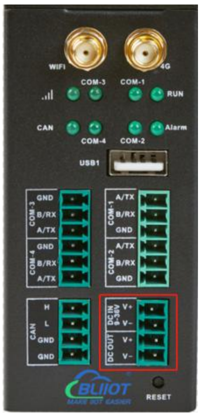
```
./opt/fsl-imx-x11/4.1.15-2.0.0/environment-setup-cortexa7hf-neon-poky-linux-gnueabi
```

Then use the command arm-poky-linux-gnueabi-gcc -v to determine if the setting is successful (note: -v is preceded by a space). Under normal circumstances, the version of gcc will be printed out.

Go to /opt/fsl-imx-x11/4.1.15-2.0.0/sysroots/cortexa7hf-neon-poky-linux-gnueabi and pack the folder with the name rootfs-console.tar.bz2 or rootfs-qt.tar.bz2, corresponding to BL301T/302T and BL301/302, and place the files in mfgtools\Profiles\Linux\OS Firmware\files, respectively corresponding to BL301T/302T and BL301/302, and place the files under mfgtools\Profiles\Linux\OS Firmware\files. Then just start programming.

3 Hardware Specifications

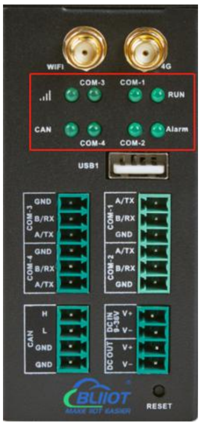
3.1 Power Interface



BL302 comes with 1 power input and 1 power output. Support DC 9~36V input/output, anti-reverse connection protection.

3.2 LED Indicators

The following figures shows the LED indicators, and the order from left to right and from top to bottom is LED6, LED5, LED1, LED2, LED8, LED7, LED3, LED4, correspondence with the LEDs in the /sys/class/leds directory.



View trigger conditions:

```
root@fl-imx6ull:~# cat /sys/class/leds/led1/trigger

[none] rc-feedback nand-disk mmc0 timer oneshot heartbeat backlight gpio
```

[none] means that the current trigger condition of led1 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 and off through commands.

```
root@fl-imx6ull:~# echo none > /sys/class/leds/led1/trigger
```

Control LED1 on

```
root@fl-imx6ull:~# echo 1 > /sys/class/leds/led1/brightness
```

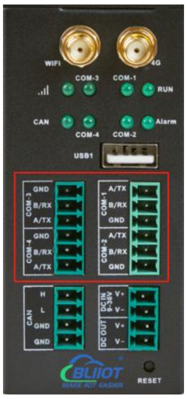
Control LED1 off

```
root@fl-imx6ull:~# echo 0 > /sys/class/leds/led1/brightness
```

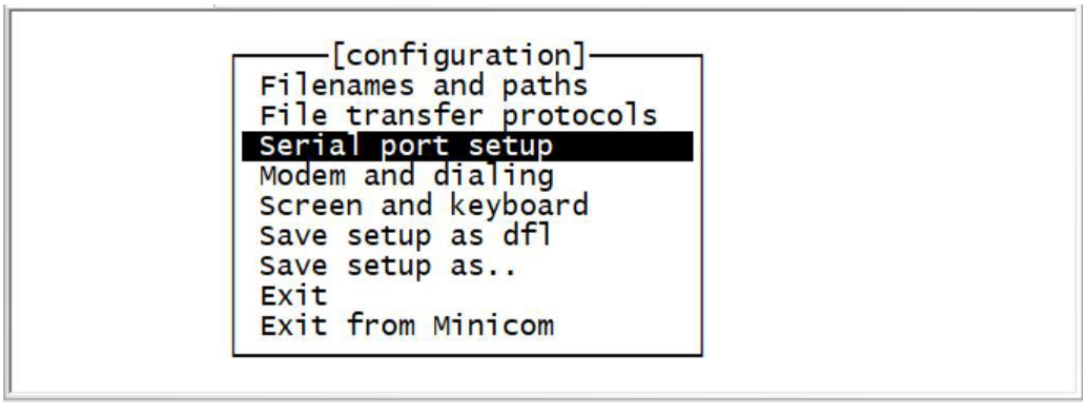
The other 7 LED lights are similar, just change the **/sys/class/leds** to the LED light corresponding to the corresponding ledx. Such as /sys/class/leds/led2/brightness

3.3 RS485&RS232 Serial Port

Depending on the chip on board, BL302 comes with RS485 or RS232. COM1, COM2, COM3 and COM4 are corresponding to /dev/ttymxc1, /dev/ttymxc2, /dev/ttymxc5 and /dev/ttymxc4 respectively. The R485 serial port supports a maximum baud rate of 115200 with a cable length of 200 meters.

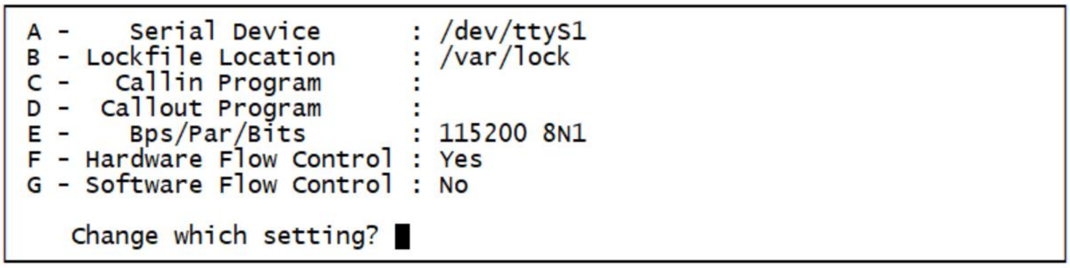


For debugging, enter "minicom -s" in the device, open the minicom configuration interface, and then select "Serial port setup", as shown in the figure.



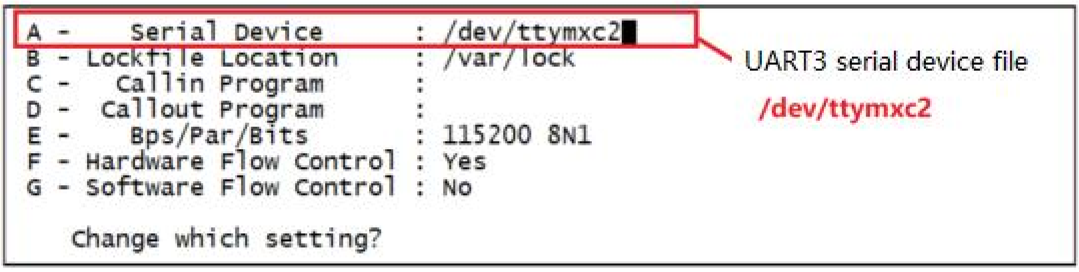
```
┌─────────[configuration]──────────┐
│ Filenames and paths              │
│ File transfer protocols          │
│ Serial port setup                │
│ Modem and dialing                │
│ Screen and keyboard              │
│ Save setup as dfl                │
│ Save setup as..                  │
│ Exit                             │
│ Exit from Minicom                │
└──────────────────────────────────┘
```

Select "Serial port setup" and press Enter to enter the setup menu, as shown in the figure.



```
A -    Serial Device      : /dev/ttyS1
B - Lockfile Location     : /var/lock
C -   Callin Program      :
D -  Callout Program      :
E -    Bps/Par/Bits       : 115200 8N1
F - Hardware Flow Control : Yes
G - Software Flow Control : No

   Change which setting?
```

There are 7 setting items in the figure, corresponding to A, B...G, for example, the first one is to select the serial port, which is /dev/ttymxc2, and the serial port file of UART3 is /dev/ttymxc2, so the serial port setting should be set to /dev/ ttymxc2. . The setting method is to press 'A' on the keyboard, and then enter "/dev/ttymxc2", as shown in the figure.

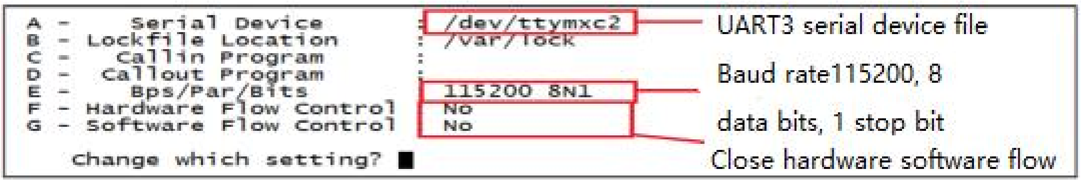


```
A -    Serial Device      : /dev/ttymxc2
B - Lockfile Location     : /var/lock
C -   Callin Program      :
D -  Callout Program      :
E -    Bps/Par/Bits       : 115200 8N1
F - Hardware Flow Control : Yes
G - Software Flow Control : No

   Change which setting?
```

UART3 serial device file

/dev/ttymxc2

After setting, press the Enter key to confirm. After confirming, you can set other configuration items. For example, E sets the baud rate, data bits and stop bits, and F sets the hardware flow control, the setting methods are the same, as shown in the figure after setting.



```
A -    Serial Device      : /dev/ttymxc2
B - Lockfile Location     : /var/lock
C -   Callin Program      :
D -  Callout Program      :
E -    Bps/Par/Bits       : 115200 8N1
F - Hardware Flow Control : No
G - Software Flow Control : No

   Change which setting?
```

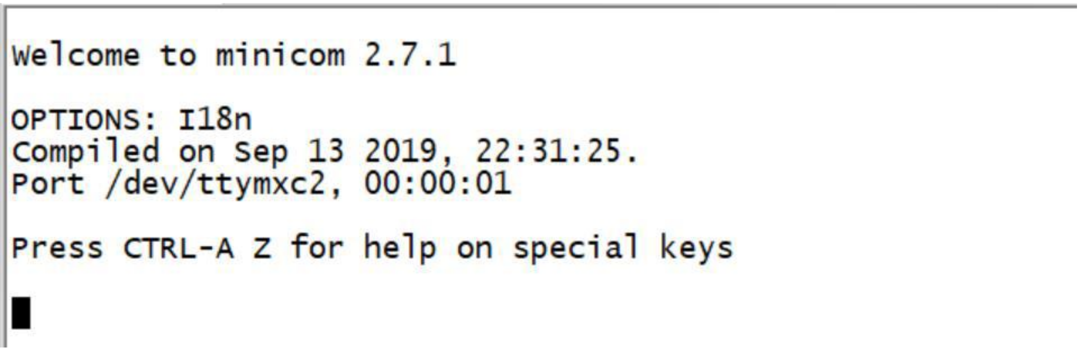
UART3 serial device file

Baud rate115200, 8

data bits, 1 stop bit

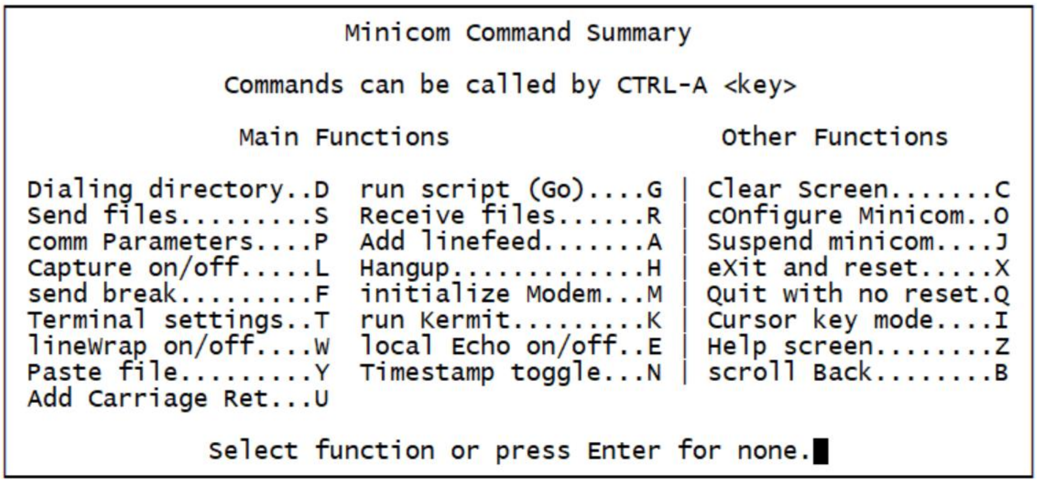
Close hardware software flow

After all settings are completed, press the Enter key to confirm and exit. At this time, it will return to the interface of the setting menu. Press the ESC key to exit the interface of the setting menu. After exiting, it will be as shown in the figure.



```
Welcome to minicom 2.7.1

OPTIONS: I18n
Compiled on Sep 13 2019, 22:31:25.
Port /dev/ttymxc2, 00:00:01

Press CTRL-A Z for help on special keys

█
```

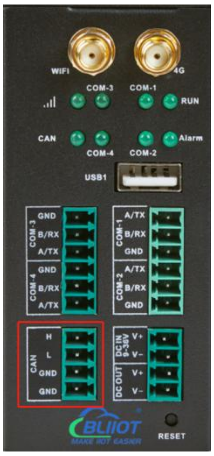
This is the serial port debugging interface. It can be seen that the current serial port file is /dev/ttymxc2, press CTRL-A, and then press Z to open the minicom help information interface, as shown in the figure.



```
                     Minicom Command Summary

              Commands can be called by CTRL-A <key>

               Main Functions                  Other Functions

 Dialing directory..D  run script (Go)....G | Clear Screen.......C
 Send files.........S  Receive files......R | cOnfigure Minicom..O
 comm Parameters....P  Add linefeed.......A | Suspend minicom....J
 Capture on/off.....L  Hangup.............H | eXit and reset.....X
 send break.........F  initialize Modem...M | Quit with no reset.Q
 Terminal settings..T  run Kermit.........K | Cursor key mode....I
 lineWrap on/off....W  local Echo on/off..E | Help screen........Z
 Paste file.........Y  Timestamp toggle...N | scroll Back........B
 Add Carriage Ret...U

             Select function or press Enter for none.█
```

The minicom has many shortcut keys. In this experiment, we enable the echo function of minicom. The echo function configuration item is "local Echo on/off..E", press E to turn on/off the echo display function.

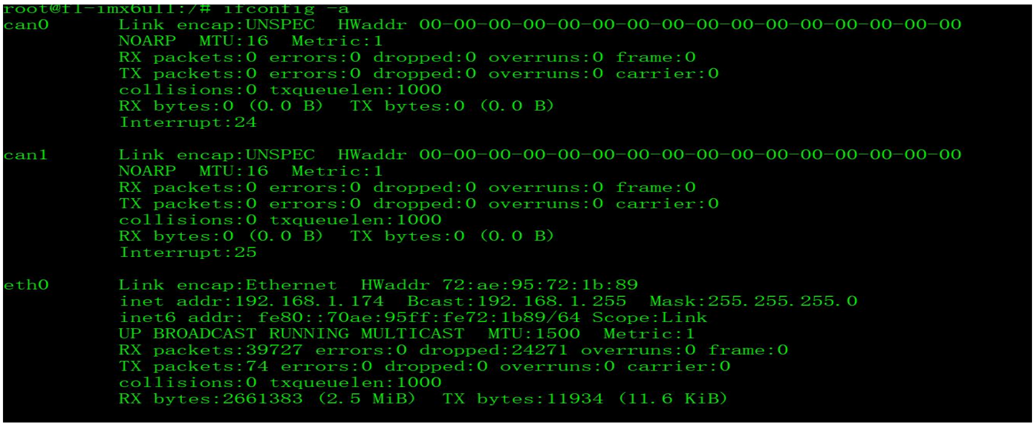
3.4 CAN Interface



The CAN interface is as shown in the figure, enter the following command:

ifconfig -a //View all network cards

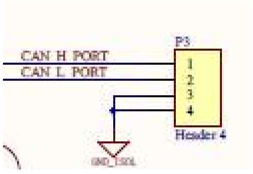
If the FlexCAN driver works well, you will see the network card interface corresponding to CAN, as shown in the figure, there is a network card named "can0", which is the CAN network card corresponding to the CAN1 interface on the BL302 board.



```
root@fl-imx6ull:/# ifconfig -a
can0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          NOARP  MTU:16  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:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:24

can1      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          NOARP  MTU:16  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:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:25

eth0      Link encap:Ethernet  HWaddr 72:ae:95:72:1b:89
          inet addr:192.168.1.174  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::70ae:95ff:fe72:1b89/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:39727 errors:0 dropped:24271 overruns:0 frame:0
          TX packets:74 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2661383 (2.5 MiB)  TX bytes:11934 (11.6 KiB)
```

There is a CAN interface on the BL302 board. If you need to test the CAN interface, you need a CAN device. You can use another BL302 board or a board with CAN for testing.

Prepare two BL302 devices, and then connect the CAN interfaces, the CAN terminals on the BL302 devices are as shown in the figure



Connect the CAN interfaces of the two devices. Note that CAN H is connected to CAN_H, and CAN_L is connected to CAN_L.

Firstly, use the IP command to set the CAN interface of the two devices, set the speed of the CAN interface, and enter the following command:

ip link set can0 type can bitrate 500000

The above command sets the speed of can0 to 500Kbit/S, and the speed of the two CAN devices should be set to the same. When the speed setting is complete, open the can0 network card, the command is as follows:

ifconfig can0 up//open can0

When can0 is opened, you can use the small tools in can-utils to perform data sending and receiving tests. One device is used to receive data, and the other is used to send data. The device that receives data uses the candump command and enters the following command:

candump can0 //receive data

The device sending data uses the cansend command to send 8 bytes of data to the receiving unit:

0X11, 0X22, 0X33, 0X44, 0X55, 0X66, 0X77, 0X88. Enter the following command:

```
cansend can0 5A1#11.22.33.44.55.66.77.88
```

The cansend command is used to send can data, "5A1" is the frame ID, "11.22.33.44.55.66.77.88" behind the "#" is the data to be sent, in hexadecimal. CAN2.0 can send up to 8 bytes of data at a time, and the 8 bytes of data are separated by ".".

If CAN works well, the receiving end will receive the 8 bytes of data sent above.

```
/ # candump can0
can0    5A1    [8] 11 22 33 44 55 66 77 88
```

The can0 interface of the receiving end has received 8 bytes of data, and the frame ID is 5A1, indicating that the CAN driver is working well.

If you want to close can0, enter the following command:

```
ifconfig can0 down
```

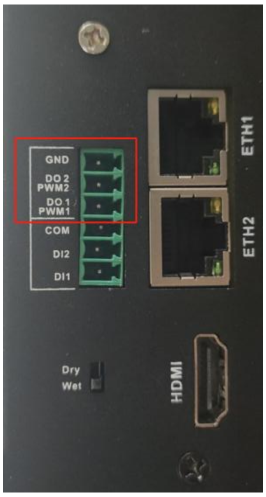
If you want to test CAN loopback on a board, set CAN as follows:

```
ifconfig can0 down //If can0 is already open, close it first
ip link set can0 type can bitrate 500000 loopback on //Enable loopback test
ifconfig can0 up //reopen can0
candump can0 & //candump receives data in the background
cansend can0 5A1#11.22.33.44.55.66.77.88 //cansend send data
```

If the loopback test is successful, the device will receive the data sent to itself, as shown in the figure:

```
/ # cansend can0 5A1#11.22.33.44.55.66.77.88
can0    5A1    [8] 11 22 33 44 55 66 77 88
```

3.5 PWM Interface

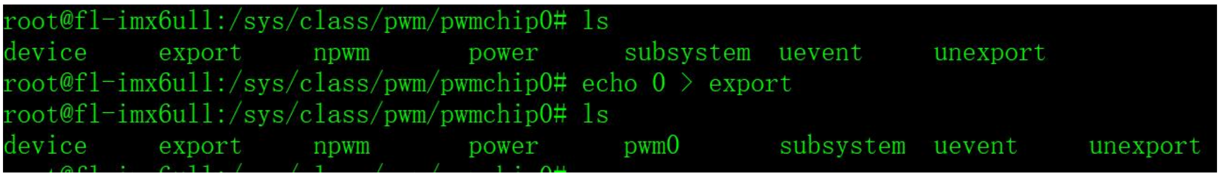


The PWM port is shown in the figure. PWM devices are under the directory /sys/class/pwm, where PWM1 and PWM2 are applied to pwmchip0 and pwmchip1.

Taking PWM1 as an example, first you need to call out the pwm0 directory under pwmchip0, and enter the following command:

echo 0 > /sys/class/pwm/pwmchip0/export

After the execution is completed, a subdirectory named "pwm0" will be generated under the pwmchip0 directory, as shown in the figure



```
root@fl-imx6ull:/sys/class/pwm/pwmchip0# ls
device     export     npwm       power      subsystem  uevent     unexport
root@fl-imx6ull:/sys/class/pwm/pwmchip0# echo 0 > export
root@fl-imx6ull:/sys/class/pwm/pwmchip0# ls
device     export     npwm       power      pwm0       subsystem  uevent     unexport
```

Enable PWM1: Enter the following command to enable PWM1

echo 1 > /sys/class/pwm/pwmchip0/pwm0/enable

Set the frequency of PWM1: Note that the period value is set here, and the unit is ns. For example, the period of 20KHz frequency is 50000ns. Enter the following command:

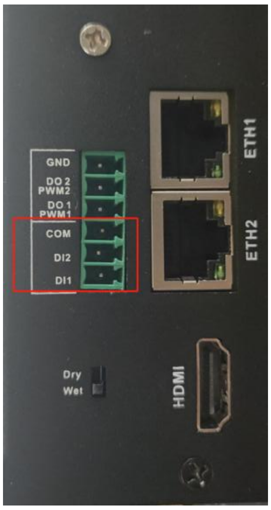
echo 50000 > /sys/class/pwm/pwmchip0/pwm0/period

Set the duty cycle of PWM1: You cannot set the duty cycle directly, please set the ON time of a cycle, that is, the high-level time, for example, the ON time of 20% duty cycle at 20KHz frequency is 10000, enter the following command

echo 10000 > /sys/class/pwm/pwmchip0/pwm0/duty_cycle

If you need to adjust the frequency or duty cycle, pay attention to the high-level time when adjusting.

3.6 DI

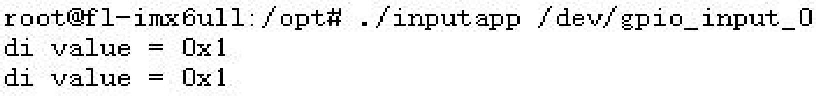


The devices corresponding to DI1 and DI2 are gpio_input_0 and gpio_input_1 respectively. Select the dry/wet contact mode by the switch below.

Take DI1 as an example, when debugging the DI interface, enter the command:

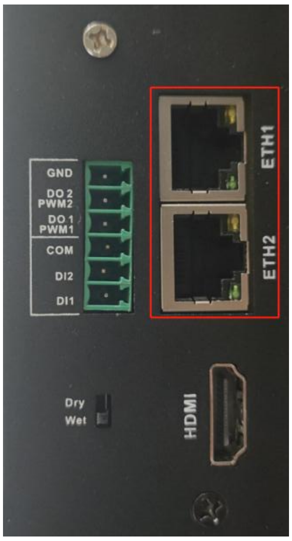
./inputapp /dev/gpio_input_0

When the dry contact mode is selected, the port is shorted; when the wet contact mode is selected, the input is greater than 3V; the console output is as follows:



```
root@fl-imx6ull:/opt# ./inputapp /dev/gpio_input_0
di value = 0x1
di value = 0x1
```

3.7 LAN



There are eth0 and eth1 network cards on the bottom board of BL302. When the power is just turned on and the startup is complete, if the network cable is not inserted, you can see that the network port has no IP address with ifconfig. This is because dhcp dynamically allocates ip, when network cable is inserted, the console will print the corresponding Ethernet, and you can check the corresponding IP address with ifconfig.

Backplane screen printing	Software equipment
NET1	eth1
NET2	eth0

Note: eth1 and eth0 cannot be used in the same LAN.

Take eth0 as an example.

Under the Linux system, use the ifconfig command to display or configure network devices, and use ethtool to query and set network card parameters.

Set the IP address and view the details of the current network card:

```
root@fl-imx6ull:~# ifconfig eth0 192.168.1.120 //set ip
root@fl-imx6ull:~# ifconfig eth0 //Check the network status after setting
eth0 Link encap:Ethernet HWaddr 3A:D9:93:8E:A8:A4
inet addr:192.168.1.120 Bcast:192.168.1.255 Mask:255.255.255.0
inet6 addr: fe80::38d9:93ff:fe8e:a8a4%2124311408/64 Scope:Link
inet6 addr: fec0::38d9:93ff:fe8e:a8a4%2124311408/64 Scope:Site
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:28 errors:0 dropped:0 overruns:0 frame:0
TX packets:63 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:11550 (11.2 KiB) TX bytes:11579 (11.3 KiB)
```

inet addr:192.168.1.120 indicates IP setting successful

If your device is connected to a router, and the router supports DHCP automatic IP address assignment, you can enter the command in the HyperTerminal:

```
root@fl-imx6ull:~# udhcpc -i eth0
udhcpc (v1.24.1) started
Sending discover...
Sending select for 192.168.20.101...
Lease of 192.168.20.101 obtained, lease time 86400
/etc/udhcpc.d/50default: Adding DNS 222.222.222.222
```

It is used to dynamically obtain the IP address. The "-i" parameter is used to specify the name of the network card. The name of the network card of the wired network is eth0.

The dns server information in the /etc/resolv.conf file will be added automatically.

Modify mac address:

```
root@fl-imx6ull:~# ifconfig eth0 hw ether 00:00:00:00:00:01
root@fl-imx6ull:~# ifconfig eth0
eth0 Link encap:Ethernet HWaddr 00:00:00:00:00:01
inet addr:192.168.20.101 Bcast:192.168.20.255 Mask:255.255.255.0
inet6 addr: fec0::38d9:93ff:fe8e:a8a4%2128292720/64 Scope:Site
inet6 addr: fec0::200:ff:fe00:1%2128292720/64 Scope:Site
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:85 errors:0 dropped:0 overruns:0 frame:0
TX packets:118 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:22942 (22.4 KiB) TX bytes:22259 (21.7 KiB)
```

Set subnet mask:

```
root@fl-imx6ull:~# ifconfig eth0 netmask 255.255.255.0 //set eth0 subnet mask 255.255.255.0
root@fl-imx6ull:~# ifconfig eth0
eth0 Link encap:Ethernet HWaddr 00:00:00:00:00:01
inet addr:192.168.20.101 Bcast:192.168.20.255 Mask:255.255.255.0
inet6 addr: fec0::38d9:93ff:fe8e:a8a4%2128915312/64 Scope:Site
inet6 addr: fec0::200:ff:fe00:1%2128915312/64 Scope:Site
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:107 errors:0 dropped:0 overruns:0 frame:0
TX packets:118 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:25700 (25.0 KiB) TX bytes:22259 (21.7 KiB)
```

Set broadcast address

```
root@fl-imx6ull:~# ifconfig eth0 broadcast 192.168.1.255//eth0 broadcast address 192.168.1.255
root@fl-imx6ull:~# ifconfig eth0
```

The print information is as follows:

```
eth0 Link encap:Ethernet HWaddr 00:00:00:00:00:01
inet addr:192.168.20.101 Bcast:192.168.1.255 Mask:255.255.255.0
inet6 addr: fec0::38d9:93ff:fe8e:a8a4%2123332464/64 Scope:Site
inet6 addr: fec0::200:ff:fe00:1%2123332464/64 Scope:Site
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:111 errors:0 dropped:0 overruns:0 frame:0
TX packets:132 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:26130 (25.5 KiB) TX bytes:25947 (25.3 KiB)
```

Bcast:192.168.1.255 indicates broadcast address setting is successful

Add default gateway:

```
root@fl-imx6ull:~# route add default gw 192.168.20.1
```

Delete the default gateway:

```
root@fl-imx6ull:~# route del default gw 192.168.20.1
```

Close the eth0 network card:

```
root@fl-imx6ull:~# ifconfig eth0 down
```

Open the eth0 network card:

```
root@fl-imx6ull:~# ifconfig eth0 up
fec 20b4000.ethernet eth0: Freescale FEC PHY driver [Micrel KSZ8081 or KSZ8091]
(mii_bus:phy_addr=20b4000.ethernet:01, irq=-1)
root@fl-imx6ull:~# fec 20b4000.ethernet eth0: Link is Up - 100Mbps/Full - flow control rx/tx
```

Ping test

First, make sure network cable works well, connect the device and the host or virtual machine with a network cable, and set the device and the host or virtual machine on the same network segment. For example. The IP address of my device is 192.168.1.174, and the IP address of my virtual machine is 192.168.1.141, and the ping command can be used to ping.

3.8 WiFi Module

The WiFi module is PCIe interface, supports 2.4G frequency, it is compatible with 8821cu, 8723du and 8822BU three kinds of WiFi drivers, and the default router adopts wpa encryption.

After connecting the module and powering on the device, enter the command line. You can check the USB status through the lsusb command as follows

```
root@fl-imx6ull:/opt# ./app /dev/gpiopci 1          //Power on the WIFI module，
root@fl-imx6ull:~# lsmod
Module Size Used by
mx6s_capture 14876 0
8723du 1313893 0       //WiFi is automatically loaded, and 8723du has been loaded successfully
ov9650_camera 12446 0
```

3.8.1 STA Mode

STA mode is to connect to the wireless network as a station, the operation method is as follows:

-i indicates the WiFi model; -s indicates the name of the WiFi hotspot; -p indicates the password, if there is no password, enter -p NONE; the router uses wpa encryption, and the specific operation instructions can be found in the wifi.sh script

```
root@fl-imx6ull:~# wifi.sh -i wlan0 -s beilai -p xxx //Execute the test script
```

Print information as below

```
wifi 8723du
ssid beilai
pasw xxx
usbcore: deregistering interface driver rtl8723du
usbcore: registered new interface driver rtl8723du
IPv6: ADDRCONF(NETDEV_UP): wlan0: link is not ready
Successfully initialized wpa_supplicant
rfkill: Cannot open RFKILL control device
udhcpc (v1.24.1) started
Sending discover...
wlan0: CTRL-EVENT-REGDOM-CHANGE init=BEACON_HINT type=UNKNOWN
wlan0: Trying to associate with 04:d7:a5:f9:26:1d (SSID='beilai' freq=2427 MHz)
```

```
    wlan0: Associated with 04:d7:a5:f9:26:1d
    IPv6: ADDRCONF(NETDEV_CHANGE): wlan0: link becomes ready
    wlan0: WPA: Key negotiation completed with 04:d7:a5:f9:26:1d [PTK=CCMP GTK=TKIP]
    wlan0: CTRL-EVENT-CONNECTED - Connection to 04:d7:a5:f9:26:1d completed [id=0 id_str=]
    nf_conntrack: automatic helper assignment is deprecated and it will be removed soon. Use the iptables CT
target to attach helpers instead.
    Sending discover...
    Sending select for 192.168.5.186...
    Lease of 192.168.5.186 obtained, lease time 1800
    /etc/udhcpc.d/50default: Adding DNS 222.222.202.202
    /etc/udhcpc.d/50default: Adding DNS 222.222.222.222
    WLAN Finshed!
```

After the script runs, it can automatically assign IP and generate DNS, and the WiFi connection is successful.

To ping IP or domain name, the command is as follows:

```
root@fl-imx6ull:~# ping -I 192.168.1.118 www.baidu.com
```

Print information as below

```
     ping -I 192.168.1.118 www.baidu.com : 56 data bytes
    64 bytes from 192.168.1.118: seq=0 ttl=128 time=39.783 ms
    64 bytes from 192.168.1.118: seq=1 ttl=128 time=81.529 ms
    64 bytes from 192.168.1.118: seq=2 ttl=128 time=15.236 ms
    64 bytes from 192.168.1.118: seq=3 ttl=128 time=12.076 ms
    64 bytes from 192.168.1.118: seq=4 ttl=128 time=16.300 ms
    --- 192.168.1.118 ping statistics ---
    5 packets transmitted, 5 packets received, 0% packet loss
    round-trip min/avg/max = 12.076/32.984/81.529 ms
```

Method of checking WiFi signal:

```
    root@fl-imx6ull:~# cat /proc/net/wireless | grep wlan0 | awk '{print $3}' //Get signal strength
    root@fl-imx6ull:~# cat /proc/net/wireless | grep wlan0 | awk '{print $4}' //Get the signal quality, in dBm
    root@fl-imx6ull:~# cat /proc/net/wireless | grep wlan0 | awk '{print $5}' //Network port background
noise, in dBm
```

3.8.2 AP Mode

In AP mode, the device can connect with maximum 8 users theoretically

For example, Ethernet eth0 connecting to the router. After configuring the Ethernet, you need to test whether eth0 can connect to the external network. If you can connect to the external network (refer to the "LAN" chapter for the method), please follow the steps. If not, please check whether the Ethernet or router connection is good.

Working in AP mode, mobile phones and other devices can directly connect to the module.

Set Ethernet IP, configure network firewall:

```
root@fl-imx6ull:~# udhcpc -i eth0 //Automatically assign IP, if the testing of eth0 network works, this
step is not required
root@fl-imx6ull:~# echo 1 > /proc/sys/net/ipv4/ip_forward //Turn on IP forwarding
root@fl-imx6ull:~# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE//Set forwarding rules
```

Set WiFi mode and IP

```
root@fl-imx6ull:~# ifconfig wlan0 up //open WiFi
root@fl-imx6ull:~# ifconfig wlan0 192.168.0.10 netmask 255.255.255.0 //Set IP and subnet mask
root@fl-imx6ull:~# ifconfig wlan0 promisc //Set wlan0 to promiscuous mode
```

Enable AP

```
root@fl-imx6ull:~# udhcpd /etc/udhcpd.conf & //Configuration information such as WiFi address and
gateway
root@fl-imx6ull:~# hostapd -d /etc/hostapd/hostapd.conf & //Encryption method, user name, password
and other settings
```

In the hostapd.conf file: ssid is user name, wpa_passphrase is password; mobile phones can connect to the AP hotspot of the device through WiFi, and the device uses the following user name and password by default: Hotspot name: beilaitest Password: 1234567890

Unload modules that have been added to the kernel:

```
root@fl-imx6ull:~# rmmod 8723du
usbcore: deregistering interface driver rtl8723du
wlan0: CTRL-EVENT-DISCONNECTED bssid=04:d7:a5:f9:26:1d reason=0
```

3.9 4G/5G

The 4G/5G module is PCIE interface. Taking the 4G module as an example, please confirm the firmware version of the module when using IoT card for testing. Low version firmware does not suppot it, the firmware needs to upgrade to EC20. After connecting the module and powering on the board, enter the command line, and you can check the USB status through the lsusb command as follows

```
root@fl-imx6ull:/opt# ./app /dev/gpiopci 1          //Power on the 4G module, and look at the 4 virtual
devices. The device is powered on by default, and writing 0 means power off.
root@fl-imx6ull:/opt# random: nonblocking pool is initialized
usb 2-1: new high-speed USB device number 2 using ci_hdrc
option 2-1:1.0: GSM modem (1-port) converter detected
usb 2-1: GSM modem (1-port) converter now attached to ttyUSB0
```

```
option 2-1:1.1: GSM modem (1-port) converter detected
usb 2-1: GSM modem (1-port) converter now attached to ttyUSB1
option 2-1:1.2: GSM modem (1-port) converter detected
usb 2-1: GSM modem (1-port) converter now attached to ttyUSB2
option 2-1:1.3: GSM modem (1-port) converter detected
usb 2-1: GSM modem (1-port) converter now attached to ttyUSB3
GobiNet 2-1:1.4 eth2: kevent 12 may have been dropped
GobiNet 2-1:1.4 eth2: register 'GobiNet' at usb-ci_hdrc.1-1, GobiNet Ethernet Device, 9e:33:27:a1:5a:2c
creating qcqmi2
GobiNet 2-1:1.4 eth2: kevent 12 may have been dropped
IPv6: ADDRCONF(NETDEV_UP): eth2: link is not ready
root@imx6ulevk:~# lsusb
Bus 001 Device 004: ID 0bda:b720
Bus 001 Device 005: ID 2c7c:0125                    //EC20 VID and PID
Bus 001 Device 002: ID 0424:2514
Bus 001 Device 001: ID 1d6b:0002
```

/dev Check the device node status

```
root@fl-imx6ull:/opt# ls /dev/ttyUSB*
/dev/ttyUSB0    /dev/ttyUSB1    /dev/ttyUSB2    /dev/ttyUSB3
```

EC20 dial-up: Sometimes Sending discover... will appear several times, which is caused by poor signal.

```
    root@fl-imx6ull:~# quectel-CM &
    [1] 598
    root@fl-imx6ull:/forlinx/cmdbin#[04-26_19:16:06:781]
WCDMA&LTE_QConnectManager_Linux&Android_V1.1.34
    [04-26_19:16:06:783] ./quectel-CM profile[1] = (null)/(null)/(null)/0, pincode = (null)
    [04-26_19:16:06:790] Find /sys/bus/usb/devices/1-1.1 idVendor=2c7c idProduct=0125
    [04-26_19:16:06:791] Find /sys/bus/usb/devices/1-1.1:1.4/net/eth2
    [04-26_19:16:06:791] Find usbnet_adapter = eth2
    [04-26_19:16:06:792] Find /sys/bus/usb/devices/1-1.1:1.4/GobiQMI/qcqmi2
    [04-26_19:16:06:792] Find qmichannel = /dev/qcqmi2
    [04-26_19:16:06:851] Get clientWDS = 7
    [04-26_19:16:06:882] Get clientDMS = 8
    [04-26_19:16:06:914] Get clientNAS = 9
    [04-26_19:16:06:946] Get clientUIM = 10
    [04-26_19:16:06:978] Get clientWDA = 11
    [04-26_19:16:07:011] requestBaseBandVersion EC20CEHCLGR06A05M1G
    //If the version number in the printed information is 5Mxx, that means it supports the IoT card, if it is
2Mxx, then it does not support IoT card
```

```
[04-26_19:16:07:106] requestGetSIMStatus SIMStatus: SIM_READY
[04-26_19:16:07:138] requestGetProfile[1] ctnet///0
[04-26_19:16:07:171] requestRegistrationState2 MCC: 460, MNC: 11, PS: Attached, DataCap: LTE
[04-26_19:16:07:202] requestQueryDataCall IPv4ConnectionStatus: DISCONNECTED
[04-26_19:16:07:266] requestRegistrationState2 MCC: 460, MNC: 11, PS: Attached, DataCap: LTE
[04-26_19:16:07:300] requestSetupDataCall WdsConnectionIPv4Handle: 0xe1645ec0
[04-26_19:16:07:394] requestQueryDataCall IPv4ConnectionStatus: CONNECTED
[04-26_19:16:07:427] ifconfig eth2 up
[04-26_19:16:07:471] busybox udhcpc -f -n -q -t 5 -i eth2
[04-26_19:16:07:506] udhcpc (v1.24.1) started
[04-26_19:16:07:631] Sending discover...
[04-26_19:16:07:691] Sending select for 172.29.86.131...
[04-26_19:16:07:751] Lease of 172.29.86.131 obtained, lease time 7200
[04-26_19:16:07:869] /etc/udhcpc.d/50default: Adding DNS 222.222.222.222
[04-26_19:16:07:869] /etc/udhcpc.d/50default: Adding DNS 222.222.202.202
```

After the connection is successful, ping Baidu to test:

```
root@fl-imx6ull:~# ping www.baidu.com
PING www.baidu.com (220.181.38.150): 56 data bytes
64 bytes from 220.181.38.150: seq=0 ttl=53 time=137.243 ms
64 bytes from 220.181.38.150: seq=1 ttl=53 time=51.239 ms
64 bytes from 220.181.38.150: seq=2 ttl=53 time=94.440 ms
```

The 4G module has two SIM card slots, select SIM1 or SIM2 by setting /dev/gpiosgm, after the system starts, the default is SIM1, write 1 to the /dev/gpiosgm device, and select SIM2. After replacing the card slot, please power on the module again.

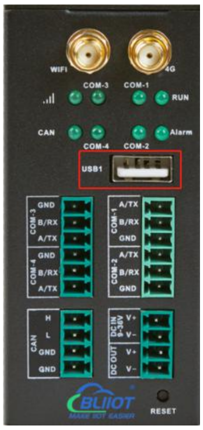
```
root@fl-imx6ull:/opt# ./app /dev/gpiosgm 1
```

If both wired network and 4G network are used at the same time, one of them will be unavailable due to gateway priority. If you need to change it, you can enter the following command to view the current default gateway information.

Change the gateway information.

```
ip route show
sudo route add default gw 172.29.86.131    //add 4G module gateway, gw followed by IP address
sudo route del default gw 172.29.86.0    //delete 4G module gateway, gw followed by IP address
```

3.10 USB Port



Support hot swapping of USB mouse, USB keyboard, and U disk devices.

When using a USB flash drive, it is recommended to use a formatting tool to format it into a FAT32 format that can be recognized by the linux system. The mounted directory of the U disk is /run/media, insert the U disk, and the following information is displayed:

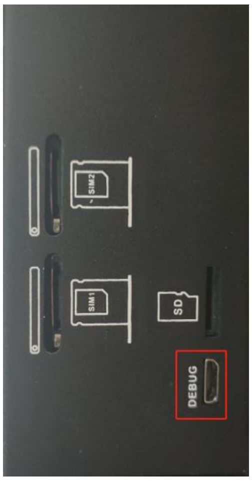
```
root@fl-imx6ull:~# usb 1-1.3: new high-speed USB device number 5 using ci_hdrc
usb-storage 1-1.3:1.0: USB Mass Storage device detected
scsi host1: usb-storage 1-1.3:1.0
scsi 1:0:0:0: Direct-Access Generic MassStorageClass 1536 PQ: 0 ANSI: 6
sd 1:0:0:0: [sda] 31116288 512-byte logical blocks: (7.94 GB/7.40 GiB)
sd 1:0:0:0: [sda] Write Protect is off
sd 1:0:0:0: [sda] Write cache: disabled, read cache: enabled, doesn't support DPO or FUA
sda: sda1 //the mount device name is sda1
sd 1:0:0:0: [sda] Attached SCSI removable disk
FAT-fs (sda1): Volume was not properly unmounted. Some data may be corrupt. Please run fsck.
```

Check the usb storage device, /run/media is the mounting directory of the U disk, and the device name after the U disk is mounted is sda1.

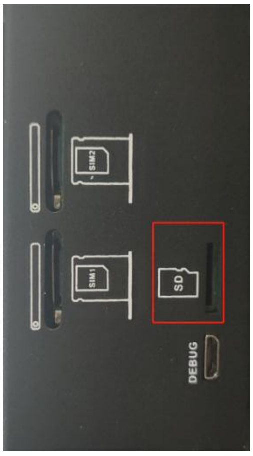
If you don’t need to use the U disk anymore, please use umount to uninstall the U disk before unplug it.

```
root@fl-imx6ull:~# umount /run/media/sda1
```

3.11 Debug



3.12 SD Card slot



This device does not support NTFS and exFAT format file systems. If you do not know the SD card format, please format it into FAT32 format before use.

The SD card mount directory is /run/media, supports hot swapping, and the terminal will print information about the SD card. Different SD cards may display different information. After the SD card is inserted into the SD card slot of the device, the system will automatically check and mount the SD card. After the mount is successful, the SD card can be read and written.

Plug in the 32G SD card, after mounting, you can see the device name after the SD card is mounted from the print information. The print information is as follows:

```
root@fl-imx6ull:/# mmc0: host does not support reading read-only switch, assuming write-enable
mmc0: new high speed SDHC card at address 59b4
mmcblk0: mmc0:59b4 SD32G 29.1 GiB
Mmcblk0: p1
FAT-fs (mmcblk0p1): Volume was not properly unmounted. Some data may be corrupt. Please run fsck.
```

/run/media is the mounting directory of the SD card, and view the files in this directory

```
root@fl-imx6ull:~# ls /run/media                    //list file under /run/media directory
```

The printed information is as follows, mmcblk0p1 is the file name after the SD card is mounted

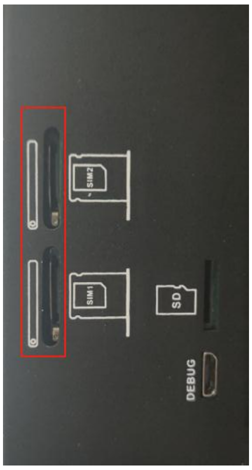
```
mmcblk0p1 mmcblk1p1
```

If you don't need to use the SD card anymore, please use umount to uninstall the SD card before removing the SD card.

```
root@fl-imx6ull:~# umount /run/media/mmcblk0p1
```

Note: First exiting the SD card mounting path, then insert and remove the SD card.

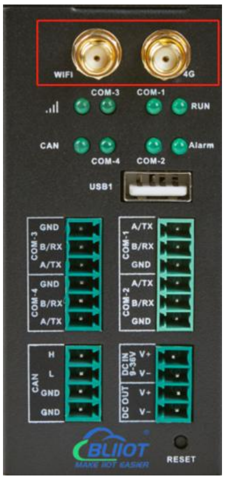
3.13 SIM Card Slot



When inserting/removing the SIM card, please make sure the device is turned off

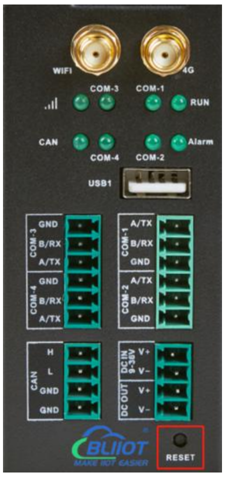
Note: Please place the device flat when inserting/removing the SIM card.

3.14 Antenna Interface



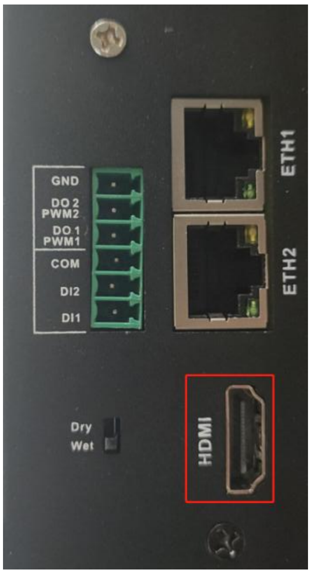
The left is the WIFI antenna interface, and the right is the 4G antenna interface.

3.15 Reset Button



After the device is running normally, press the reset button and the device will automatically restart.

3.16 HDMI



The HDMI version is HDMI1.3, the resolution is 1600x900, 50Hz. Please connect to HDMI first and then power on.

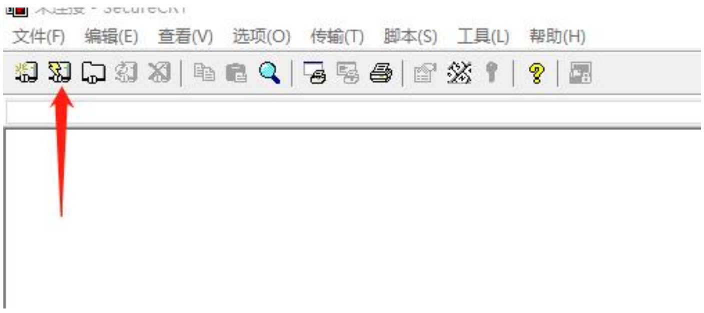
4 Software

4.1 Login

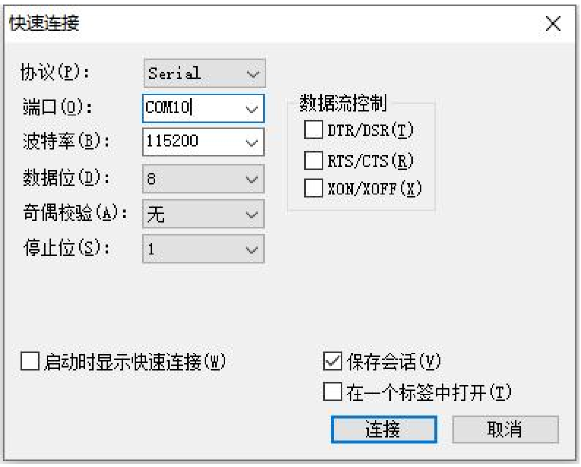
Serial Port Login method:

The device can be logged in via micro-USB, and the default system login name is: root. Take

SecureCRT as an example, connect the power supply and USB cable of the device, open SecureCRT, and click the quick login button in the upper left corner.



Select Serial protocol, select the port, and set the baud rate to 115200.



快速连接

协议(P): Serial

端口(O): COM10

波特率(B): 115200

数据位(D): 8

奇偶校验(A): 无

停止位(S): 1

数据流控制

☐ DTR/DSR(T)

☐ RTS/CTS(R)

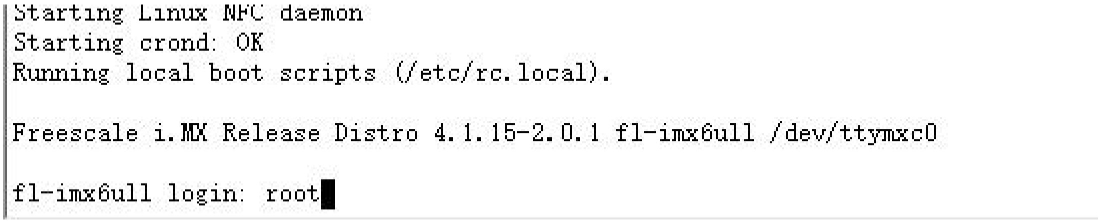
☐ XON/XOFF(X)

☐ 启动时显示快速连接(W)

☑ 保存会话(V)

☐ 在一个标签中打开(T)

Enter the login name root to log in.



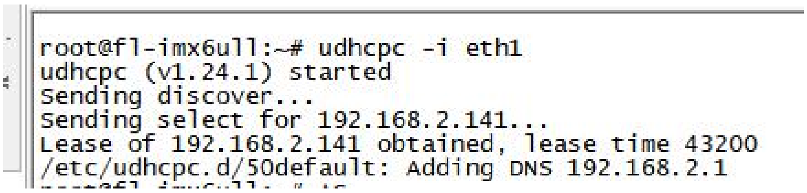
```
Starting Linux NFC daemon
Starting crond: OK
Running local boot scripts (/etc/rc.local).

Freescale i.MX Release Distro 4.1.15-2.0.1 fl-imx6ull /dev/ttymxc0

fl-imx6ull login: root
```

Ethernet Login Method

Make sure the network is working before using this method.

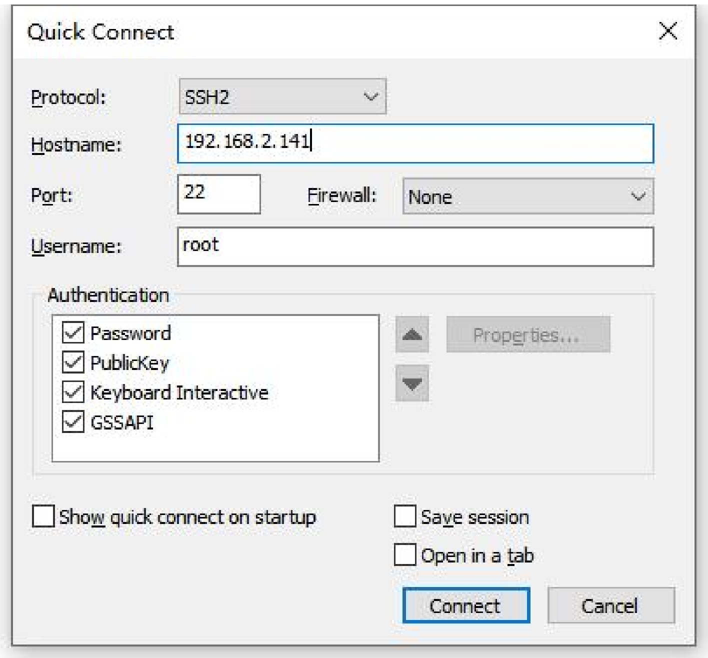
Default IP: eth0:192.168.0.232 eth1:192.168.2.232

Modify the default IP setting: Just modify the corresponding IP under /etc/rc5.d/S99autorun.sh file.



```
root@fl-imx6ull:~# udhcpc -i eth1
udhcpc (v1.24.1) started
Sending discover...
Sending select for 192.168.2.141...
Lease of 192.168.2.141 obtained, lease time 43200
/etc/udhcpc.d/50default: Adding DNS 192.168.2.1
```

Hostname fills in the device IP, Username is root.


Quick Connect

Protocol: SSH2

Hostname: 192.168.2.141

Port: 22 Firewall: None

Username: root

Authentication

☑ Password
☑ PublicKey
☑ Keyboard Interactive
☑ GSSAPI

□ Show quick connect on startup □ Save session
□ Open in a tab

Click Connect to enter the device.

4.2 Time Setting

By using the date and hwclock tools to set the software and hardware time, test whether the software clock reads the RTC clock synchronously when the board is powered off and on again.

```
root@fl-imx6ull:~# date -u 031912002020.00 //Set software time
Thu Mar 19 12:00:00 UTC 2020
root@fl-imx6ull:~# hwclock -r //Show hardware time
Fri May 3 17:50:51 2019 0.000000 seconds
root@fl-imx6ull:~# hwclock -w //Synchronize software time to hardware time
Fri May 3 17:50:51 2019 0.000000 seconds
```

Power off the board and power it on again. After entering the system, use the command date to read the system time, and the time has been synchronized.

4.3 MCU Frequency Modulation

When the user needs to modify the MCU frequency, BL302 supports adjusting the MCU frequency by command. All cpufreq governor types supported in the current kernel:

```
root@fl-imx6ull:~# cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_available_governors
interactive conservative userspace powersave ondemand performance
```

userspace represents the user mode, in which other user programs are allowed to adjust the CPU frequency. View the frequency gear supported by the current CPU:

```
root@fl-imx6ull:~# cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_available_frequencies
198000 396000 528000 792000
```

Modify it to user mode, modify the frequency to 792000:

```
root@fl-imx6ull:~# echo userspace > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
root@fl-imx6ull:~# echo 792000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_setspeed
```

Check the current frequency:

```
root@fl-imx6ull:~# cat /sys/devices/system/cpu/cpu0/cpufreq/cpuinfo_cur_freq
 792000
```

4.4 Temperature Control

In the default setting of the kernel, the CPU junction temperature, if it exceeds 85 degrees, the CPU will reduce the frequency; if it exceeds 105 degrees, the CPU will restart;

View the current CPU temperature value:

```
root@fl-imx6ull:~# cat /sys/class/thermal/thermal_zone0/temp
 51890 //temperature is 51.890℃（51890/1000）
```

View the CPU frequency reduction temperature value in the kernel

```
root@fl-imx6ull:~# cat /sys/devices/virtual/thermal/thermal_zone0/trip_point_0_temp
 85000 //temperature is 85℃
```

View the CPU restart temperature value in the kernel

```
root@fl-imx6ull:~# cat /sys/devices/virtual/thermal/thermal_zone0/trip_point_1_temp
 105000 //temperature is 105℃
```

4.5 Wake From Sleep

There are three modes of sleep:

freeze: Freeze I/O devices, put them in a low-power state, make the processor enter an idle state, wake up the fastest, and consume more power than other methods. When only connected to the serial cable, the power supply is 5v, and the current is about 0.112A.

standby: In this state, the CPU is in a low power consumption state, and no data is saved to RAM or disk, and the standby and recovery of this state are usually very fast. 5v power supply only connected to the serial cable, the current is about 0.085A.

mem: Suspend to the memory, the computer stores the current running status and other data in the memory, turns off the hard disk, peripherals and other devices, and enters the waiting state. At this time, the memory still needs power to maintain its data, but the whole machine keeps low power consumption level. When resuming, the computer reads the data from the memory and returns to the state before the suspend, and the resuming speed is faster. When only connected to the serial cable, the power supply is 5v, and the current is about 0.076A.

cat /sys/power/state   check the supported modes

echo freeze > /sys/power/state Enter freeze mode

echo standby > /sys/power/state Enter standby mode

echo mem > /sys/power/state Enter mem mode

```
root@fl-imx6ull:~# echo mem > /sys/power/state

PM: Syncing filesystems ... done.

rtk_btusb: rtkbt_pm_notify: pm_event 3

rtk_btusb: rtkbt_pm_notify: suspend prepare

rtk_btusb: Remote wakeup not support, set intf->needs_binding = 1

Freezing user space processes ... (elapsed 0.002 seconds) done.

Freezing remaining freezable tasks ... (elapsed 0.001 seconds) done.

Suspending console(s) (use no_console_suspend to debug)
```

SNVS mode

1) Press and hold the on/off key (S4) for about 5s. At this time, most of the power supplies are turned off, and only the VDD_SNVS power supply is turned on.

2) Exit SNVS mode

Press and hold the on/off key for about 2 seconds, the power of the board is turned on, and the serial port restarts from uboot.

Wake up regularly through rtc

echo +15 > /sys/class/rtc/rtc1/wakealarm

15 seconds timing, you can set the time freely, it will take effect after the command is executed, rtc will time it separately, if it enters sleep after 15 seconds, it will not trigger wake-up.

4.6 Node-Red

If you need to use node-v18.12.1-linux-armv7l.tar.xz, you need to upgrade the lib library to 2.5, 2.6, 2.7; the default lib library of this machine is 2.3 (enter ldd --version to view the local glibc version). Take node-redV16.14.0 as an example, first copy the node-v16.14.0-linux-armv7l.tar.xz file to a directory of the device (or create a new one on the root directory).

```
root@fl-imx6ull:~# cp /run/media/sda1/node-v16.14.0-linux-armv7l.tar.xz    /test
```

Then use the tar xf command to decompress the file.

```
root@fl-imx6ull:~# tar xf node-v16.14.0-linux-armv7l.tar.xz
```

Then link node, npm, and npx in the file to /usr/bin.

```
root@fl-imx6ull:~# ln -sf /test/node-v16.14.0-linux-armv7l/bin/node /usr/bin

root@fl-imx6ull:~# ln -sf /test/node-v16.14.0-linux-armv7l/bin/npm /usr/bin

root@fl-imx6ull:~# ln -sf /test/node-v16.14.0-linux-armv7l/bin/npx /usr/bin
```

Connect to the network, enter the following command and wait for a few minutes to install node-red.

Installation should be done under node-v16.14.0-linux-armv7l/bin/.

```
root@fl-imx6ull:~# npm install -g --unsafe-perm node-red
```

If an error occurs that the certificate is invalid, you can enter the following command

npm set strict-ssl false

If you are stuck at timing idealTree:#root Completed in 75683ms without response, enter the following command to resolve it:

```
npm config set registry https://registry.npm.taobao.org

npm config get registry

npm install -g node-red
```

After the installation is successful, check whether the installation is successful and the corresponding version number node -v; npm -v.

After the node is installed successfully, you need a soft link to /usr/bin

```
root@fl-imx6ull:~#  ln  -sf  /test/node-v16.14.0-linux-armv7l/bin/node-red     /usr/bin
```

In this way, node-red can be executed in any directory;

```
root@fl-imx6ull:~# node-red
```

otherwise execute

node/test/node-v16.14.0-linux-armv7l/bin/node-red

If the execution fails, you need to operate npm uninstall, and then npm install.

After running node-red, open Google Chrome, enter http://(BL302 Internet accessible IP):1880; for example: http://192.168.2.232:1880, and enter the node-red interface.

4.7 SQLite

SQLite3 is a light embedded database, this device supports version V3.1~V3.4. Use less resources, with fast processing speed, and no need to install database server process. The device transplanted is sqlit3 version 3.11.0.

If you need to install other versions of SQLite3, you need to copy the corresponding version files to the /usr/ directory of the device, after decompression, enter the /usr/lib directory, and generate a link

```
ln -s libsqlite3.so.0.8.6 libsqlite3.so.0
ln -s libsqlite3.so.0.8.6 libsqlite3.so
```

Run the database:

```
root@fl-imx6ull:~# sqlite3
SQLite version 3.11.0 2016-02-15 17:29:24
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite>
```

Test the SQLite software:

```
 SQLite version 3.11.0 2016-02-15 17:29:24
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> create table tbl1 (one varchar(10), two smallint); //create table tbl1
sqlite> insert into tbl1 values('hello!',10); //Insert data into the table hello!|10
sqlite> insert into tbl1 values('goodbye', 20); //Insert data into the table goodbye|20
sqlite> select * from tbl1; //Query the content in table tbl1
hello!|10
goodbye|20
sqlite>
```

Exit the database:

```
sqlite> .exit //exit the database (or use the .quit command)
root@fl-imx6ull:~#
```

4.8 Python

This device supports Python V3.6~V3.10.

If you need to install it, copy the corresponding version of the file to the device, unzip it after the copy is complete (unzip it to the root directory), and set the environment variable (X represents the corresponding version, such as version 3.10, python3.10)

```
export PYTHONPATH=$PYTHONPATH:/lib/python3.X
export PYTHONHOME=$PYTHONHOME:/lib/python3.X
```

Enter python3.X to start

```
root@fl-imx6ull:~#export PYTHONPATH=$PYTHONPATH:/lib/python3.10
root@fl-imx6ull:~#export PYTHONHOME=$PYTHONHOME:/lib/python3.10
root@fl-imx6ull:/# python3.10
Python 3.10.8 (main, Nov 20 2022, 06:26:16) [GCC 5.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Enter quit() or type Ctrl+D to exit

4.9 QT

The computer supports QT version 4.8~5.15.

Get tslib first, git address: https://github.com/kergoth/tslib. After compiling, extract it to /usr/lib. Then pack the compiled qt file arm-qt into tar.bz2 format, and extract it to the /usr/lib/ directory. Edit /etc/profile and add the following to the end of the file. Note that the path should be your actual path.

```
export TSLIB_ROOT=/usr/lib/arm-tslib
export TSLIB_CONSOLEDEVICE=none
export TSLIB_FBDEVICE=/dev/fb0
export TSLIB_TSDEVICE=/dev/input/event1
export TSLIB_CONFFILE=$TSLIB_ROOT/etc/ts.conf
export TSLIB_PLUGINDIR=$TSLIB_ROOT/lib/ts
export TSLIB_CALIBFILE=/etc/pointercal
export QT_ROOT=/usr/lib/arm-qt
export QT_QPA_GENERIC_PLUGINS=tslib:/dev/input/event1
export QT_QPA_FONTDIR=/usr/share/fonts
export QT_QPA_PLATFORM_PLUGIN_PATH=$QT_ROOT/plugins
export QT_QPA_PLATFORM=linuxfb:tty=/dev/fb0
export QT_PLUGIN_PATH=$QT_ROOT/plugins
export LD_LIBRARY_PATH=$QT_ROOT/lib:$QT_ROOT/plugins/platforms
export QML2_IMPORT_PATH=$QT_ROOT/qml
export QT_QPA_FB_TSLIB=1
```

Then enter the enable environment variable:

```
source /etc/profile
```

Then you can run the QT program. If you want the Qt program to display Chinese, please put the Chinese font library under windows (path C:\Windows\Fonts) into a new /usr/share/fonts/ directory. If the routine uses characters, it will display that no fonts are found.

4.10 MySQL

The computer supports MySQL versions 5.1.51~5.1.73.

Copy the compiled file to the /usr/local/mysql directory of BL302, copy the executable file inside to the /usr/sbin directory or set the environment variable on the device

```
export PATH="$PATH:/usr/local/mysql/bin
```

Then add the //etc/my.conf configuration file. The content is as follows:

```
datadir=/var/lib/mysql
socket=/tmp/mysql.sock
user=root
#Default to using old password format for compatibility with mysql 3.x
#clients (those using the mysqlclient10 compatibility package).
old_passwords=1
[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

and then build

```
# mkdir /var/run/mysqld
# touch /var/run/mysqld/mysqld.pid
```

Install the database:

```
#mysql_install_db -u root
```

Start the MySQL service:

```
# mysqld_safe --user=root --skip-grant-tables --skip-networking &
# mysql
```

5 Firmware update

Please contact us if you need to upgrade firmware.

6 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

7 Technical Support

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