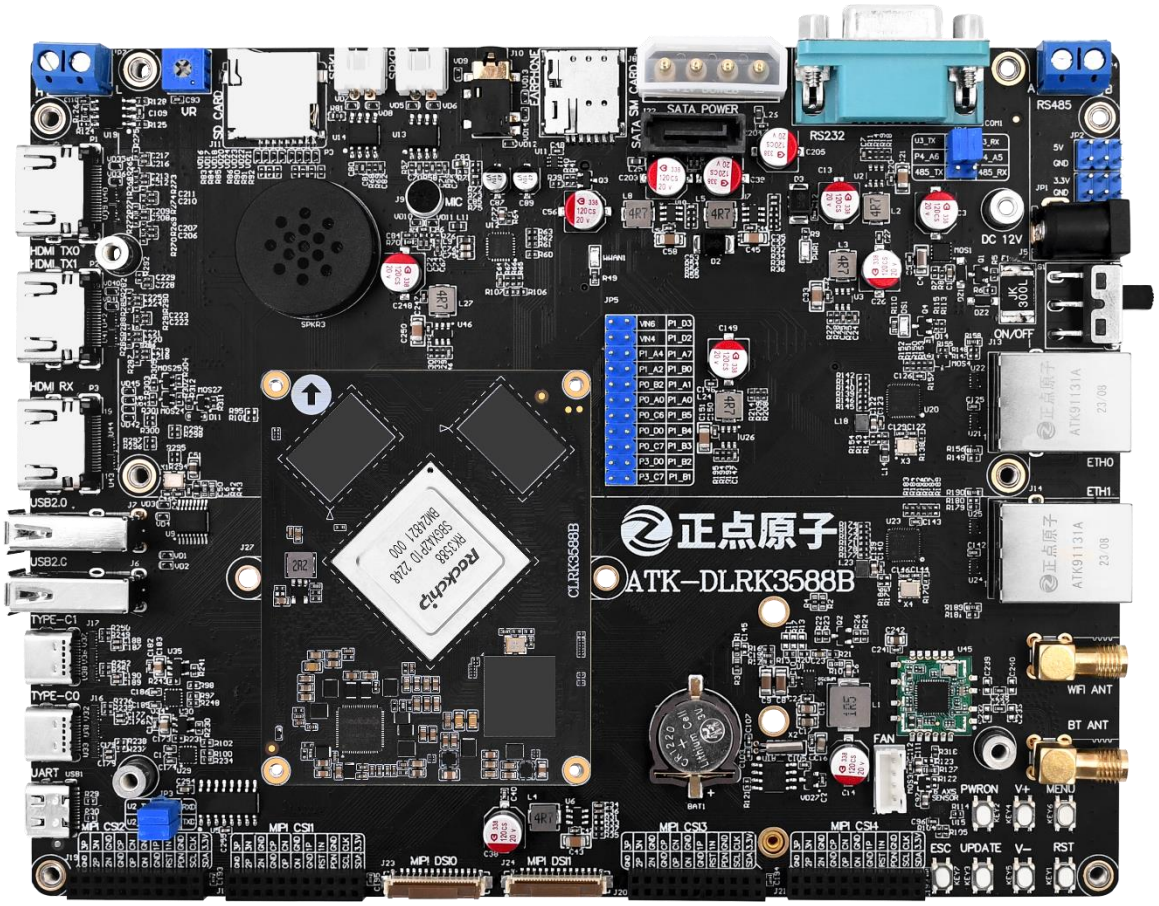
ATK-DLRK3588

Buildroot System Quick test manual

V1.1



ALIENTEK

1. Shopping:

TMALL: https://zhengdianyuanzi.tmall.com

TAOBAO: https://openedv.taobao.com

2. Download

Address: http://www.openedv.com/docs/index.html

3. FAE

Website : www.alientek.com

Forum : http://www.openedv.com/forum.php

Videos : www.yuanzige.com

Fax : +86 - 20 - 36773971

Phone : +86 - 20 - 38271790

Disclaimer

The product specifications and instructions mentioned in this document are for reference only and subject to update without prior notice; Unless otherwise agreed, this document is intended as a product guide only, and none of the representations made herein constitutes a warranty of any kind.

In order to get the latest version of product information, please regularly visit the download center or contact the customer service of Taobao ALIENTEK flagship store. Thank you for your tolerance and support.

Revision History:

Version	Version Update Notes	Responsible person	Proofreading	Date
V1.0	release officially	ALIENTEK	ALIENTEK	2024.05.08
V1.1	New additions: 1. Recording test instructions 2. Testing method for 4-channel IMX415 cameras 3. Testing method for Quectel 5G module 4. Explanation of PCIE WIFI module	ALIENTEK	ALIENTEK	2024.11.22

Brief

This document is about the user experience of the ALIENTEK DLRK3588 development board. It can be regarded as a hardware testing document. Here, we can learn about the usage methods and common testing methods of the development board, as well as the resources on the board. Therefore, this document is quite important. This document is similar to other quick experience documents of ALIENTEK. I believe everyone has some understanding of the testing methods for Linux. If there are any errors or omissions, please correct them on the ALIENTEK forum or contact the author of this document via QQ 1252699831 to point out the mistakes.

The environment used in this document:

- Windows 10 64-bit, also applicable to Windows 7-10 (Window 11 has not been tested, we all develop using Windows 10. If you are using Window 11, theoretically it is also applicable.) The documents are all introduced based on the 64-bit operating system.
- Ubuntu 20.04. It is mandatory to use 20.04 for Ubuntu. Using a new version of Ubuntu will result in a different installation environment, causing errors. Please solve it yourself!
- The document requires readers to be able to use FileZilla, WinSCP and Windows Git to transfer files between Ubuntu and Windows.
- This document is written for the ALIENTEK ATK-DLRK3588 Buildroot system. If users use other file systems, there may be no corresponding instructions or modules in the file system, causing the experiment to fail. Please pay attention!

Catalogue

Chapter 1. Preparation for using the ATK-DLRK3588 development board

Here, we will introduce the matters that need attention when using the development board. Please read carefully. Because if you do not pay attention to these matters, it may lead to the failure of your experiments or damage to the development board, etc. Therefore, please be sure to pay attention.

1.1 Precautions before powering on

- Firstly, check if the power supply is plugged in. If it is plugged in, then check whether the power switch is on and if the power supply is normal.
- Before leaving the factory, the entire board was covered with acrylic sheet and screen protector. However, in some cases, users may remove the acrylic sheet or the screen. At this time, it is necessary to pay attention to observing that the bottom board should not be placed on the clutter table. Instead, a protective film should be placed under the bottom board. Otherwise, it is easy to come into contact with metal foreign objects, causing short circuits between certain two contacts on the bottom board. When using for a long time, also pay attention to observing whether any foreign objects have fallen on the development board to prevent short circuits. Pay attention to waterproofing, moisture-proofing, dust-proofing, etc.

1.2 Serial port software installation

Here is a brief introduction to the installation of the CH340 USB serial port driver and the installation of the MobaXterm terminal. No detailed installation tutorials will be provided as it is relatively basic.

- Install the MobaXterm terminal software (or install terminal software such as Xshell, SecureCRT). This example uses MobaXterm. Network drive path: Development board network drive A - Basic materials -> 4_softwares -> MobaXterm_Installer_v12.3.zip. Double-click to open this compressed file. After waiting for the decompression, simply double-click MobaXterm_installer_12.3.msi to install. The MobaXterm installation program will guide the installation.
- Install the CH340 USB serial port driver (For the PC (computer) to communicate with the development board's serial port, we need to install this driver)

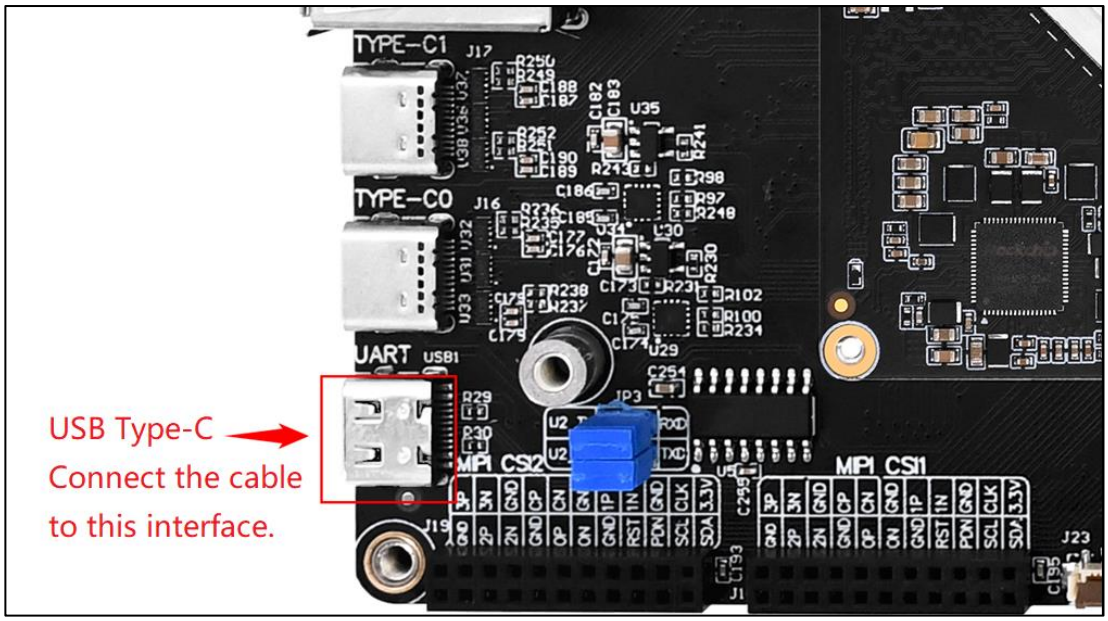
In the development board's CD-ROM A - Basic materials -> 4_softwares -> CH341SER.EXE (If you have already installed it, you don't need to install again). Find SETUP.EXE in the folder under the "4_softwares -> CH341SER.EXE" of the development board's CD-ROM. Double-click to run it. Then, in the popped-up window, directly click "Install" and wait for the pre-installation or the success window to appear. Note: The default serial port baud rate of the development board is 1500000 (1.5M), simply input 1500000 in MobaXterm.

1.3 Connecting the development board to the PC

During the development process, we exchange information with the development board through the serial port. Of course, it can also be done via the network. The network needs to know the IP address of the development board first. Therefore, the serial port is very important at this time.

The ATK-DLRK3588 development board is configured with two USB Type-C connection cables by default. On the development board, find the USB Type-C socket, find the socket with the "UART"

marking (which is the debugging serial port), and then connect the USB Type-C connection cable, and the other end connects to the USB interface of the computer host.



Turn on the power switch, power on, and the pre-installed system is the Buildroot Linux system. The serial port output information is as follows (partially).

```
U-Boot 2017.09-g6ccaae8d94-230909 #alientek (Apr 23 2024 - 10:05:10 +0800)

Model: Rockchip RK3588 Evaluation Board
MPIDR: 0x81000000
PreSerial: 2, raw, 0xfeb50000
DRAM:  16 GiB
Sysmem: init
Relocation Offset: eda1c000
Relocation fdt: eb9fa680 - eb9fecc8
CR: M/C/I
Using default environment

optee api revision: 2.0
mmc@fe2c0000: 1, mmc@fe2e0000: 0
Bootdev(atags): mmc 0
MMC0: HS400 Enhanced Strobe, 200Mhz
PartType: EFI
TEEC: Waring: Could not find security partition
DM: v2
boot mode: None
RESC: 'boot', blk@0x0001945e
resource: sha256+
FIT: no signed, no conf required
   - dev=saradc, channel=7, dtb_adc=2748, read=9, found=0
   - dev=saradc, channel=7, dtb_adc=1410, read=9, found=0
   - dev=saradc, channel=7, dtb_adc=0, read=9, found=1
DTB: rk3588-atk-mipi-5p5_720x1280#_saradc_ch7=0.dtb
HASH(c): OK
I2c0 speed: 100000Hz
vsel-gpios- not found!
en-gpios- not found!
vdd_cpu_big0_s0 800000 uV
vsel-gpios- not found!
en-gpios- not found!
vdd_cpu_big1_s0 800000 uV
I2c1 speed: 100000Hz
vsel-gpios- not found!
en-gpios- not found!
vdd_npu_s0 800000 uV
spi2: RK806: 2
ON=0x40, OFF=0x00
vdd_gpu_s0 750000 uV
vdd_cpu_lit_s0 750000 uV
vdd_log_s0 750000 uV
vdd_vdenc_s0 init 750000 uV
vdd_ddr_s0 850000 uV
Device 'gpio@fd8a0000': seq 0 is in use by 'gpio@fd8a0000'
get vp0 plane mask:0x5, primary id:2, cursor_plane:-1, from dts
get vp1 plane mask:0xa, primary id:3, cursor_plane:-1, from dts
get vp2 plane mask:0x140, primary id:8, cursor_plane:-1, from dts
get vp3 plane mask:0x280, primary id:9, cursor_plane:-1, from dts
Could not find baseparameter partition
Model: ATK-DLRK3588 Board
MPIDR: 0x81000000
Rockchip UBOOT DRM driver version: v1.0.1
vp0 have layer nr:2[0 2 ], primary plane: 2
```

1.4 SSH login password

The SSH login password for the root user of the ALIENTEK ATK-DLRK3588 is "root". The default user for booting up is also root, and the password is also "root". The default boot process does not require entering a password. The root user automatically logs in. Alright, now you can proceed to the third chapter for hardware testing. The second chapter is about burning the Buildroot Linux system. If you have modified the Buildroot Linux system or damaged the system and want to restore it to the factory state, then you can follow the operations in the second chapter.

Chapter 2. Burn the Buildroot Linux system

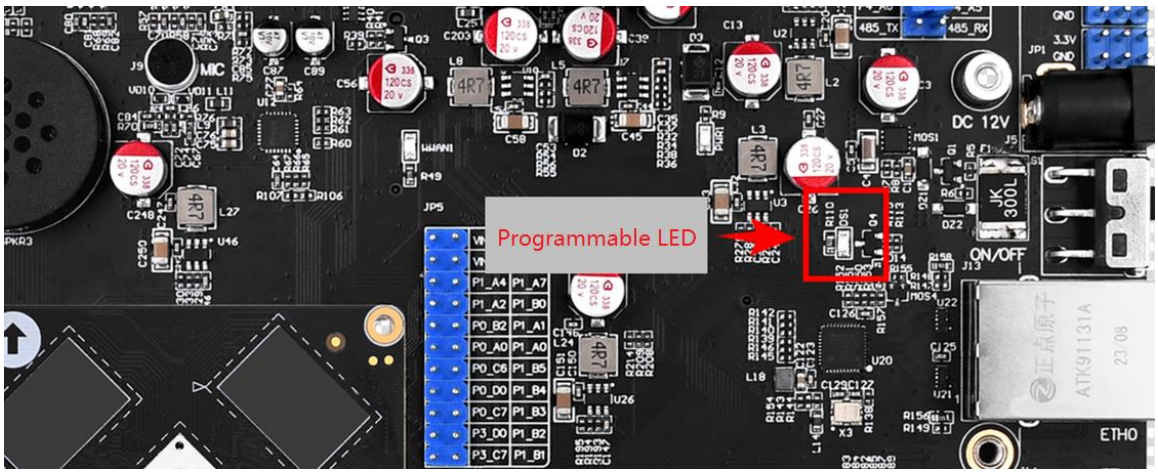
The default system pre-installed on the ATK-DLRK3588 development board is the Buildroot Linux system. You don't need to re-flash it. Here is an example of how to flash the system onto the development board. Or if you have already experienced flashing an Android system and now want to revert to the Linux system, or if you have damaged the Linux system and want to restore it to the factory state, then you can refer to this chapter. We have prepared a detailed document for flashing the system. Please proceed to view Chapter 10_user_manual -> 03_Supplementary_Documents -> XX "[ALIENTEK] ATK-DLRK3588 Factory Image Flashing Guide" (XX represents an undetermined number). From Chapter 1 to Chapter 3, you can flash the Buildroot Linux system.

Chapter 3. Function testing of the ATK-DLRK3588 development board

This chapter will introduce how to test the ATK-DLRK3588 development board. It explains the usage and testing methods of the development board. It is very helpful for the development process. Note that many tests require self-provided testing tools. Please complete the testing steps according to your personal situation. The hardware has already been tested when it was manufactured, so users do not need to test again.

3.1 LED Testing

The ALIENTEK ATK-DLRK3588 development board is equipped with a user LED. The relationship between them and the corresponding hardware pins on the baseboard is as shown in the following table. In the factory system, we use the LED as a heartbeat indicator (can be programmed), and it is generally used to indicate whether the system is operating normally.



Pin name	WORKING_LEDEN_H
ATK-DLRK3588	GPIO1_A3

In the factory system, we can control the LED status by using the following instructions. After inputting them, check the status of the LED on the development board.

cat /sys/class/leds/work/trigger # View the current trigger mode and the supported trigger modes of LED0. The default mode is [heartbeat].

echo none > /sys/class/leds/work/trigger # Change the trigger mode of LED0

echo 1 > /sys/class/leds/work/brightness # Turn on LED0. At this time, it is in a constantly illuminated state. The value is a positive integer greater than 0, usually written as 1.

echo 0 > /sys/class/leds/work/brightness # Turn off LED0. At this moment, it is in the off state, and the value is 0.

3.2 Key Test

The ATK-DLRK3588 development board of the ALIENTEK is equipped with 7 keys. Among them, 4 keys are connected to the first channel of the ADC. When these keys are pressed, the ADC can capture different voltage changes on the first channel.

	KEY4	KEY5	KEY6	KEY7
Name of the button	V+	V-	MENU	ESC
Value of ADC0	40	945	2027	2807

Run the following command in the serial terminal of the development board for testing, then input the number "9", as the key event is event9. To stop the test, press Ctrl + c. Please press the above four keys on the development board. Do not press any other keys, such as the reset key REST, etc.

```
evtest
```

The running result is as follows. As you can see, by pressing the corresponding key, the key type will be printed and the pressing time value will be 1, while the releasing time value will be 0.

```
root@ATK-DLRK3588:/# evtest
No device specified, trying to scan all of /dev/input/event*
Available devices:
/dev/input/event0:      rk805 pwrkey
/dev/input/event1:      gsensor
/dev/input/event2:      dsi0_ts_gt9xx
/dev/input/event3:      rockchip-dp0 rockchip-dp0
/dev/input/event4:      rockchip-dp1 rockchip-dp1
/dev/input/event5:      rockchip,hdmiin rockchip,hdmiin
/dev/input/event6:      headset-keys
/dev/input/event7:      rockchip-es8388 Headset
/dev/input/event8:      gyro
/dev/input/event9:      adc-keys
/dev/input/event10:     rockchip-hdmi0 rockchip-hdmi0
/dev/input/event11:     rockchip-hdmi1 rockchip-hdmi1
Select the device event number [0-11]: 9
Input driver version is 1.0.1
Input device ID: bus 0x19 vendor 0x1 product 0x1 version 0x100
Input device name: "adc-keys"
Supported events:
  Event type 0 (EV_SYN)
  Event type 1 (EV_KEY)
    Event code 114 (KEY_VOLUMEDOWN)
    Event code 115 (KEY_VOLUMEUP)
    Event code 139 (KEY_MENU)
    Event code 158 (KEY_BACK)
Properties:
Testing ... (interrupt to exit)
Event: time 1714029593.079111, type 1 (EV_KEY), code 114 (KEY_VOLUMEDOWN), value 1
Event: time 1714029593.079111, -------------- SYN_REPORT ------------
Event: time 1714029593.388858, type 1 (EV_KEY), code 114 (KEY_VOLUMEDOWN), value 0
Event: time 1714029593.388858, -------------- SYN_REPORT ------------
Event: time 1714029596.798994, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), value 1
Event: time 1714029596.798994, -------------- SYN_REPORT ------------
Event: time 1714029597.005655, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), value 0
Event: time 1714029597.005655, -------------- SYN_REPORT ------------
```

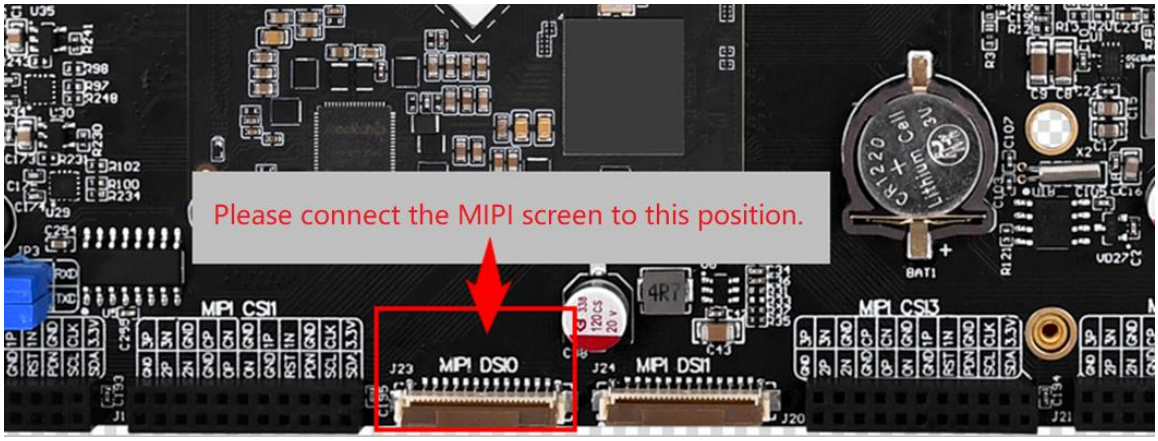
3.3 MIPI screen test

In the factory system, since the screen wiring of ALIENTEK 's screen has an "ID resistor" connected to the 7th channel of the ADC of the main chip, the factory system can recognize the

resolution of the loaded different screens. ALIENTEK uses three types of MIPI screens, as shown in the table below:

Screen size	Touch chip	ADC value
5.5-inch screen (720x1280)	gt911	0
5.5-inch screen (1080x1920)	gt911	1410
10.1-inch screen (800x1280)	gt928	2748

The default Linux images all only enable the DSI0 interface. If you have purchased a MIPI screen, please connect it to the DSI0 interface. DSI1 is default to be off. Because Weston cannot use two DSI as the main screens.



5.5-inch 1080P MIPI screen offers a clearer display effect in reality.

3.3.1 Touch Test

After the development board is powered on, we use the "evtest" command to check the touch events corresponding to the touch screen. The same method as in Section 3.2 can also be used to test whether the screen is functioning properly for touch input. Inputting the number 2, the corresponding driver for the screen is "goodix-ts". Of course, clicking the Qt interface preset at the factory and being able to touch normally is also acceptable.

3.3.2 Multi-touch testing

The MIPI screen supports up to 5 points of touch at its maximum. Although Qt supports multi-touch, the controls we use are all single-point. Multi-touch requires a multi-touch program. Therefore, at the factory interface, you cannot experience multi-touch. So we still use the evtest command for testing. At that time, you could use the tslib test (the default file system is not configured). Similarly, the same method as in Section 3.2 can also be used to test if the screen is functioning properly. Enter the number 2, and the corresponding driver of the screen is goodix-ts.

evtest

```
root@ATK-DLRK3588:/# evtest
No device specified, trying to scan all of /dev/input/event*
Available devices:
/dev/input/event0:      rk805 pwrkey
/dev/input/event1:      qsensor
/dev/input/event2:      dsi0_ts_gt9xx
/dev/input/event3:      rockchip-dp0 rockchip-dp0
/dev/input/event4:      rockchip-dp1 rockchip-dp1
/dev/input/event5:      rockchip,hdmiin rockchip,hdmiin
/dev/input/event6:      headset-keys
/dev/input/event7:      rockchip-es8388 Headset
/dev/input/event8:      gyro
/dev/input/event9:      adc-keys
/dev/input/event10:     rockchip-hdmi0 rockchip-hdmi0
/dev/input/event11:     rockchip-hdmi1 rockchip-hdmi1
Select the device event number [0-11]: 2
input driver version is 1.0.1
Input device ID: bus 0x18 vendor 0xdead product 0xbeef version 0x28bb
Input device name: "dsi0_ts_gt9xx"
Supported events:
  Event type 0 (EV_SYN)
  Event type 1 (EV_KEY)
    Event code 330 (BTN_TOUCH)
  Event type 3 (EV_ABS)
    Event code 48 (ABS_MT_TOUCH_MAJOR)
      Value      0
      Min        0
      Max      255
    Event code 50 (ABS_MT_WIDTH_MAJOR)
      Value      0
      Min        0
      Max      255
    Event code 53 (ABS_MT_POSITION_X)
      Value      0
      Min        0
      Max     1080
    Event code 54 (ABS_MT_POSITION_Y)
      Value      0
      Min        0
      Max     1920
    Event code 57 (ABS_MT_TRACKING_ID)
      Value      0
      Min        0
      Max      255
Properties:
  Property type 1 (INPUT_PROP_DIRECT)
Testing ... (interrupt to exit)
Event: time 1714030943.376169, type 1 (EV_KEY), code 330 (BTN_TOUCH), value 1
Event: time 1714030943.376169, type 3 (EV_ABS), code 53 (ABS_MT_POSITION_X), value 653
Event: time 1714030943.376169, type 3 (EV_ABS), code 54 (ABS_MT_POSITION_Y), value 1851
Event: time 1714030943.376169, type 3 (EV_ABS), code 48 (ABS_MT_TOUCH_MAJOR), value 49
```

As can be seen below, when all 5 fingers are pressed simultaneously, a positive value indicates the number of points, while a negative value indicates that the corresponding point has been released. There are a total of 5 points, and the 5-point touch test is normal.

```
Testing ... (interrupt to exit)
Event: time 1680837017.859986, type 3 (EV_ABS), code 47 (ABS_MT_SLOT), value 0
Event: time 1680837017.859986, type 3 (EV_ABS), code 57 (ABS_MT_TRACKING_ID), value 0
Event: time 1680837017.859986, type 3 (EV_ABS), code 53 (ABS_MT_POSITION_X), value 340
Event: time 1680837017.859986, type 3 (EV_ABS), code 54 (ABS_MT_POSITION_Y), value 1093
Event: time 1680837017.859986, type 3 (EV_ABS), code 48 (ABS_MT_TOUCH_MAJOR), value 49
Event: time 1680837017.859986, type 3 (EV_ABS), code 50 (ABS_MT_WIDTH_MAJOR), value 49
Event: time 1680837017.859986, type 3 (EV_ABS), code 47 (ABS_MT_SLOT), value 1
Event: time 1680837017.859986, type 3 (EV_ABS), code 57 (ABS_MT_TRACKING_ID), value 1
Event: time 1680837017.859986, type 3 (EV_ABS), code 53 (ABS_MT_POSITION_X), value 564
Event: time 1680837017.859986, type 3 (EV_ABS), code 54 (ABS_MT_POSITION_Y), value 1034
Event: time 1680837017.859986, type 3 (EV_ABS), code 48 (ABS_MT_TOUCH_MAJOR), value 66
Event: time 1680837017.859986, type 3 (EV_ABS), code 50 (ABS_MT_WIDTH_MAJOR), value 66
Event: time 1680837017.859986, type 3 (EV_ABS), code 47 (ABS_MT_SLOT), value 2
Event: time 1680837017.859986, type 3 (EV_ABS), code 57 (ABS_MT_TRACKING_ID), value 2
Event: time 1680837017.859986, type 3 (EV_ABS), code 53 (ABS_MT_POSITION_X), value 169
Event: time 1680837017.859986, type 3 (EV_ABS), code 54 (ABS_MT_POSITION_Y), value 1013
Event: time 1680837017.859986, type 3 (EV_ABS), code 48 (ABS_MT_TOUCH_MAJOR), value 67
Event: time 1680837017.859986, type 3 (EV_ABS), code 50 (ABS_MT_WIDTH_MAJOR), value 67
Event: time 1680837017.859986, type 3 (EV_ABS), code 47 (ABS_MT_SLOT), value 3
Event: time 1680837017.859986, type 3 (EV_ABS), code 57 (ABS_MT_TRACKING_ID), value 3
Event: time 1680837017.859986, type 3 (EV_ABS), code 53 (ABS_MT_POSITION_X), value 360
Event: time 1680837017.859986, type 3 (EV_ABS), code 54 (ABS_MT_POSITION_Y), value 560
Event: time 1680837017.859986, type 3 (EV_ABS), code 48 (ABS_MT_TOUCH_MAJOR), value 61
Event: time 1680837017.859986, type 3 (EV_ABS), code 50 (ABS_MT_WIDTH_MAJOR), value 61
Event: time 1680837017.859986, type 3 (EV_ABS), code 47 (ABS_MT_SLOT), value 4
Event: time 1680837017.859986, type 3 (EV_ABS), code 57 (ABS_MT_TRACKING_ID), value 4
Event: time 1680837017.859986, type 3 (EV_ABS), code 53 (ABS_MT_POSITION_X), value 254
Event: time 1680837017.859986, type 3 (EV_ABS), code 54 (ABS_MT_POSITION_Y), value 232
Event: time 1680837017.859986, type 3 (EV_ABS), code 48 (ABS_MT_TOUCH_MAJOR), value 74
Event: time 1680837017.859986, type 3 (EV_ABS), code 50 (ABS_MT_WIDTH_MAJOR), value 74
Event: time 1680837017.859986, -------------- SYN_REPORT ------------
Event: time 1680837018.021889, type 3 (EV_ABS), code 47 (ABS_MT_SLOT), value 1
Event: time 1680837018.021889, type 3 (EV_ABS), code 57 (ABS_MT_TRACKING_ID), value -1
Event: time 1680837018.021889, -------------- SYN_REPORT ------------
Event: time 1680837018.028818, type 3 (EV_ABS), code 47 (ABS_MT_SLOT), value 0
Event: time 1680837018.028818, type 3 (EV_ABS), code 57 (ABS_MT_TRACKING_ID), value -1
Event: time 1680837018.028818, type 3 (EV_ABS), code 47 (ABS_MT_SLOT), value 3
Event: time 1680837018.028818, type 3 (EV_ABS), code 57 (ABS_MT_TRACKING_ID), value -1
Event: time 1680837018.028818, type 3 (EV_ABS), code 47 (ABS_MT_SLOT), value 4
Event: time 1680837018.028818, type 3 (EV_ABS), code 57 (ABS_MT_TRACKING_ID), value -1
Event: time 1680837018.028818, -------------- SYN_REPORT ------------
Event: time 1680837018.048725, type 3 (EV_ABS), code 47 (ABS_MT_SLOT), value 2
Event: time 1680837018.048725, type 3 (EV_ABS), code 57 (ABS_MT_TRACKING_ID), value -1
Event: time 1680837018.048725, -------------- SYN_REPORT ------------
```

3.3.3 MIPI backlight test

The backlight of the MIPI screen supports 255 levels of PWM adjustment. The brightness range is from 0 to 255, with the default value being 255. The larger the value, the brighter the screen.

cat /sys/class/backlight/backlight/max_brightness # Check the maximum brightness level of the LCD.

cat /sys/class/backlight/backlight/brightness # Check the current brightness level

echo 200 > /sys/class/backlight/backlight/brightness # Modify the brightness level of the current observation screen and adjust its brightness.

cat /sys/class/backlight/backlight/brightness # Check the current brightness level again

3.4 HDMI Test

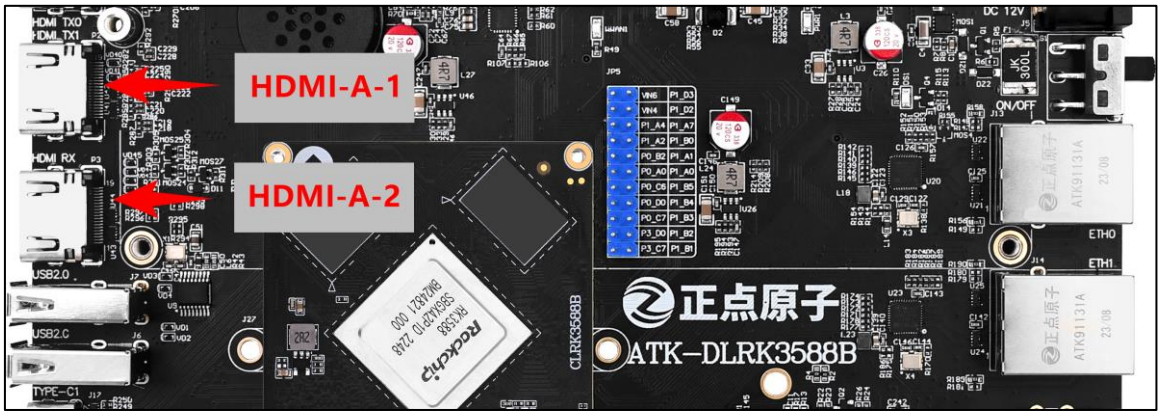
Note: Due to the wide variety of HDMI monitors available on the market, some may display slowly. This is mainly because the monitor needs initialization. However, Weston starts up faster, causing Weston to be unable to find the output display, resulting in a black screen. The following steps need to be followed to solve the problem. If some special monitors cannot recognize the corresponding resolution with the development board, then you can force set the corresponding resolution according to the Qt display settings of the Weston desktop based on the Buildroot system of ALIENTEK. For more information, please refer to the official RK documentation for HDMI-related settings.

```
/etc/init.d/S49weston restart          # Restart Weston
/etc/init.d/S50systemui restart        # Restart ui
```

The default factory-installed Linux system supports HDMI. Please connect the display using an HDMI cable for testing the display. Note: The Qt interface in the factory is adapted to a resolution screen of 9:16, which means resolutions such as 720x1280, 1080x1920, and 2160x3840 are all supported. However, most displays have a 16:9 resolution, which is 1280x720, 1920x1080, and 3840x2160 (with a maximum resolution not exceeding 8K at 30fps). The default MIPI screen is the main screen. When you connect an MIPI, HDMI, or other screens, they are secondary screens. The secondary screens will map the content of the main screen, so you will see a centered UI interface. This is because the MIPI is a vertical screen and the HDMI is a horizontal screen, and their proportions do not match, so the mapped content will be centered.

For Weston-related configurations, please refer to Rockchip_Developer_Guide_Buildroot_Weston_CN.pdf and [ALIENTEK] Weston Desktop Qt Display Settings based on the Buildroot system.

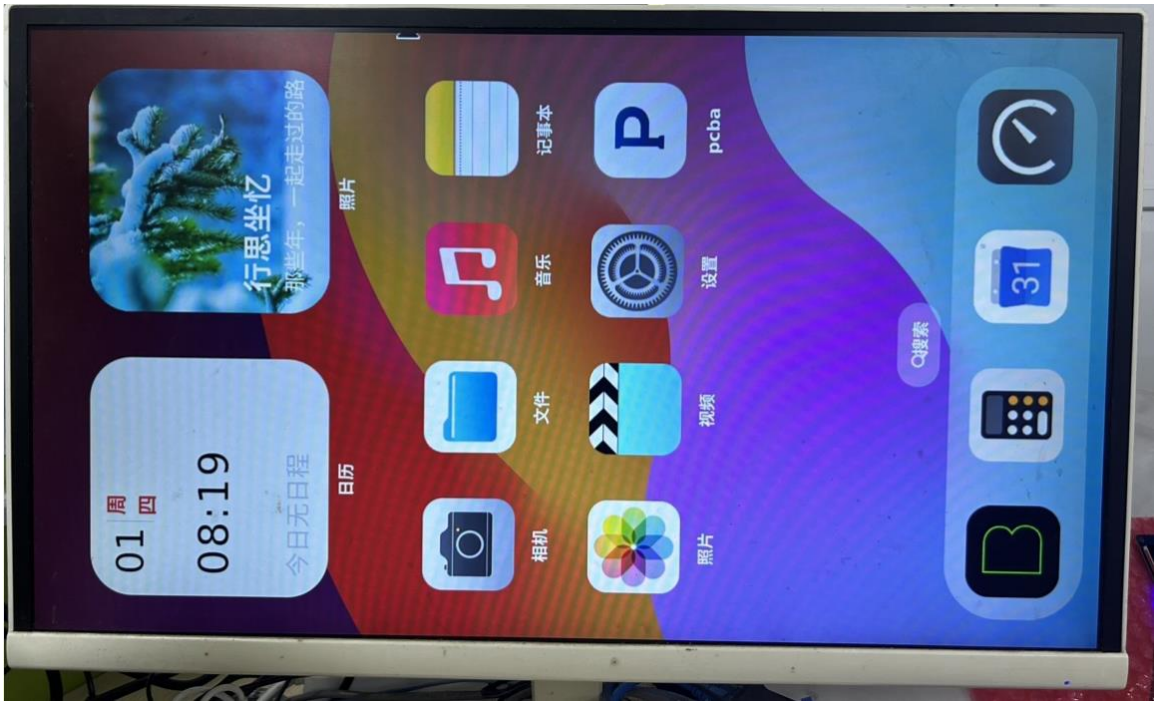
Plug the HDMI cable into the HDMI-A-2 or HDMI-A-1 interface in the following diagram.



The display result of the HDMI screen is as follows.

Note: When you only have HMDI and do not connect the MIPI screen, then HDMI will be the main screen. Thus, the UI on HDMI will cover the entire HDMI screen, causing display distortion. If you want to experience the UI functions we developed, then you need to rotate the HMDI screen.

```
echo "output:HDMI-A-2:rotate90" > /tmp/.weston_drm.conf    # It might be HDMI-A-1, depending on the position of the HDMI interface you plug in!
/etc/init.d/S50systemui restart      # The UI will only adapt to the portrait mode after a restart. It is recommended that you physically rotate your screen by 90 degrees to fully experience the UI.
```

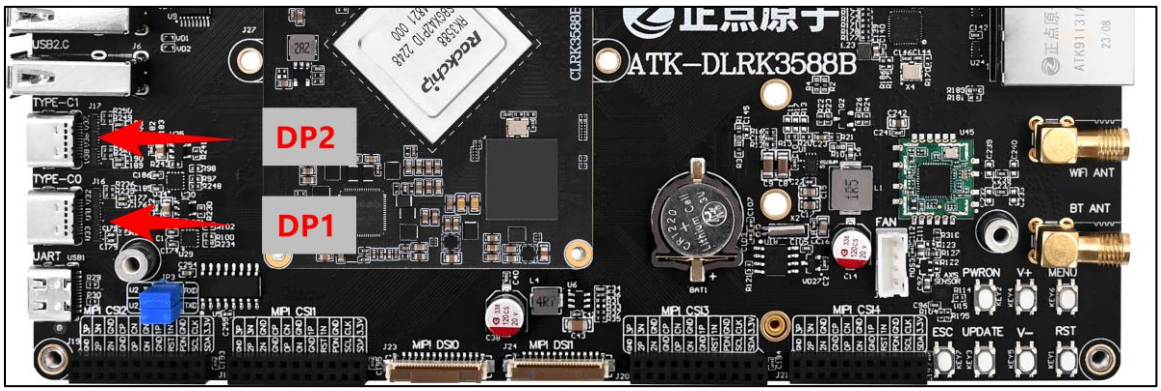


Please note: When the hot-pluggable device is connected to the main screen, it cannot be unplugged. Otherwise, Weston will fail to locate the screen output, and Weston will crash. You need to restart Weston and then restart the UI. It is also recommended that you turn on the device after connecting the HDMI cable, otherwise Weston will also fail to find a place to display.

```
/etc/init.d/S49weston restart          # Reset weston
/etc/init.d/S50systemui restart        # Reset ui
```

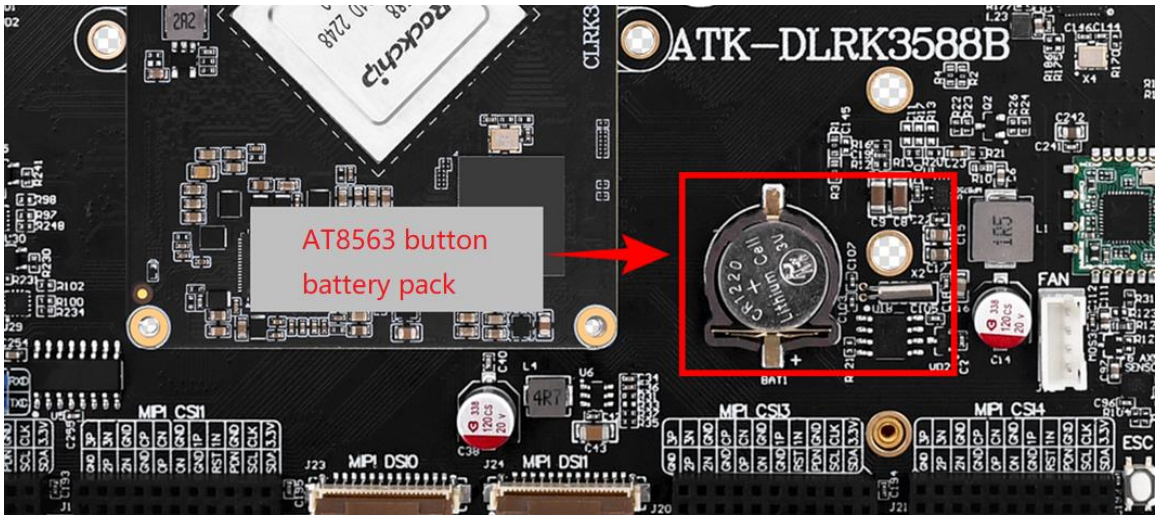
3.5 DP interface test

Please prepare a Type-C to DP cable. Please refer to the description of the HDMI interface in the previous section. Note that the TYPE-C0 interface is also the interface used for ADB and also the interface for programming.



3.6 RTC Clock Test

The ATK-DLRK3588 development board has a hardware clock. There is an RTC clock chip AT8563T on the baseboard, both of which are external RTC clocks of the chip. If the clock accuracy needs to be improved, a high-precision crystal oscillator should be used.



Please check if the RTC button battery is installed on the bottom plate of the development board. You can also use a multimeter to check if the RTC battery is charged. If the reading is around 3.3v, it is normal. To prevent the time from not being saved due to the battery running out of power (note: the battery is a consumable item. If it is dead, please replace the button battery).

The Linux system has two clocks: one is the system time (software clock), and the other is the hardware clock (hardware clock). The date and hwclock commands can be used to view and set the system time and hardware time respectively. The system clock disappears when power is off, while the RTC clock will operate for a long time with a battery. The system clock will synchronize with the RTC clock during system restart.

To view the system clock, use the command date. Note that the factory system uses the ntpd clock synchronization service, so if you have an internet connection when you boot up and plug in an internet cable, your system clock will be synchronized with the time on the network. However, this synchronized clock will not be written into the hardware clock. Therefore, when you use date and hwclock, you may see two different times.

date

Set the current system clock

date -s "2024-04-26 10:00:00"

Check the system clock

date

Write to the hardware clock using hwclock.

hwclock -w

Check the hardware clock to see if it is the one set by the above "hwclock -w".

hwclock

```
root@ATK-DLRK3588:/# date
Fri Apr 26 11:06:56 AM CST 2024
root@ATK-DLRK3588:/# date -s "2024-04-26 10:00:00"
Fri Apr 26 10:00:00 AM CST 2024
root@ATK-DLRK3588:/# date
Fri Apr 26 10:00:06 AM CST 2024
root@ATK-DLRK3588:/# hwclock -w
root@ATK-DLRK3588:/# hwclock
Fri 26 Apr 2024 10:00:17 AM CST  0.000000 seconds
root@ATK-DLRK3588:/#
```

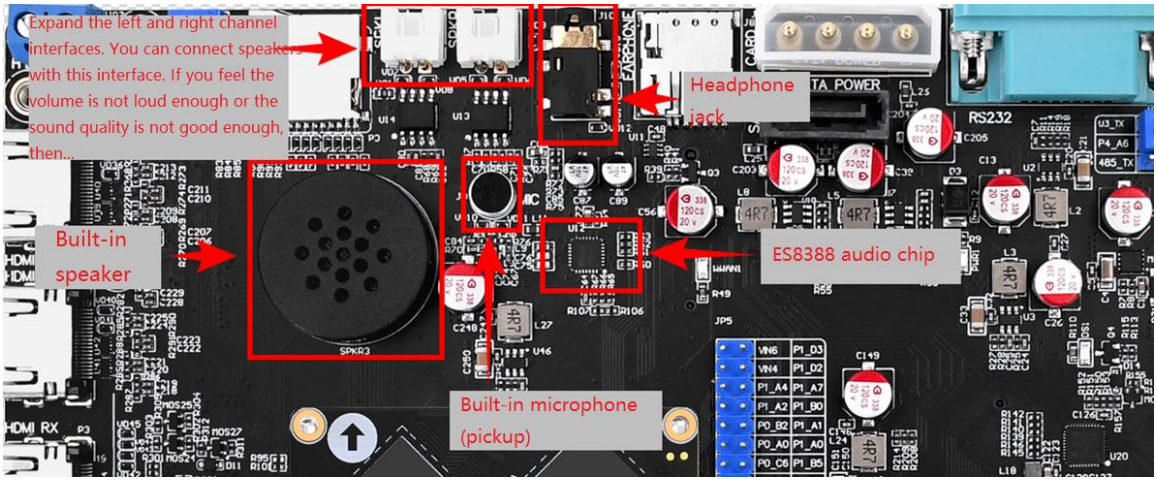
When I started up, I plugged in the network cable, so I saw the time as 11:06. Then I set it to 10 o'clock and restarted the system without plugging in the network cable to prevent ntpd from synchronizing the time from the network. The result was as follows after three minutes.

date

```
root@ATK-DLRK3588:/# date
Fri Apr 26 10:03:10 AM CST 2024
root@ATK-DLRK3588:/#
```

3.7 Audio Test

The ATK-DLRK3568 baseboard contains a CODEC ES8388 audio chip. The ES8388 is a high-performance, low-power, and low-cost stereo audio codec. It consists of 2-ch ADC, 2-ch DAC, microphone amplifier, headphone amplifier, digital audio effects, and analog mixing and gain functions. This chip uses high-order multi-bit delta-sigma modulation technology to convert data between digital and analog, making the device less sensitive to clock jitter and with lower out-of-band noise.



3.7.1 Play audio test

You can play audio files using system commands aplay or gst-play-1.0. Execute the following command to play the audio file. Users can also copy the audio file to the file system and use the command aplay/gst-play-1.0 + audio file to play it.

```
aplay /usr/share/sounds/test.wav
# Use aplay to play audio files. It supports wav format but does not support mp3.
gst-play-1.0 /opt/ui/src/apps/resource/audio/*mp3
# Use gst-play-1.0 to play. It supports wav and other formats such as mp3.
```

3.7.2 Audio test

```
aplay -l    # Here is -L, in lowercase!
```

Check the sound card number, and you will find that es8388 is the 3rd sound card. All the places that use the number 3 are related to this.

```
root@ATK-DLRK3588:/# aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: rockchipdp0 [rockchip-dp0], device 0: rockchip-dp0 spdif-hifi-0 [rockchip-dp0 spdif-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 1: rockchipdp1 [rockchip-dp1], device 0: rockchip-dp1 spdif-hifi-0 [rockchip-dp1 spdif-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 3: rockchipes8388 [rockchip-es8388], device 0: dailink-multicodecs ES8323.7-0011-0 [dailink-multicodecs ES8323.7-0011-0]
  Subdevices: 0/1
  Subdevice #0: subdevice #0
card 4: rockchiphdmi0 [rockchip-hdmi0], device 0: rockchip-hdmi0 i2s-hifi-0 [rockchip-hdmi0 i2s-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 5: rockchiphdmi1 [rockchip-hdmi1], device 0: rockchip-hdmi1 i2s-hifi-0 [rockchip-hdmi1 i2s-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
root@ATK-DLRK3588:/#
```

```
amixer -c 3 cset name='PCM Volume' '192' '192'
# Set the volume of es8388 to 0 - 192. The maximum is 192. The default setting is already at the maximum, so this instruction is not necessary to execute.
arecord -D hw:3,0 -r 44100 -f S16_LE -d 10 -c 2 record.wav          # Start recording
```

Other settings do not need to be made. The default is the maximum. If you need to make settings, please use alsamixer and then press F6 to select the corresponding sound card, and then set the corresponding item.

Explanation of arcanord command parameters:

- -c 0: Use sound card 0.
- -f S16_LE: Sample in S16_LE format.
- -r 44100: Sampling rate 44.1K
- -d 10: Recording duration 10s
- -c 2 specifies dual-channel
- record.wav: Audio file generated by the recording

Play the audio file recorded above. Note: The size of the generated file is related to the format set by the user and the length of the recording. Also, since the volume was set too high during the recording and the microphone did not have noise reduction, there might be some background noise. Please lower the capture volume (sensitivity) and test again.

Explanation:

If you are using the development board CD, the B drive - Development Environment and SDK -> 02, ATK-DLRK3588 development board SDK -> 06, linux_r8_sdk -> atk-rk3588_linux_release_R8_v1.0_20250104.tgz - the SDK source code compilation of the Buildroot image, before playing the audio recording, execute the following command to convert the audio file:

```
ffmpeg -i record.wav -af "pan=stereo|c0=c0|c1=c0" output.wav
```

Execute the command to play the recording again

```
aplay record.wav    # play recording
```

```
root@ATK-DLRK3588:/# amixer -c 3 cset name='PCM Volume' '192' '192'
numid=31,iface=MIXER,name='PCM Volume'
  ; type=INTEGER,access=rw---R--,values=2,min=0,max=192,step=0
  : values=192,192
  | dBscale-min=-96.00dB,step=0.50dB,mute=1
root@ATK-DLRK3588:/# arecord -D hw:3,0 -r 44100 -f S16_LE -d 10 -c 2 record.wav
Recording WAVE 'record.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
root@ATK-DLRK3588:/# aplay record.wav
Playing WAVE 'record.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
root@ATK-DLRK3588:/#
```

3.7.3 Volume control

Input instructions can regulate the volume.

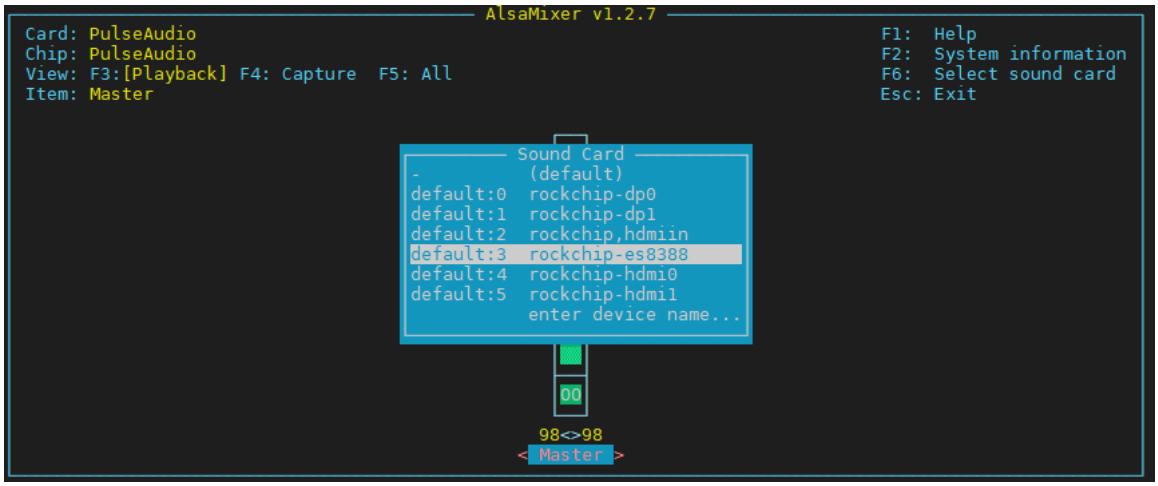
The volume value range is from 0 to 192. The factory system has already set the maximum volume by default. Please note that our Qt UI App all set the volume of the software within the application.

```
amixer -c 3 cset name='PCM Volume' '192' '192'
```

```
root@ATK-DLRK356X:/# amixer cset name='Master Playback Volume' '63','63'
numid=4,iface=MIXER,name='Master Playback Volume'
  ; type=INTEGER,access=rw---RW-,values=2,min=0,max=100,step=0
  : values=63,63
  | dBscale-min=-30.00dB,step=0.30dB,mute=0
root@ATK-DLRK356X:/#
```

Or you can directly use alsamixer to adjust the volume through the graphical interface. The left and right arrow keys can be used to select the item to be adjusted, and the up and down arrow keys can be used to adjust the volume. If you need to control the es83388 sound card, please press F6, select es83388, and then set the relevant items.

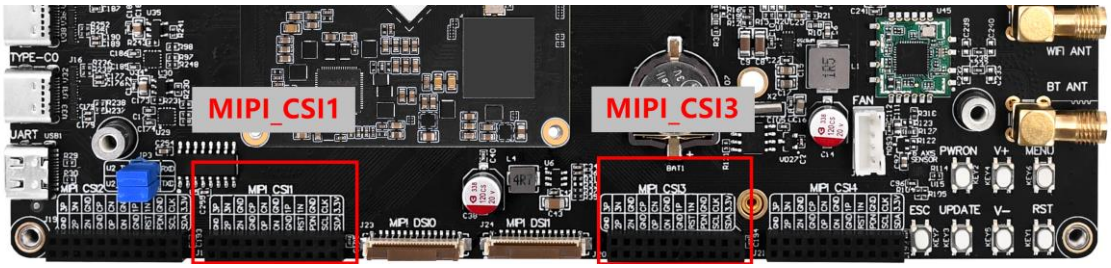
```
alsamixer
```



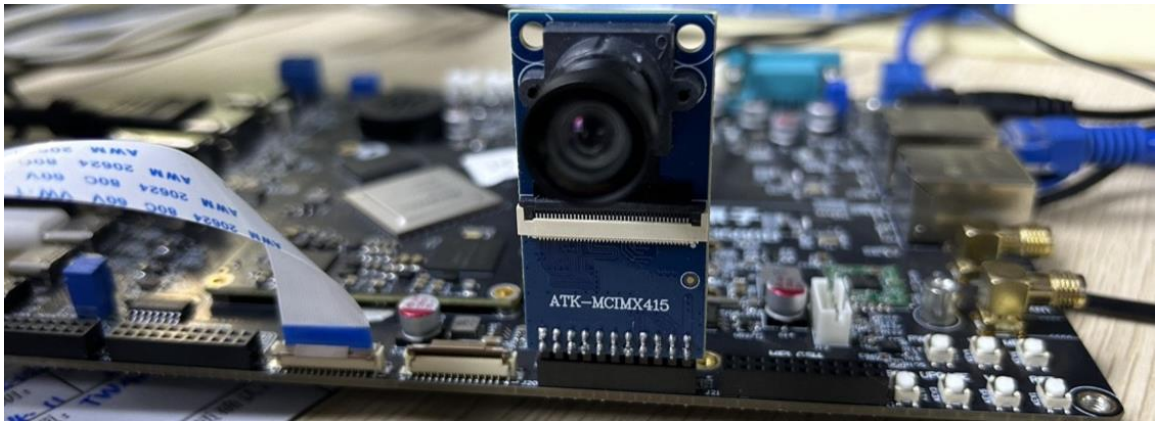
3.8 Camera Test

ATK-DLRK3588 supports MIPI CSI cameras. ALIENTEK has developed a camera module for this board. Currently, this module only supports IMX415 (800W pixels). If there are future updates supporting other cameras, please consult the Taobao technical support.

Please note that the default configuration of the factory-installed Linux system includes MIPI CSI1 and MIPI CSI3 interfaces. Please insert the camera module into any one of the two middle CSI interfaces. Currently, four MIPI cameras are not supported on Linux. This is because RK has some bugs when using four cameras, so we only configured two! However, the hardware supports it. We have tested it on the Android system! Please insert one of these two positions. Of course, if you have two cameras, you can insert both at the same time!



The installation method is as follows: The camera lens should not face the main board. Do not insert it in the reverse direction or off-center. Inserting it incorrectly may cause the module to be damaged. As shown in the picture below.



If you have a MIPI screen or are connected to a monitor, then you can click on the Qt camera app developed by ALIENTEK. As shown in the picture below, the author used a MIPI screen to capture an image from the camera. The result is as follows.

By default in the factory kernel, the camera may be the video22 node or the video31 node, depending on whether you have it connected to CSI3 or CSI1. Note that the following instructions use the maximum resolution for display, which may extend beyond the screen. Please modify it according to your personal screen resolution, such as 1080x1920 resolution. Correspondingly, modify the width and height below. MIPI cameras can take any resolution within the available range. Of course, there is a minimum range. Please refer to the RK source code for details. For example, you can choose 1000x1000 as well.

```
gst-launch-1.0 v4l2src device=/dev/video22 ! video/x-raw,format=NV12,width=3840,height=2160,framerate=30/1 ! autovideosink
```

New addition:

If using the development board CD, B drive - Development environment and SDK -> 02, ATK-DLRK3588 development board SDK -> 06, linux_r8_sdk -> atk-rk3588_linux_release_R8_v1.0_20250104.tgz, by default, it supports simultaneously calling 4 IMX415 cameras. The correspondence between the silk screen on the development board and the /dev/videoX node is shown in the following table:

Development board silk-screening	/dev/videoX
MIPI CSI1	/dev/video62
MIPI CSI2	/dev/video71
MIPI CSI3	/dev/video44
MIPI CSI4	/dev/video53

If all four IMX415 cameras have been connected simultaneously, you can execute the following command for testing (do not simply copy):

```
gst-launch-1.0 \
v4l2src device=/dev/video44 ! video/x-raw,format=NV12,width=640,height=480,framerate=30/1 ! videoconvert ! autovideosink sync=false \
```

v4l2src device=/dev/video62 ! video/x-raw,format=NV12,width=640,height=480,framerate=30/1 ! videoconvert ! autovideosink sync=false \

v4l2src device=/dev/video53 ! video/x-raw,format=NV12,width=640,height=480,framerate=30/1 ! videoconvert ! autovideosink sync=false \

v4l2src device=/dev/video71 ! video/x-raw,format=NV12,width=640,height=480,framerate=30/1 ! videoconvert ! autovideosink sync=false

The correct format of the command is as shown in the following figure.

```
gst-launch-1.0 \
v4l2src device=/dev/video44 ! video/x-raw,format=NV12,width=640,height=480,framerate=30/1 ! videoconvert ! autovideosink sync=false \
v4l2src device=/dev/video62 ! video/x-raw,format=NV12,width=640,height=480,framerate=30/1 ! videoconvert ! autovideosink sync=false \
v4l2src device=/dev/video53 ! video/x-raw,format=NV12,width=640,height=480,framerate=30/1 ! videoconvert ! autovideosink sync=false \
v4l2src device=/dev/video71 ! video/x-raw,format=NV12,width=640,height=480,framerate=30/1 ! videoconvert ! autovideosink sync=false
```

The test results are shown in the following figure.

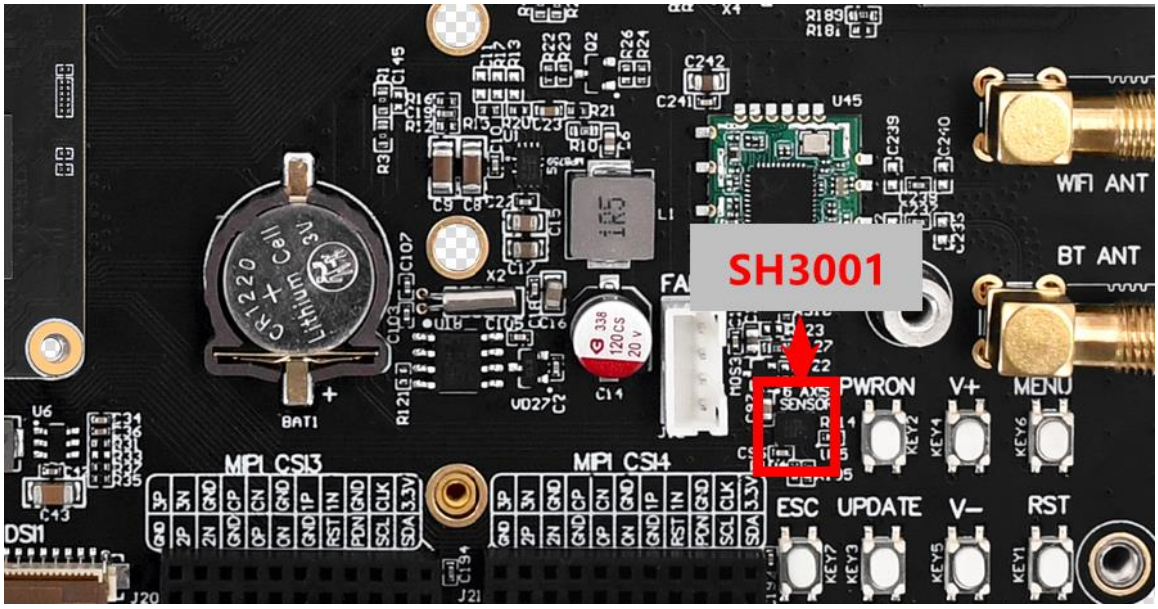
3.9 SH3001 Six-Axis Sensor Test

SH3001 Introduction:

SH3001 is a six-axis IMU (Inertial Measurement Unit) inertial measurement unit. SH3001 integrates a three-axis gyroscope and a three-axis accelerometer internally. It is small in size, has low power consumption, and is suitable for applications in the consumer electronics market. It can provide high-precision real-time angular velocity and linear acceleration data. SH3001 has excellent temperature stability and maintains high resolution within the operating range of -40°C to 85°C. Application scenarios: smartphone games, optical image stabilization; smart watches, fitness bands for posture detection and positioning navigation; TWS headphones for posture detection and 3D sound effects; automatic counting and motion trajectory recording for smart sports equipment such as rope

skipping, pull ropes, abdominal wheel, etc.; smart electric toothbrushes, smart water cups for posture detection; attitude control of twisty cars; combined navigation for intelligent farming, cleaning robots; platform stability and attitude measurement in intelligent industrial applications.

- Technical Parameters Brand: Senodia
- Model: SH3001
- Package: 14 Pins LGA
- Gyroscope range: 262, 131, 65.5, 32.8, 16.4 (LSB/°/s)
- Gyroscope sensitivity: 262, 131, 65.5, 32.8, 16.4 (LSB/°/s)
- Acceleration range: ±2, ±4, ±8, ±16 (g)
- Accelerometer sensitivity: 16384, 8192, 4096, 2048 (LSB/g)
- Minimum operating temperature: -40°C
- Maximum operating temperature: 100°C
- Minimum power supply voltage: 3V
- Maximum power supply voltage: 8V
- Size: 2.5 × 3.0 × 0.9 mm³



On the ATK-DLRK3588 baseboard, an I2C4 interface is used to connect a six-axis sensor SH3001. The factory system has already powered up the SH3001, but a C application needs to be written to obtain the data. The program source code is located in the development board's network drive A - Basic Materials -> 01_codes -> 06_Peripheral_Test_Source_Code. The testing method is as follows. The executable program path is located in /opt/ui/src/apps/resource/pcba/. The program in the pcba directory is used as the factory test program to detect whether the hardware functions normally.

3.9.1 Acceleration Test

Execute the following instructions to test the acceleration. Note that the development board needs to be placed horizontally. This program will start the calibration, and then the data will be obtained. The calibrated data is saved by the driver to the storage device.

```
/opt/ui/src/apps/resource/pcba/ATK-DLRK3588/accel_test
```

As shown in the figure below, from left to right, they are the accelerations of the xyz axes. You can move the development board and then take a look at the test data. If you need to stop, press Ctrl + c to end the test program.

```
root@ATK-DLRK3588:/# /opt/ui/src/apps/resource/pcba/accel_test
>>> 校准之前开发板必须处于水平放置状态 <<<
>>> 1秒后开始执行校准 <<<
[ 4251.346407] gsensor_sh3001 4-0036: 0 times, read accel data is 791, -1087, 19019
[ 4251.386679] gsensor_sh3001 4-0036: 1 times, read accel data is 775, -1036, 19016
[ 4251.423618] gsensor_sh3001 4-0036: 2 times, read accel data is 771, -1095, 18944
[ 4251.463423] gsensor_sh3001 4-0036: 3 times, read accel data is 805, -1112, 19020
[ 4251.500063] gsensor_sh3001 4-0036: 4 times, read accel data is 771, -1087, 18892
[ 4251.536903] gsensor_sh3001 4-0036: 5 times, read accel data is 772, -1034, 18988
[ 4251.573481] gsensor_sh3001 4-0036: 6 times, read accel data is 756, -1089, 18963
[ 4251.613560] gsensor_sh3001 4-0036: 7 times, read accel data is 833, -1102, 18934
[ 4251.650084] gsensor_sh3001 4-0036: 8 times, read accel data is 807, -1036, 18991
[ 4251.686756] gsensor_sh3001 4-0036: 9 times, read accel data is 868, -1089, 18890
[ 4251.723556] gsensor_sh3001 4-0036: 10 times, read accel data is 809, -1060, 18937
[ 4251.760213] gsensor_sh3001 4-0036: 11 times, read accel data is 862, -1027, 18932
[ 4251.796706] gsensor_sh3001 4-0036: 12 times, read accel data is 870, -1080, 18876
[ 4251.833375] gsensor_sh3001 4-0036: 13 times, read accel data is 848, -1115, 18951
[ 4251.870147] gsensor_sh3001 4-0036: 14 times, read accel data is 752, -1104, 18920
[ 4251.906869] gsensor_sh3001 4-0036: 15 times, read accel data is 752, -1124, 18871
[ 4251.943413] gsensor_sh3001 4-0036: 16 times, read accel data is 819, -1050, 18937
[ 4251.980137] gsensor_sh3001 4-0036: 17 times, read accel data is 792, -1139, 18903
[ 4252.016798] gsensor_sh3001 4-0036: 18 times, read accel data is 805, -1080, 18901
[ 4252.056956] gsensor_sh3001 4-0036: 19 times, read accel data is 808, -1002, 18843
[ 4252.092490] gsensor_sh3001 4-0036: accel offset is 803, -1077, 2552
>>> 校准完成 <<<
offset: [803, -1077, 2552]
[ 4252.639351] gsensor_sh3001 4-0036: sensor on: starting poll sensor data 30ms
0.048450        -0.002991       9.801794
0.008374        -0.007776       9.782654
0.025122        0.005383        9.788635
0.026318        0.012561        9.768896
-0.026917       0.000598        9.787439
0.013159        0.039478        9.799402
0.008374        0.065198        9.798804
-0.002991       0.018542        9.770691
0.016150        -0.006580       9.749158
^C
root@ATK-DLRK3588:/#
```

3.9.2 Angular velocity

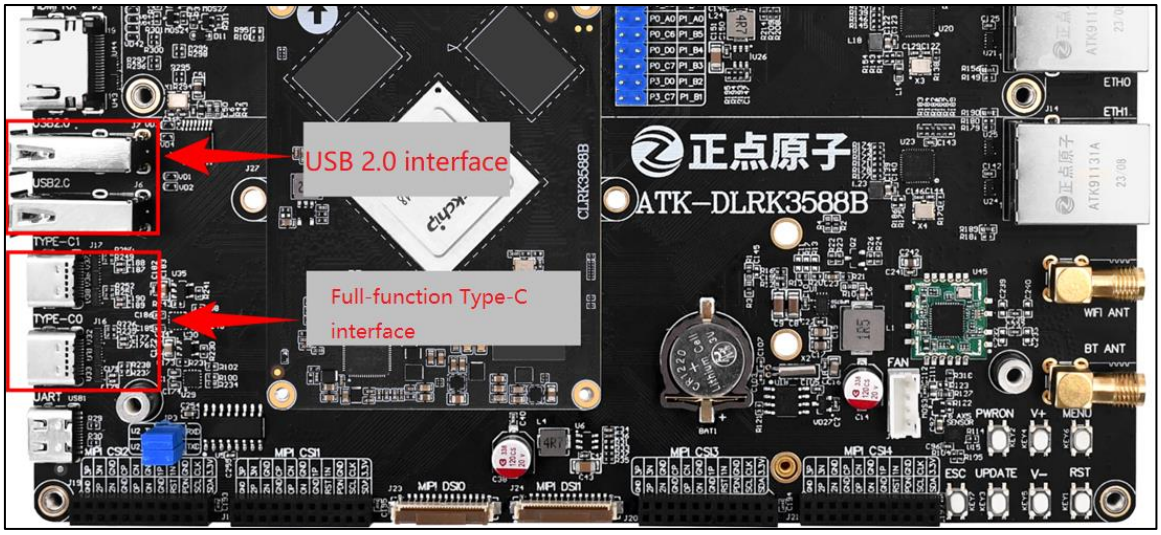
Similarly, execute the following program to obtain the angular velocity data.

/opt/ui/src/apps/resource/pcba/ATK-DLRK3588/gyro_test

```
root@ATK-DLRK3588:/# /opt/ui/src/apps/resource/pcba/gyro_test
>>> 校准之前开发板必须处于水平放置状态 <<<
>>> 1秒后开始执行校准 <<<
[ 4404.081869] gyro_sh3001 4-0036-1: 0 times, read gyro data is -10, -127, -17
[ 4404.117989] gyro_sh3001 4-0036-1: 1 times, read gyro data is -7, -132, -18
[ 4404.156051] gyro_sh3001 4-0036-1: 2 times, read gyro data is -3, -129, -16
[ 4404.191229] gyro_sh3001 4-0036-1: 3 times, read gyro data is -11, -132, -20
[ 4404.227891] gyro_sh3001 4-0036-1: 4 times, read gyro data is -12, -131, -20
[ 4404.264645] gyro_sh3001 4-0036-1: 5 times, read gyro data is -7, -136, -15
[ 4404.301370] gyro_sh3001 4-0036-1: 6 times, read gyro data is -4, -136, -17
[ 4404.337892] gyro_sh3001 4-0036-1: 7 times, read gyro data is -6, -126, -18
[ 4404.374529] gyro_sh3001 4-0036-1: 8 times, read gyro data is -7, -128, -17
[ 4404.411430] gyro_sh3001 4-0036-1: 9 times, read gyro data is -12, -133, -18
[ 4404.447911] gyro_sh3001 4-0036-1: 10 times, read gyro data is -10, -141, -20
[ 4404.484503] gyro_sh3001 4-0036-1: 11 times, read gyro data is -4, -127, -18
[ 4404.522242] gyro_sh3001 4-0036-1: 12 times, read gyro data is -7, -128, -22
[ 4404.557947] gyro_sh3001 4-0036-1: 13 times, read gyro data is -3, -132, -17
[ 4404.598014] gyro_sh3001 4-0036-1: 14 times, read gyro data is -3, -134, -19
[ 4404.634473] gyro_sh3001 4-0036-1: 15 times, read gyro data is -7, -132, -17
[ 4404.671238] gyro_sh3001 4-0036-1: 16 times, read gyro data is -10, -132, -17
[ 4404.711182] gyro_sh3001 4-0036-1: 17 times, read gyro data is -6, -132, -17
[ 4404.751244] gyro_sh3001 4-0036-1: 18 times, read gyro data is -5, -130, -19
[ 4404.788828] gyro_sh3001 4-0036-1: 19 times, read gyro data is -6, -130, -20
[ 4404.823431] gyro_sh3001 4-0036-1: gyro offset is -7, -131, -18
>>> 校准完成 <<<
offset: [-7, -131, -18]
0.427246        7.995605        0.244141
0.000000        7.995605        0.000000
0.122070        -0.122070       -0.122070
0.122070        0.366211        -0.122070
0.427246        -0.122070       0.061035
0.000000        0.122070        0.183105
-0.244141       0.000000        0.244141
0.061035        -0.244141       0.305176
-0.061035       0.122070        0.000000
-0.061035       -0.549316       0.000000
^C
root@ATK-DLRK3588:/#
```

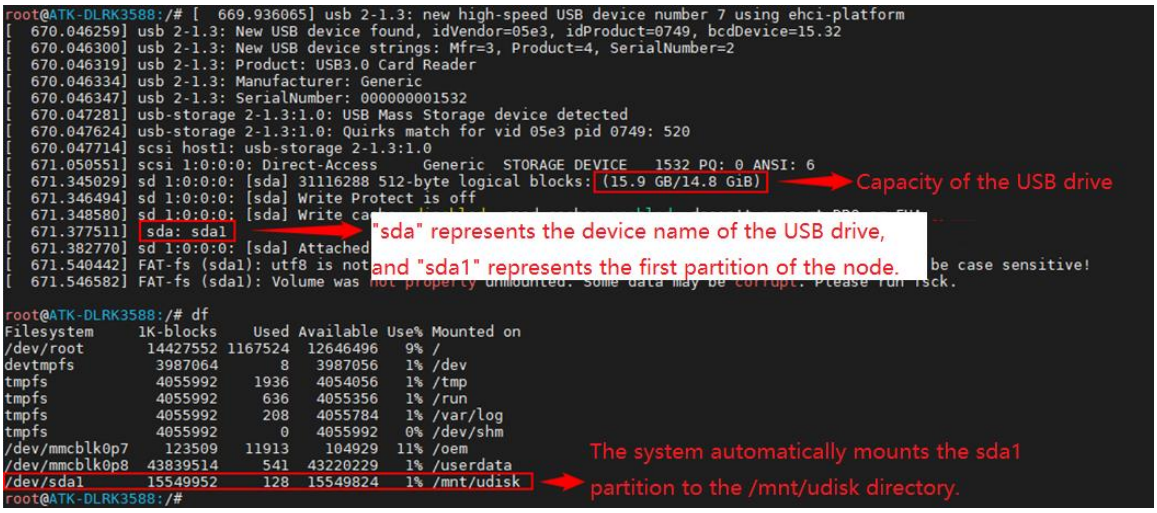
3.10 USB test

USB consists of two USB 2.0 interfaces and two full-function USB Type-C interfaces.



3.10.1 USB HOST TEST

Insert a FAT32 formatted USB drive directly into the USB interface on the ATK-DLRK3588 baseboard using a card reader (either of the two USB 2.0 interfaces will work).



We can directly enter the /mnt/udisk directory to perform read and write file operations. Note: The test of read/write speed is related to the maximum read/write speed of your personal card reader or U disk/TF card.

Read speed test:

```
hdparm -t /dev/sda1
```

```
root@ATK-DLRK3588:/# hdparm -t /dev/sda1

/dev/sda1:
Timing buffered disk reads:   80 MB in 3.02 seconds = 27045 kB/s
root@ATK-DLRK3588:/#
```

Speed writing test:

```
time dd if=/dev/zero of=/mnt/udisk/test bs=1024k count=100 conv=fdatasync
rm /mnt/udisk/test
```

Here, it represents the writing of a 100 MiB file, with a writing speed of 15.8 MB/s. In fact, the larger the file written, the closer the average writing speed is to the actual value.

```
root@ATK-DLRK3588:/# time dd if=/dev/zero of=/mnt/udisk/test bs=1024k count=100 conv=fdatasync
100+0 records in
100+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 6.65405 s, 15.8 MB/s

real    0m6.662s
user    0m0.000s
sys     0m0.385s
root@ATK-DLRK3588:/#
```

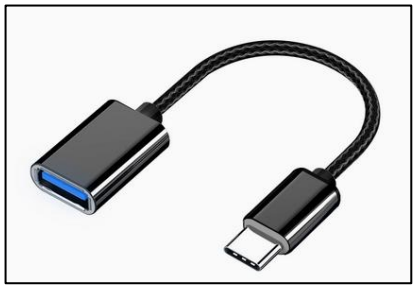
3.10.2 USB TypeC Slave Device Test

The full-function interface USB TypeC interface USB TypeC0 on the ATK-DLRK3588 baseboard can use the ADB function, acting as a slave device. Please refer to the ADB usage manual written by ALIENTEK. USB TypeC1 does not have the ADB function.

3.10.3 USB TypeC Host Device Test

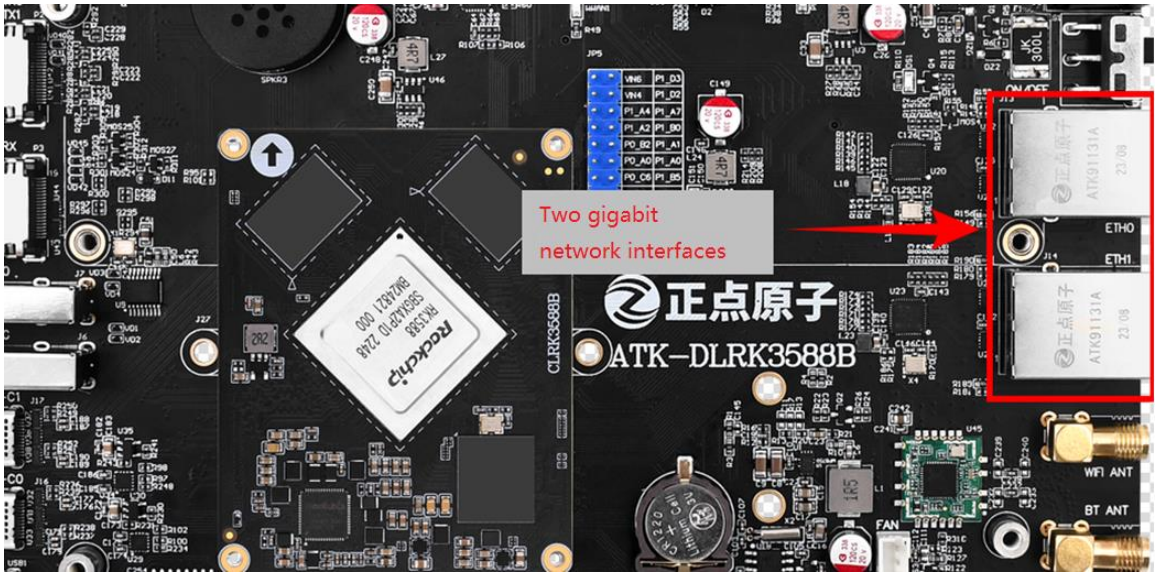
Connect a USB to Type-C adapter (if the user does not have this adapter, this experiment can be skipped) to a USB flash drive and plug it into the USB TypeC0/USB TypeC1 interface. Follow the same test procedure as in 3.10.1 section.

The USB to Type-C adapter is as follows. This adapter should be purchased as needed by yourself.



3.11 Network Testing

The ATK-DLRK3588 motherboard is equipped with two gigabit network chips, which are capable of adapting to 10/100/1000M. The network can be tested using the following commands in the file system.



Plug the network cable into one of the two network ports and make sure the network can connect to the internet. Then check the IP address obtained from the network.

ifconfig

```
root@ATK-DLRK3588:/# ifconfig
eth0      Link encap:Ethernet  HWaddr 06:B6:A4:B1:51:87
          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:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:93

eth1      Link encap:Ethernet  HWaddr 02:B6:A4:B1:51:87
          inet addr:192.168.6.58  Bcast:192.168.6.255  Mask:255.255.255.0
          inet6 addr: fe80::b6:a4ff:feb1:5187/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:308 errors:0 dropped:0 overruns:0 frame:0
          TX packets:87 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:25074 (24.4 KiB)  TX bytes:9171 (8.9 KiB)
          Interrupt:148

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:54 errors:0 dropped:0 overruns:0 frame:0
          TX packets:54 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4451 (4.3 KiB)  TX bytes:4451 (4.3 KiB)

root@ATK-DLRK3588:/#
```

Check if the network is accessible. If there is a data response, it indicates that the network function is normal. Press Ctrl+c to terminate the command.

ping www.baidu.com

```
root@ATK-DLRK3588:/# ping www.baidu.com
PING www.a.shifen.com (183.2.172.185) 56(84) bytes of data.
64 bytes from 183.2.172.185 (183.2.172.185): icmp_seq=1 ttl=52 time=3.61 ms
64 bytes from 183.2.172.185 (183.2.172.185): icmp_seq=2 ttl=52 time=3.79 ms
64 bytes from 183.2.172.185 (183.2.172.185): icmp_seq=3 ttl=52 time=3.80 ms
64 bytes from 183.2.172.185 (183.2.172.185): icmp_seq=4 ttl=52 time=4.04 ms
64 bytes from 183.2.172.185 (183.2.172.185): icmp_seq=5 ttl=52 time=3.63 ms
^C
--- www.a.shifen.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 3.607/3.774/4.043/0.156 ms
root@ATK-DLRK3588:/#
```

At the same time, the other network port was also tested in the same way. Plug the network cable into the other port and repeat the above operation.

Test whether it is a gigabit network. Pay attention to using a gigabit network cable, a gigabit router (most home routers are of the hundred-megabit type) or a gigabit switch. The PC (computer) network card must be a gigabit network card. Otherwise, the speed measured or identified may be hundred-megabit! As shown in the figure below, seeing "1Gbps/Full" indicates that it is a gigabit network.

```
root@ATK-DLRK3588:/# [  277.306081] rk_gmac-dwmac fe1b0000.ethernet eth1: Link is Up - 1Gbps/Full - flow control rx/tx
[  277.306192] IPv6: ADDRCONF(NETDEV_CHANGE): eth1: link becomes ready

root@ATK-DLRK3588:/#
```

The iperf3 command can be used to test the connection speed of a gigabit network.

Set Ubuntu as the server. Ubuntu and the development board must be in the same local area network. If Ubuntu does not have the iperf3 command, you need to enter "sudo apt-get install iperf3" to install it.

View Ubuntu IP

ifconfig

```
alientek@alientek:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.6.45  netmask 255.255.255.0  broadcast 192.168.6.255
        inet6 fe80::1582:a35c:fb40:d654  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:1c:f2:34  txqueuelen 1000  (以太网)
        RX packets 253499  bytes 27103856 (27.1 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 7627  bytes 1362120 (1.3 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (本地环回)
        RX packets 737  bytes 73007 (73.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 737  bytes 73007 (73.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

alientek@alientek:~$
```

Enter "iperf3 -s" to set Ubuntu as the server.

iperf3 -s

```
alientek@alientek:~$ iperf3 -s
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

ATK-DLRK3588 is the client. By using the iperf3 command, simply input the IP address of the server (Ubuntu) to proceed.

iperf3 -c 192.168.6.45 -i 1

Command parsing:

- -s: Indicates that the current device acts as a server.
- -c: Indicates that the current device acts as a client.
- 192.168.6.45: The IP address of the server.
- -i: Trigger period.

```
root@ATK-DLRK3588:/# iperf3 -c 192.168.6.45 -i 1
Connecting to host 192.168.6.45, port 5201
[  5] local 192.168.6.58 port 52366 connected to 192.168.6.45 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec   113 MBytes   948 Mbits/sec    0    465 KBytes
[  5]   1.00-2.00   sec   111 MBytes   934 Mbits/sec    0    515 KBytes
[  5]   2.00-3.00   sec   112 MBytes   940 Mbits/sec    0    515 KBytes
[  5]   3.00-4.00   sec   112 MBytes   943 Mbits/sec    0    540 KBytes
[  5]   4.00-5.00   sec   112 MBytes   939 Mbits/sec    0    540 KBytes
[  5]   5.00-6.00   sec   112 MBytes   943 Mbits/sec    0    540 KBytes
[  5]   6.00-7.00   sec   112 MBytes   941 Mbits/sec    0    540 KBytes
[  5]   7.00-8.00   sec   112 MBytes   942 Mbits/sec    0    540 KBytes
[  5]   8.00-9.00   sec   112 MBytes   941 Mbits/sec    0    540 KBytes
[  5]   9.00-10.00  sec   112 MBytes   938 Mbits/sec    0    609 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec  1.10 GBytes   941 Mbits/sec    0             sender
[  5]   0.00-10.05  sec  1.09 GBytes   936 Mbits/sec                  receiver

iperf Done.
root@ATK-DLRK3588:/#
```

The Bitrate received and transmitted during the test were approximately 900 Mbits/sec, which is the speed of a gigabit network (please do not run any CPU-consuming applications before the test to avoid affecting the test speed). If it is not so, please check whether a gigabit network cable is used, whether the network card of the PC is a gigabit card, whether the router is gigabit, etc. No device with a hundred-megabit speed can be present; otherwise, the result of the test will be a hundred-megabit network.

3.12 Onboard USB WIFI Test

The ATK-DLRK3588 board is equipped with a RTL8733BU chip, which is a dual-function chip for WIFI and Bluetooth. After powering on, use the /etc/init.d/S36wifibt-init.sh script to initialize the WIFI driver and Bluetooth driver.

3.12.1 Station mode

The Station mode of WiFi is a working mode that enables WiFi devices to connect to a WiFi network and communicate with other devices. In Station mode, WiFi devices can act as a client to connect to an existing WiFi network, which can be a wireless network provided by a home, office, or public place. After connecting to the WiFi network, the device can transmit and communicate data with other devices through this network. Generally, Station mode is suitable for scenarios where devices need to be connected to an existing WiFi network, such as mobile phones, computers, smart home devices, etc.

Note that wlan0 is not enabled by default.

rfkill unblock 2 # Unlock the radio frequency, wlan0 is number 2. If you need to view the corresponding device, please enter "rfkill".

ifconfig wlan0 up # Turn on wlan0. If it is not turned on, you can input this command.

ifconfig # Check if wlan0 is turned on. You can see the following picture indicating it is turned on.

```
root@ATK-DLRK3588:/# rfkill unblock 2
root@ATK-DLRK3588:/# [  947.380308] start_addr=(0x0), end_addr=(0x20000), buffer_size=(0x20000), smp_number_max=(16384)
ifconfig wlan0 up
root@ATK-DLRK3588:/# ifconfig
eth0      Link encap:Ethernet  HWaddr 06:B6:A4:B1:51:87
          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:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:93

eth1      Link encap:Ethernet  HWaddr 02:B6:A4:B1:51:87
          inet addr:192.168.6.58  Bcast:192.168.6.255  Mask:255.255.255.0
          inet6 addr: fe80::b6:a4ff:feb1:5187/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:92963 errors:0 dropped:0 overruns:0 frame:0
          TX packets:35219 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6178640 (5.8 MiB)  TX bytes:2274741893 (2.1 GiB)
          Interrupt:148

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:54 errors:0 dropped:0 overruns:0 frame:0
          TX packets:54 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4470 (4.3 KiB)  TX bytes:4470 (4.3 KiB)

wlan0     Link encap:Ethernet  HWaddr F0:A8:82:3A:5C:68
          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:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@ATK-DLRK3588:/#
```

We can use the following instructions to connect to Wi-Fi.

connmanctl	# Enter the WIIF operation terminal
connmanctl> enable wifi	# Enable WIFI. It is already enabled by default.
connmanctl> agent on	# Activate the proxy
connmanctl> scan wifi	# Turn on WIFI scanning. Multiple scans are available.
connmanctl> services	# List of scanned WIFI networks

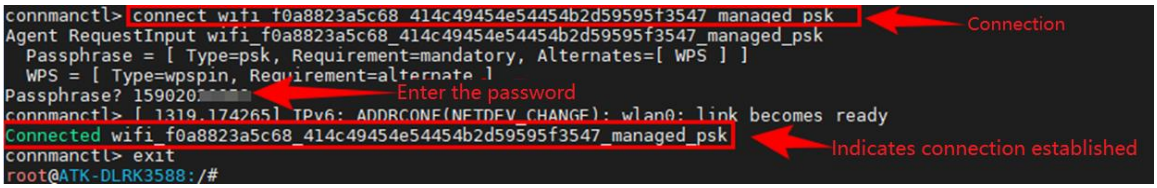
The command execution result is as follows:

```
root@ATK-DLRK3588:/# connmanctl
connmanctl> enable wifi
wifi is already enabled
connmanctl> agent on
Agent registered
connmanctl> scan wifi
Scan completed for wifi
connmanctl> services
*AO Wired                ethernet_02b6a4b15187_cable
    wifi6_5g             wifi_f0a8823a5c68_77696669365f3567_managed_psk
    ALIENTEK-YY_5G       wifi_f0a8823a5c68_414c49454e54454b2d59595f3547_managed_psk
    ALIENTEK-YY_5G_Wi-Fi5 wifi_f0a8823a5c68_414c49454e54454b2d59595f35475f57692d466935_managed_psk
    ChinaNet-ymyk-5G     wifi_f0a8823a5c68_4368696e614e65742d796d796b2d3547_managed_psk
                         wifi_f0a8823a5c68_hidden_managed_psk
    ALIENTEK-YY          wifi_f0a8823a5c68_414c49454e54454b2d5959_managed_psk
    RICH901-5G           wifi_f0a8823a5c68_524943483930312d3547_managed_psk
    ChinaNet-ymyk        wifi_f0a8823a5c68_4368696e614e65742d796d796b_managed_psk
    Tenda_480398         wifi_f0a8823a5c68_54656e64615f343830333938_managed_psk
    ALIENTEK-WX_5G       wifi_f0a8823a5c68_414c49454e54454b2d57585f3547_managed_psk
    RICH901              wifi_f0a8823a5c68_52494348393031_managed_psk
    ALIENTEK-WX          wifi_f0a8823a5c68_414c49454e54454b2d5758_managed_psk
connmanctl>
```

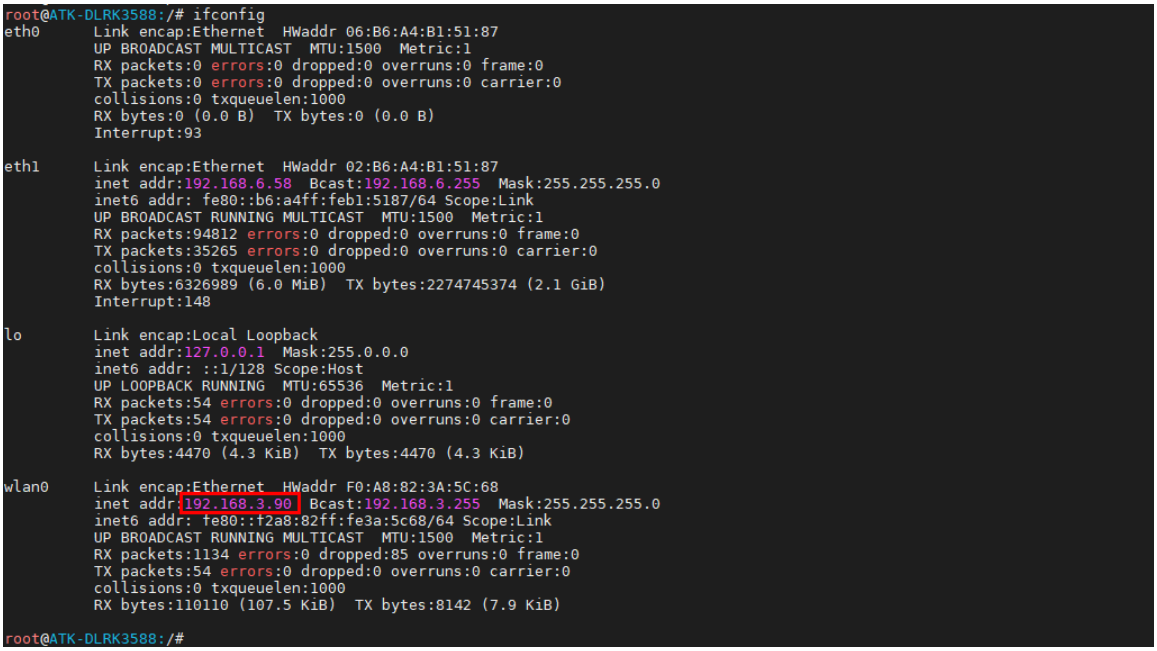
From the scan results to the WIFI list, use "connect XXXX" to connect, then enter the password and press Enter to confirm. For example, if I want to connect to "ALIENTEK-YY_5G", (if using the hotspot of an IOS phone, when turning on the IOS hotspot, you may need to select compatibility, and also note that if the WIFI signal is very weak (too far away), or if the password is incorrect, it may not

be possible to connect to the WIFI. During testing, please confirm these two factors). The operation result is as follows: command:

```
connect wifi_f0a8823a5c68_414c49454e54454b2d59595f3547_managed_psk    # Connect
exit                                                                  # Exit terminal
```

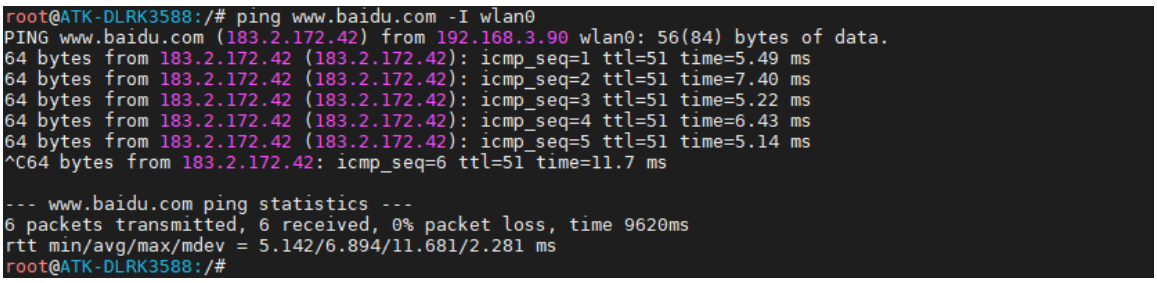


After the connection is successful, you can use the "ifconfig" command to check the IP address obtained by our "wlan0".



You can test the connectivity by pinging Baidu, or you can ping the gateway to test the connectivity of the WIFI.

```
ifconfig eth1 down
# Since the gateway of wlan0 is different from that of eth1, it needs to be turned off
ping www.baidu.com -I wlan0          # ping Baidu to test
ping 192.168.3.1 -I wlan0            # ping the gateway. Please replace it with your own gateway, 192.168.x
```



3.12.2 SoftAP mode

The WiFi SoftAP mode is a working mode that allows WiFi devices to operate like a wireless router. In SoftAP mode, the WiFi device can act as a hotspot, allowing other devices to connect and use the network services it provides. Usually, the SoftAP mode is suitable for scenarios where devices need to be transformed into a wireless hotspot, such as smart home devices, wireless printers, game consoles, etc. In SoftAP mode, the device can set parameters such as SSID and password, allowing other devices to connect and conduct data transmission and communication through the network service provided by it. Since the SoftAP mode enables the device to work like a wireless router, it is usually used in some special scenarios, such as network debugging, data collection, etc.

The scripts written by ALIENTEK for enabling the hotspot are all placed in the /root/shell/wifi directory, as shown in the following figure.

```
root@ATK-DLRK3588:/root/shell/wifi# ls
autoconnect.sh  dnsmasq.conf  hostapd.conf  wifi_apmode.sh
root@ATK-DLRK3588:/root/shell/wifi#
```

The content of the dnsmasq.conf file is as follows:

```
interface=wlan0
dhcp-range=192.168.4.2,192.168.4.254,255.255.255.0,24h
port=49153
```

The content of the hostapd.conf file is as follows:

```
interface=wlan0
ssid=alientek_softap
driver=nl80211
channel=6
hw_mode=g
ignore_broadcast_ssid=0
auth_algs=1
wpa=3
wpa_passphrase=12345678
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

The content of the "wifi_apmode.sh" file is as follows:

```
# Note: It might be rfkill1. If 5G is not configured, by default 5G is configured, while rfkill0 is for Bluetooth.
echo 0 > /sys/class/rfkill/rfkill2/state
sleep 2
echo 1 > /sys/class/rfkill/rfkill2/state
sleep 2

if [[ "$(pidof dnsmasq)" != "" ]]
then
        kill -9 $(pidof dnsmasq)
fi
```

```
if [[ "$(pidof hostapd)" != "" ]]
then
        kill -9 $(pidof hostapd)
fi
connmanctl disable wifi
rfkill unblock all
ifconfig wlan0 down
sleep 1
ifconfig wlan0 up
sleep 1
dnsmasq -C dnsmasq.conf
ifconfig wlan0 192.168.4.1
hostapd hostapd.conf -B
```

Enter the directory "/root/shell/wifi/" and execute the following command to turn on the hotspot. If the hotspot is successfully activated, it will look like the image below.

```
./wifi_apmode.sh
```

```
root@ATK-DLRK3588:/root/shell/wifi# ./wifi_apmode.sh
Disable wifi
[ 2907.623041] [BT_RFKILL]: bt turn on power
Configuration file: hostapd.conf
Using interface wlan0 with hwaddr f0:a8:82:3a:5c:68 and ssid "alientek_softap"
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
[ 2911.151477] IPv6: ADDRCONF(NETDEV_CHANGE): wlan0: link becomes ready
root@ATK-DLRK3588:/root/shell/wifi#
```

Just connect to the test hotspot named "alientek_softap" using your mobile phone. The password is "12345678" as configured in the "hostapd.conf" file. As shown in the picture below, the author successfully connected to the hotspot emitted by the board using his mobile phone.

As can be seen, the IP address of the board was set by us in the wifi_apmode.sh script as wlan0 with the IP address 192.168.4., which acts as the gateway, i.e., the router. Moreover, the mobile phone also received an IP address of 192.168.4.4 from the development board. Note that this hotspot can be connected to, but it is not for internet access. This demonstration is limited to local network connection. You can write network communication programs to exchange information. The hotspot has many uses. For example, if you want a nearby device to communicate with your board, it can be done through this

hotspot for information exchange. For details, please refer to the network programming section of the Qt or C application tutorials.

After enabling the AP mode, how do we disable it? In the previous script, we used hostapd and dnsmasq to enable the AP mode and made them run as background processes. Therefore, we need to stop these two processes. This will disable the AP mode. Execute the following command to stop the processes.

```
killall hostapd dnsmasq
```

```
root@ATK-DLRK3588:/root/shell/wifi# killall hostapd dnsmasq
root@ATK-DLRK3588:/root/shell/wifi#
```

You can also directly use the "iw" command to set the client mode of the Wi-Fi. However, it is better to also close the above two processes to prevent resource occupation!

```
iw dev wlan0 set type managed
```

```
root@ATK-DLRK3588:/root/shell/wifi# iw dev wlan0 set type managed
root@ATK-DLRK3588:/root/shell/wifi#
```

At this point, we use the "iw dev" command to check the wireless network interface. We can see that the type of the wireless network is "managed", which indicates the client mode, which is also the "station" mode we mentioned earlier.

```
iw dev
```

```
root@ATK-DLRK3588:/root/shell/wifi# iw dev
phy#0
        Interface wlan0
                ifindex 5
                wdev 0x1
                addr f0:a8:82:3a:5c:68
                type managed
                txpower -100.00 dBm
root@ATK-DLRK3588:/root/shell/wifi#
```

3.13 Onboard USB Bluetooth Test

Bluetooth development relies on Buildroot. Files such as rtk_btusb.ko, which is the Bluetooth driver module, are generated only under the condition that the correct Wi-Fi/BT module is selected through Buildroot configuration.

There are numerous Bluetooth protocols, as follows.

- L2CAP: Logical Link Control and Adaptation Protocol, used to establish and maintain logical link connections.
- RFCOMM: Remote Serial Communication Protocol, which maps the data transmission of serial devices onto Bluetooth.
- SDP: Service Discovery Protocol, used to find and describe available Bluetooth services.
- GAP: General Access Protocol, defines the basic connection and communication rules between devices.
- GATT: General Attribute Protocol, used to establish and manage the exchange of attribute data between devices.
- ATT: Attribute Protocol, used for transmitting attribute values between Bluetooth devices.
- HFP: Mobile Phone Bluetooth Headset Protocol, used for audio communication between mobile phones and Bluetooth headsets.
- A2DP: Advanced Audio Distribution Protocol, used for transmitting high-quality audio data between Bluetooth devices.
- AVRCP: Audio/Video Remote Control Protocol, used for transmitting media control commands between Bluetooth devices.

My skills are limited, Bluetooth is a very extensive topic. In this demonstration, only the usage method of RFCOMM for remote serial port will be shown. Other protocols can be studied independently.

The board-level Bluetooth driver module/lib/modules/rtk_btusb.ko is available in the Buildroot system path.

3.13.1 Bluetooth initialization

Bluetooth initialization starts automatically upon boot. The file /etc/init.d/S36wifibt-init.sh initializes the Bluetooth driver and the WIFI driver.

3.13.2 Bluetooth rfcomm

Bluetooth RFCOMM is a Bluetooth protocol that provides a way for serial data transmission, enabling data transfer between Bluetooth devices. RFCOMM can convert serial data into Bluetooth data packets, thus achieving wireless transmission.

RFCOMM offers a concept of a virtual serial port, allowing data transmission between Bluetooth devices to be conducted like a serial port. This method is widely used in Bluetooth headphones, Bluetooth printers, and other devices.

When using RFCOMM for data transmission, a Bluetooth connection needs to be established first, and the RFCOMM channel needs to be specified. Then, on both ends of the connected devices, data can be sent and received through RFCOMM. This method is suitable for scenarios requiring serial port communication, such as wireless controllers, wireless sensors, etc.

The author has written a script for initiating the Bluetooth RFCOMM connection to establish an RFCOMM channel, which is located at /root/shell/bluetooth/rfcomm_init.sh. The content of the script is as follows.

```
if [[ "$(pidof rfcomm)" != "" ]]
then
        kill -9 $(pidof rfcomm)
fi
hciconfig hci0 piscan
hciconfig hci0 noauth
sleep 1
sdptool add SP
sleep 1
rfcomm watch hci0 &
echo "rfcomm init finished! please use bluetoothctl cmd to pair your device and then connect it!"
```

Executing the rfcomm_init.sh script, it can be seen that channel 1 has been created and is now waiting for a connection.

```
rfcomm_init.sh
```

```
root@ATK-DLRK3588:/# /root/shell/bluetooth/rfcomm_init.sh
rfcomm init finished! please use bluetoothctl cmd to pair your device and then connect it!
root@ATK-DLRK3588:/# Waiting for connection on channel 1

root@ATK-DLRK3588:/#
```

At this point, we need to perform a Bluetooth pairing, and then connect using the "Bluetooth Debugging Wizard" app on an Android phone. There is no similar software for Apple phones.

3.13.3 Bluetooth Pairing

Enter the following command to enter the bluetoothctl interactive mode.

```
bluetoothctl
```

Execute the following instructions in sequence.

```
power on                    # turn on the power
agent on                    # Activate the proxy
discoverable on             # Enable detectability. By default, there is a discoverable duration. If you want it to remain detectable indefinitely, please set it as "discoverable-timeout 0, 0" which represents an infinite duration.
scan on                     # Turn on the scanning device
```

Once your Bluetooth device is detected, input "scan off" to stop the scanning process.

```
scan off
```

```
root@ATK-DLRK3588:/# bluetoothctl
Agent registered
[CHG] Controller F0:A8:82:3A:5C:83 Pairable: yes
[bluetooth]# power on
Changing power on succeeded
[bluetooth]# agent on
Agent is already registered
[bluetooth]# discoverable-timeout 0
Changing discoverable-timeout 0 succeeded
[CHG] Controller F0:A8:82:3A:5C:83 DiscoverableTimeout: 0x00000000
[bluetooth]# scan on
[bluetooth]# [ 1256.712422] rtk_btcoex: hci (periodic)inq start
Discovery started
[CHG] Controller F0:A8:82:3A:5C:83 Discovering: yes
[NEW] Device 60:AB:67:65:93:E8 蓝牙测试
[bluetooth]# scan off
[bluetooth]# [ 1265.490060] rtk_btcoex: hci (periodic)inq cancel/exit
Discovery stopped
[CHG] Device 60:AB:67:65:93:E8 RSSI is nil
[CHG] Controller F0:A8:82:3A:5C:83 Discovering: no
[bluetooth]#
```

The author scanned the Bluetooth MAC address of the Android phone, as shown in the picture above. After confirmation, the pairing process can be carried out.

Enter the following instructions to complete the pairing.

pair 60:AB:67:65:93:E8 # Please enter your own Bluetooth MAC address.

```
[bluetooth]# pair 60:AB:67:65:93:E8
Attempting to pair with 60:AB:67:65:93:E8
[bluetooth]# [ 1939.186307] rtk_btcoex: hci create connection, start paging
[ 1939.814279] rtk_btcoex: connected, handle 0005, status 0x00
[ 1939.814308] rtk_btcoex: Page success
[CHG] Device 60:AB:67:65:93:E8 Connected: yes
[蓝牙测试]# [ 1939.870304] rtk_btcoex: io capability request
Request confirmation
[agent] Confirm passkey 326509 (yes/no): yes
[蓝牙测试]# [ 1951.877306] rtk_btcoex: link key notify
[CHG] Device 60:AB:67:65:93:E8 Modalias: bluetooth:v038Fp1200d1436
[CHG] Device 60:AB:67:65:93:E8 UUIDs: 00001105-0000-1000-8000-00805f9b34fb
[CHG] Device 60:AB:67:65:93:E8 UUIDs: 0000110a-0000-1000-8000-00805f9b34fb
[CHG] Device 60:AB:67:65:93:E8 UUIDs: 0000110c-0000-1000-8000-00805f9b34fb
[CHG] Device 60:AB:67:65:93:E8 UUIDs: 00001112-0000-1000-8000-00805f9b34fb
[CHG] Device 60:AB:67:65:93:E8 UUIDs: 00001115-0000-1000-8000-00805f9b34fb
[CHG] Device 60:AB:67:65:93:E8 UUIDs: 00001116-0000-1000-8000-00805f9b34fb
[CHG] Device 60:AB:67:65:93:E8 UUIDs: 0000111f-0000-1000-8000-00805f9b34fb
[CHG] Device 60:AB:67:65:93:E8 UUIDs: 0000112f-0000-1000-8000-00805f9b34fb
[CHG] Device 60:AB:67:65:93:E8 UUIDs: 00001132-0000-1000-8000-00805f9b34fb
[CHG] Device 60:AB:67:65:93:E8 UUIDs: 00001200-0000-1000-8000-00805f9b34fb
[CHG] Device 60:AB:67:65:93:E8 UUIDs: 00001800-0000-1000-8000-00805f9b34fb
[CHG] Device 60:AB:67:65:93:E8 UUIDs: 00001801-0000-1000-8000-00805f9b34fb
[CHG] Device 60:AB:67:65:93:E8 UUIDs: 98b97136-36a2-11ea-8467-484d7e99a198
[CHG] Device 60:AB:67:65:93:E8 ServicesResolved: yes
[CHG] Device 60:AB:67:65:93:E8 Paired: yes
Pairing successful
[蓝牙测试]# [ 1956.231407] rtk_btcoex: HCI Disconnect, handle 0005, reason 0x13
[ 1956.274272] rtk_btcoex: disconn cmpl evt: status 0x00, handle 0005, reason 0x16
[CHG] Device 60:AB:67:65:93:E8 ServicesResolved: no
[CHG] Device 60:AB:67:65:93:E8 Connected: no
[bluetooth]#
```

Input "yes"

The pairing code also pops up on the mobile device.

Enter the following instruction: Trust MAC.

trust 60:AB:67:65:93:E8 # Please enter your own Bluetooth MAC address.

```
[bluetooth]# trust 60:AB:67:65:93:E8
[CHG] Device 60:AB:67:65:93:E8 Trusted: yes
Changing 60:AB:67:65:93:E8 trust succeeded
[bluetooth]#
```

3.13.4 Bluetooth communication test

Install Bluetooth Debugging Tool on an Android phone. After opening the software, paired and unpaired Bluetooth devices will be displayed in the list (if not found, it might be necessary to turn on the scanning again). As shown in the picture, the discovery mode is turned off. To enable it, either execute "discoverable on" or exit the interactive terminal and execute "hciconfig hci0 piscan" again to re-enable it.

```
[bluetooth]# [ 3099.120701] Bluetooth: hu 00000000229fa317 retransmitting 1 pkts
[CHG] Controller FF:FF:FF:FF:FF:94 Discoverable: no
[bluetooth]#
```

Bluetooth Debugging Kit. After opening the software, if the Bluetooth has been paired, it will display as shown in the picture below.

Click the link. The default uuid is sufficient. You can see the following picture. It has been connected. Data transmission and reception can now be carried out.

The board can receive data by using the "cat" command to retrieve the data. Note that "/dev/rfcomm0" will be generated only after a successful connection.

```
cat /dev/rfcomm0
```

Then, data is sent from the mobile phone to the development board, as shown in the figure below.

The data received by the development board is as follows.

```
root@ATK-DLRK3588:/root/shell/bluetooth# cat /dev/rfcomm0
正点原子
```

Similarly, we can send data to the mobile phone. Execute the following instructions.

```
echo "正点原子" > /dev/rfcomm0
```

```
root@ATK-DLRK3588:/root/shell/bluetooth# echo "正点原子" > /dev/rfcomm0
root@ATK-DLRK3588:/root/shell/bluetooth#
```

3.13.5 Bluetooth music

After pairing is completed, connect the Android/iOS phone. The phone plays music, and the development board acts as a Bluetooth headset, capable of playing the audio from the phone. On the phone, click on the Bluetooth connection for the development board as shown in the following picture. The development board is then recognized as a headset.

3.14 TF(SD) Card & eMMC Testing

Instruction Reminder:

The "time" command is commonly used to measure the running time of a command. "dd" is used for copying, reading from the "if" (input file) file and writing to the "of" (output file) specified file. "bs" is the size of each write block, and "count" is the number of read/write blocks. "if=/dev/zero" does not generate I/O, that is, it can continuously output data, so it can be used to test pure write speed.

3.14.1 SD card test

After the development board is powered on, insert the TF card into the card slot on the bottom plate of the development board. The following information will be printed. The development board will automatically mount the TF card. (Note: A TF card in FAT32 format is required.)

Use the command "df -h" to check the partition mounted by SD. The result is as follows:

```
root@ATK-DLRK3588:/# [  464.585351] dwmmc_rockchip fe2c0000.mmc: could not set regulator OCR (-22)
[  464.585394] dwmmc_rockchip fe2c0000.mmc: failed to enable vmmc regulator
[  464.808319] mmc_host mmc1: Bus speed (slot 0) = 148500000Hz (slot req 150000000Hz, actual 148500000HZ div = 0)
[  464.944302] dwmmc_rockchip fe2c0000.mmc: Successfully tuned phase to 266
[  464.944369] mmc1: new ultra high speed SDR104 SDHC card at address aaaa
[  464.945579] mmcblk1: mmc1:aaaa SC16G 14.8 GiB
[  464.949126]  mmcblk1: p1
[  465.047958] FAT-fs (mmcblk1p1): utf8 is not a recommended IO charset for FAT filesystems, filesystem will be case sensitive!
[  465.053076] FAT-fs (mmcblk1p1): Volume was not properly unmounted. Some data may be corrupt. Please run fsck.
df
Filesystem     1K-blocks    Used Available Use% Mounted on
/dev/root       14424824 1170008  12640464   9% /
devtmpfs         8098412       8   8098404   1% /dev
tmpfs            8167340    1984   8165356   1% /tmp
tmpfs            8167340     580   8166760   1% /run
tmpfs            8167340     180   8167160   1% /var/log
tmpfs            8167340       0   8167340   0% /dev/shm
/dev/mmcblk0p7    123509   11913    104929  11% /oem
/dev/mmcblk0p8 102233381     454 101010354   1% /userdata
/dev/mmcblk1p1  15549952  102528  15447424   1% /mnt/sdcard
root@ATK-DLRK3588:/#
```

Read speed test is often used to test the reading speed of hard drives. The commands for this test include hdparm and dd. In this demonstration, we only show the reading speed of the TF card using hdparm. The data sampling time is too short and the data volume is small. The speed provided here is for reference only.

```
hdparm -t /dev/mmcblk1p1
```

```
root@ATK-DLRK3588:/# hdparm -t /dev/mmcblk1p1

/dev/mmcblk1p1:
Timing buffered disk reads:  192 MB in 3.00 seconds = 65401 kB/s
root@ATK-DLRK3588:/#
```

It can be seen that the reading speed is 65401 kB/s.

Write speed test

```
time dd if=/dev/zero of=/mnt/sdcard/test bs=1024k count=50 conv=fdatasync
```

```
root@ATK-DLRK3588:/# time dd if=/dev/zero of=/mnt/sdcard/test bs=1024k count=50 conv=fdatasync
50+0 records in
50+0 records out
52428800 bytes (52 MB, 50 MiB) copied, 3.11664 s, 16.8 MB/s

real    0m3.134s
user    0m0.003s
sys     0m0.145s
root@ATK-DLRK3588:/#
```

Use the "dd" command for reading. Due to the Linux system mechanism, generally there is no need to deliberately release the already used cache. These cache contents can increase the reading and writing speed of the file.

Execute the following command to clear the cache.

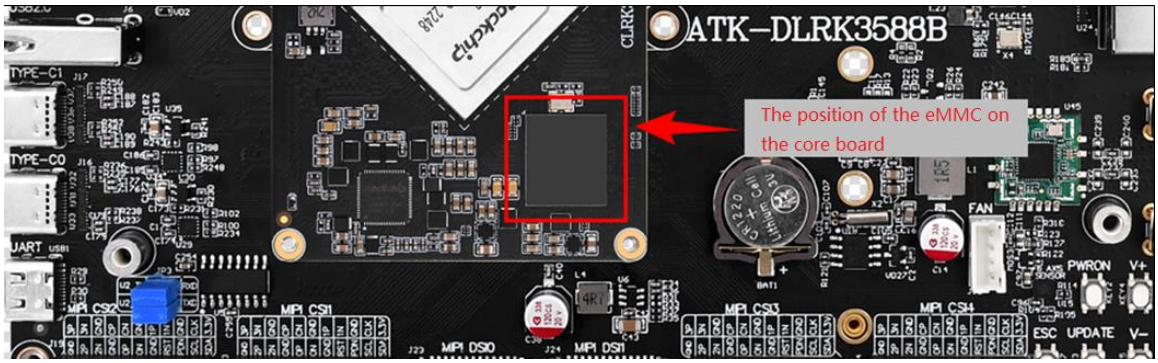
```
echo 3 > /proc/sys/vm/drop_caches                    # Clear cache
time dd if=/mnt/sdcard/test of=/dev/null bs=1024k    # Read speed test
rm /mnt/sdcard3/test                                 # Delete this file after reading it.
```

```
root@ATK-DLRK3588:/# echo 3 > /proc/sys/vm/drop_caches
[  739.620513] sh (1404): drop_caches: 3
root@ATK-DLRK3588:/# time dd if=/mnt/sdcard/test of=/dev/null bs=1024k
50+0 records in
50+0 records out
52428800 bytes (52 MB, 50 MiB) copied, 0.792628 s, 66.1 MB/s

real    0m0.801s
user    0m0.005s
sys     0m0.099s
root@ATK-DLRK3588:/# rm /mnt/sdcard/test
```

It can be seen that the reading speed is 66.1MB/s. This speed is for reference only. The larger the file being read, the more accurate the speed will be.

3.14.2 EMMC Test

As a storage device, eMMC stores the Linux system. All the applications developed by users in the future will be stored in this device (if you do not connect an external storage device).



By using the df -h command, we can determine the partitions mounted on the eMMC. The result is as follows. In fact, the eMMC storage has been divided into many partitions, and the one mounted under the root directory "/ " is also an eMMC partition. Here, we take the partition directory mmcblk0p8 mounted as /userdata as an example.

```
df
```

```
root@ATK-DLRK3588:/# df
Filesystem     1K-blocks    Used Available Use% Mounted on
/dev/root       14427552 1167732  12646288   9% /
devtmpfs         3987064       8   3987056   1% /dev
tmpfs            4055992    1916   4054076   1% /tmp
tmpfs            4055992     624   4055368   1% /run
tmpfs            4055992     216   4055776   1% /var/log
tmpfs            4055992       0   4055992   0% /dev/shm
/dev/mmcblk0p7    123509   11913    104929  11% /oem
/dev/mmcblk0p8  43839514     588  43220182   1% /userdata
root@ATK-DLRK3588:/#
```

Read speed test

```
hdparm -t /dev/mmcblk0p8
```

```
root@ATK-DLRK3588:/# hdparm -t /dev/mmcblk0p8

/dev/mmcblk0p8:
Timing buffered disk reads:  928 MB in 3.00 seconds = 316719 kB/s
root@ATK-DLRK3588:/#
```

It can be seen that the read speed is 316719 KB/s.

Write speed test

```
time dd if=/dev/zero of=/userdata/test bs=1024k count=50 conv=fdatasync
```

```
root@ATK-DLRK3588:/# time dd if=/dev/zero of=/userdata/test bs=1024k count=50 conv=fdatasync
50+0 records in
50+0 records out
52428800 bytes (52 MB, 50 MiB) copied, 0.364802 s, 144 MB/s

real    0m0.379s
user    0m0.003s
sys     0m0.104s
root@ATK-DLRK3588:/#
```

It can be seen that the writing speed is 144MB/s.

Use the dd command for reading. Due to the mechanism of the Linux system, generally there is no need to specifically release the used cache. These cache contents can increase the reading and writing speed of the file.

Execute the following command to clear the cache.

```
echo 3 > /proc/sys/vm/drop_caches                  # Clear cache
time dd if=/userdata/test of=/dev/null bs=1024k    # Read speed test
rm /userdata/test                                  # Delete this file after reading it.
```

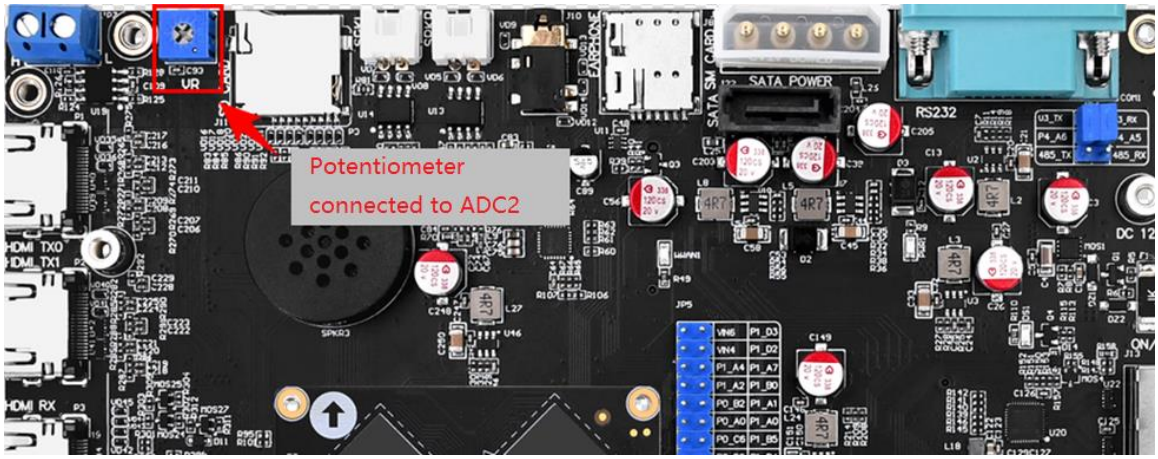
```
root@ATK-DLRK3588:/# echo 3 > /proc/sys/vm/drop_caches
[  151.183350] sh (1158): drop_caches: 3
root@ATK-DLRK3588:/# time dd if=/userdata/test of=/dev/null bs=1024k
50+0 records in
50+0 records out
52428800 bytes (52 MB, 50 MiB) copied, 0.194363 s, 270 MB/s

real    0m0.208s
user    0m0.000s
sys     0m0.076s
root@ATK-DLRK3588:/#
```

It can be seen that the reading speed is 270MB/s. The file being read is too small and this speed is for reference only.

3.15 ADC Test

Note: The absolute value of the ADC's sampling voltage is the maximum of 1.8V. Please do not exceed 1.8V, otherwise it may damage the chip.

There is a potentiometer marked with VR on the ATK-DLRK3588 baseboard. It is connected to the 2nd channel of the ADC and is convenient for users to test. As shown in the following figure:



In the Linux system, ADC sampling falls under the IIO subsystem. It can be accessed through the interface provided by SYSFS and uses a 12-bit precision. To view the raw data collected by the ADC, use the following command:

cat /sys/bus/iio/devices/iio\:device0/in_voltage2_raw

```
root@ATK-DLRK3588:/# cat /sys/bus/iio/devices/iio\:device0/in_voltage2_raw
719
root@ATK-DLRK3588:/#
```

Use the standard voltage to convert the value obtained from the AD conversion into the voltage value required by the user. The calculation formula is as follows:

Vref / (2^n-1) = Vresult / raw

Note: Vref represents the standard voltage, n denotes the number of bits in the AD conversion, raw represents the original data collected by the AD, and Vresult is the voltage that the user needs to collect.

Vresult = (1800mv * 719) /4095 = 316mV

Using the formula, we can calculate that it is approximately 316 mV.

In fact, you can also directly check the accuracy (the scaling factor of the voltage input value). The number we obtain can be multiplied by this scaling factor to potentially obtain the corresponding value.

cat /sys/bus/iio/devices/iio\:device0/in_voltage_scale

```
root@ATK-DLRK3588:/# cat /sys/bus/iio/devices/iio\:device0/in_voltage_scale
0.439453125
root@ATK-DLRK3588:/#
```

Vresult = 719 * 0.439453125 = 315mV

3.16 Main chip temperature

To view the temperature of the main chip, use the CAT command. The user layer provides an interface, and the displayed value unit is millikelvin, as shown in the following figure.

cat /sys/class/thermal/thermal_zone0/temp # Check the temperature of the main chip

```
root@ATK-DLRK3588:/# cat /sys/class/thermal/thermal_zone0/temp
47153
root@ATK-DLRK3588:/#
```

47153 represents 47.153 degrees.

3.17 CPU frequency

The command is as follows:

View the current running CPU frequency.

cat /sys/devices/system/cpu/cpu*/cpufreq/cpuinfo_cur_freq

Check the last set CPU frequency on the software.

cat /sys/devices/system/cpu/cpu*/cpufreq/scaling_cur_freq

Check the supported CPU frequencies

cat /sys/devices/system/cpu/cpufreq/policy0/scaling_available_frequencies

Enable the CPU performance mode, run at the highest frequency, for this adjustment, the small core. Note that for testing 8K video playback, it is recommended to enable the performance mode! Also, it is recommended to enable all cores; otherwise, there might be lag. I have tested this and found that enabling it makes the performance much smoother!

```
echo performance > /sys/devices/system/cpu/cpufreq/policy0/scaling_governor
echo performance > /sys/devices/system/cpu/cpufreq/policy4/scaling_governor
# Two major cores
echo performance > /sys/devices/system/cpu/cpufreq/policy6/scaling_governor
# Two major cores
```

Switch the governor to userspace

```
echo userspace > /sys/devices/system/cpu/cpufreq/policy0/scaling_governor
echo userspace > /sys/devices/system/cpu/cpufreq/policy4/scaling_governor
# Two major cores
echo userspace > /sys/devices/system/cpu/cpufreq/policy6/scaling_governor
# Two major cores
```

To set the CPU frequency, you need to first switch to the userspace. Please note that this time, it is only for demonstrating the adjustment of the main frequency of the secondary core.

```
echo 1800000 > /sys/devices/system/cpu/cpufreq/policy0/scaling_setspeed
echo 2304000> /sys/devices/system/cpu/cpufreq/policy4/scaling_setspeed
# Two major cores
echo 2304000> /sys/devices/system/cpu/cpufreq/policy6/scaling_setspeed
# Two major cores
```

Check the current CPU frequency that is running now

```
cat /sys/devices/system/cpu/cpu*/cpufreq/cpuinfo_cur_freq
```

```
root@ATK-DLRK3588:/# cat /sys/devices/system/cpu/cpu*/cpufreq/cpuinfo_cur_freq
1800000
1800000
1800000
1800000
2304000
2304000
408000
408000
root@ATK-DLRK3588:/# cat /sys/devices/system/cpu/cpu*/cpufreq/scaling_cur_freq
408000
408000
408000
408000
408000
408000
408000
408000
root@ATK-DLRK3588:/# cat /sys/devices/system/cpu/cpufreq/policy0/scaling_available_frequencies
408000 600000 816000 1008000 1200000 1416000 1608000 1800000
root@ATK-DLRK3588:/# echo performance > /sys/devices/system/cpu/cpufreq/policy0/scaling_governor
root@ATK-DLRK3588:/# echo userspace > /sys/devices/system/cpu/cpufreq/policy0/scaling_governor
root@ATK-DLRK3588:/# echo 1800000 > /sys/devices/system/cpu/cpufreq/policy0/scaling_setspeed
root@ATK-DLRK3588:/# cat /sys/devices/system/cpu/cpu*/cpufreq/cpuinfo_cur_freq
1800000
1800000
1800000
1800000
408000
408000
408000
408000
root@ATK-DLRK3588:/#
```

3.18 Headphone Insertion/Extraction Test

Use a three-section headphone plug and align it with the headphone hole on the development board for testing. The insertion and extraction of the headphone is essentially an "event", and we can use the evtest program to conduct the test.

Execute the evtest command and view the corresponding events as shown in the following figure. When inserting or extracting the headphone, it will print the corresponding information.

```
evtest
```

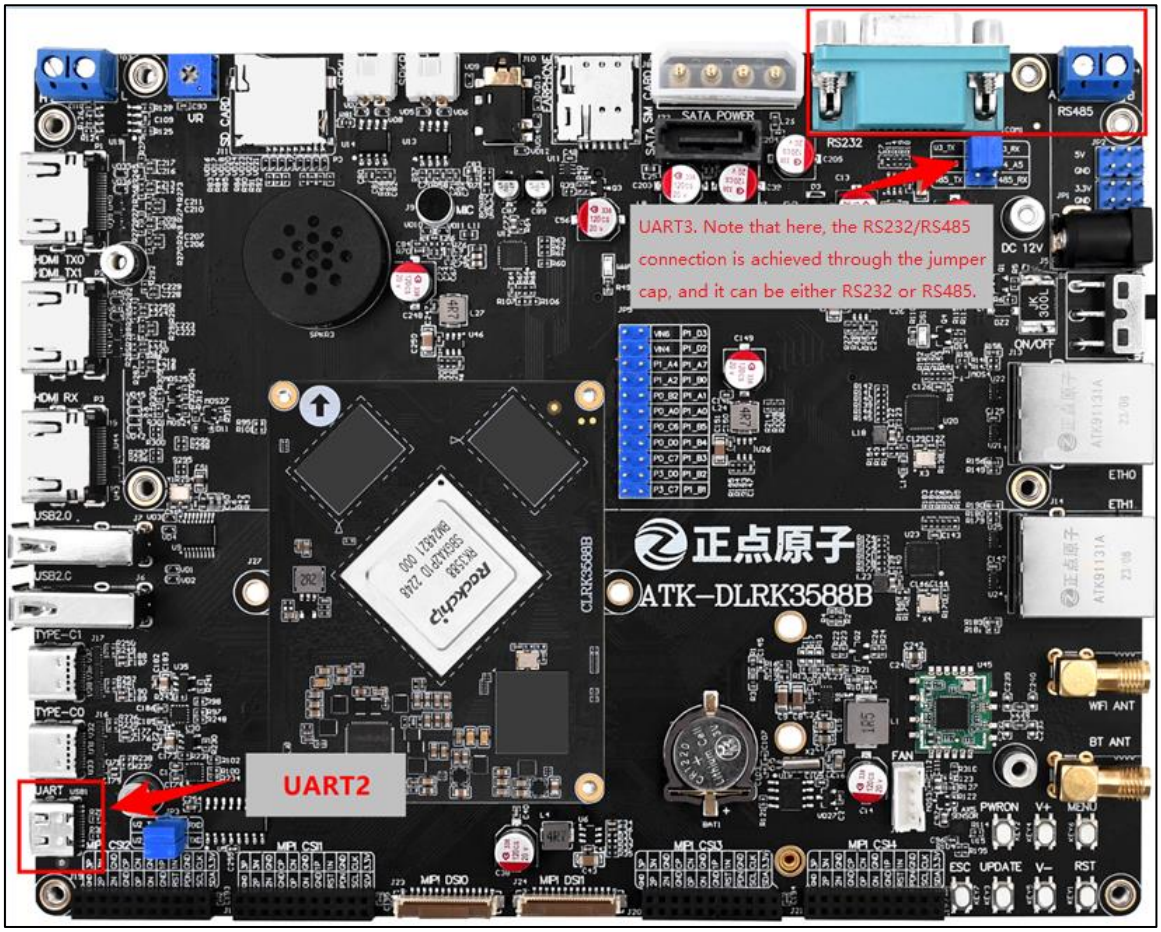
```
root@ATK-DLRK3588:/# evtest
No device specified, trying to scan all of /dev/input/event*
Available devices:
/dev/input/event0:      rk805 pwrkey
/dev/input/event1:      gsensor
/dev/input/event2:      dsi0_ts_gt9xx
/dev/input/event3:      rockchip-dp0 rockchip-dp0
/dev/input/event4:      rockchip-dp1 rockchip-dp1
/dev/input/event5:      rockchip,hdmiin rockchip,hdmiin
/dev/input/event6:      headset-keys
/dev/input/event7:      rockchip-es8388 Headset
/dev/input/event8:      gyro
/dev/input/event9:      adc-keys
/dev/input/event10:     rockchip-hdmi0 rockchip-hdmi0
/dev/input/event11:     rockchip-hdmi1 rockchip-hdmi1
Select the device event number [0-11]: 7
Input driver version is 1.0.1
Input device ID: bus 0x0 vendor 0x0 product 0x0 version 0x0
Input device name: "rockchip-es8388 Headset"
Supported events:
  Event type 0 (EV_SYN)
  Event type 5 (EV_SW)
    Event code 2 (SW_HEADPHONE_INSERT) state 1
    Event code 4 (SW_MICROPHONE_INSERT) state 0
Properties:
Testing ... (interrupt to exit)
Event: time 1714300315.244889, type 5 (EV_SW), code 2 (SW_HEADPHONE_INSERT), value 0
Event: time 1714300315.244889, -------------- SYN_REPORT ------------
Event: time 1714300316.575640, type 5 (EV_SW), code 2 (SW_HEADPHONE_INSERT), value 1
Event: time 1714300316.575640, type 5 (EV_SW), code 4 (SW_MICROPHONE_INSERT), value 1
Event: time 1714300316.575640, -------------- SYN_REPORT ------------
```

The insertion and removal of the earphones will print the following information

3.19 Serial port test

Two serial port interfaces are provided on the bottom board of the ALIENTEK ATK-DLRK3588 development board. The maximum number of serial ports supported by the RK3588 is 10.

The serial ports onboard the development board are as shown in the following figure (excluding those that are reused (the exposed IO pin sockets)).



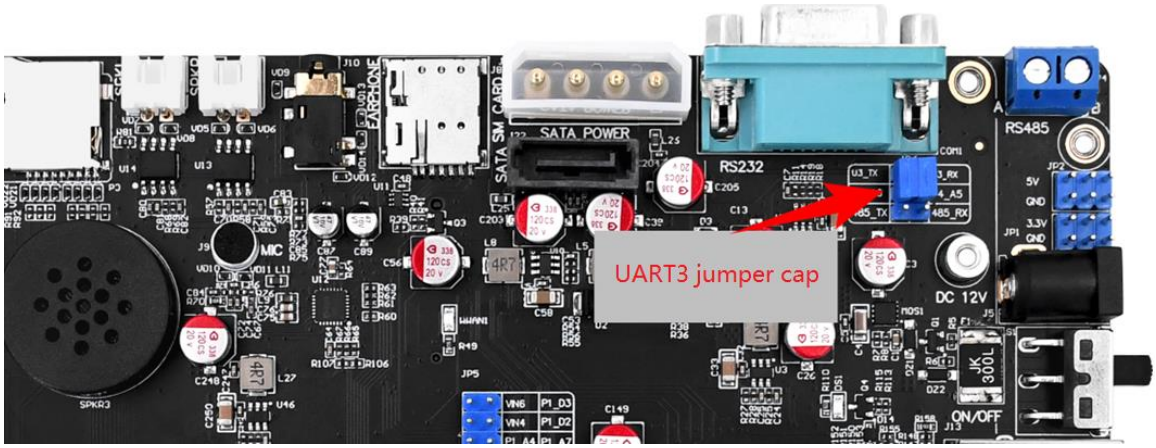
3.19.1 UART2 Test

Here, UART2 is the debugging serial port we are using. Connect it with a USB-TypeC cable to the host computer. All default information will be printed here. The default baud rate is 1500000 (1.5M). The ability to input and print information indicates that the serial port is normal.

3.19.2 UART3 (RS232) Test

The UART3 on the development board is connected to the RS232 DB9 interface (female head). If you need to test this serial port, you will need a USB-to-RS232 serial cable (the USB-to-RS232 serial cable is not an accessory for the development board, please purchase it yourself). And it needs to be male head (ALIENTEK store sells it). As shown in the following picture.

Before the test, please ensure that the jumper caps on the development board are properly connected to prevent loosening or detachment during the shipping process. Also, check if they are connected according to the wiring method as shown in the following diagram. Please carefully compare and then conduct the test! At the positions shown in the diagram.



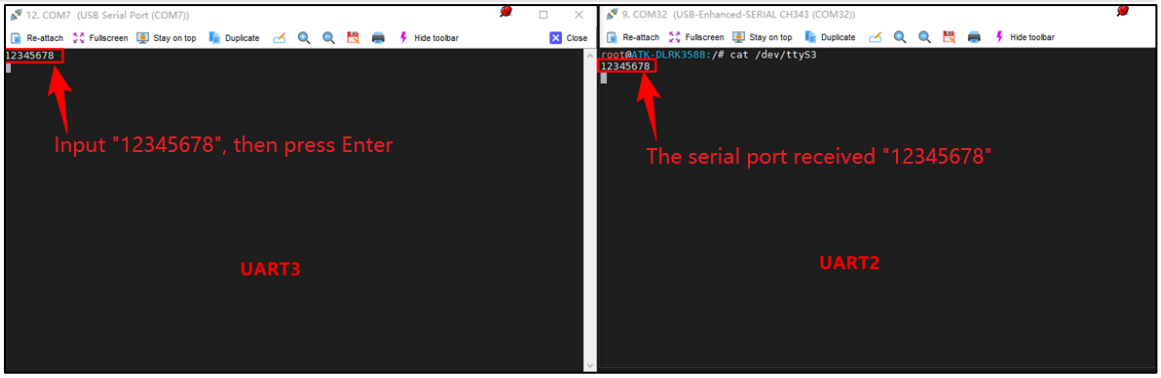
Set the serial port baud rate, connect the USB to RS232 serial cable to the development board, then connect it to the USB interface of the computer, and open the upper-level machine (Mobaxterm).

```
stty -F /dev/ttyS3 115200
```

```
root@ATK-DLRK3588:/# stty -F /dev/ttyS3 115200
[ 1158.865363] of_dma_request_slave_channel: dma-names property of node '/serial@feb60000' missing or empty
[ 1158.865417] dw-apb-uart feb60000.serial: failed to request DMA, use interrupt mode
root@ATK-DLRK3588:/#
```

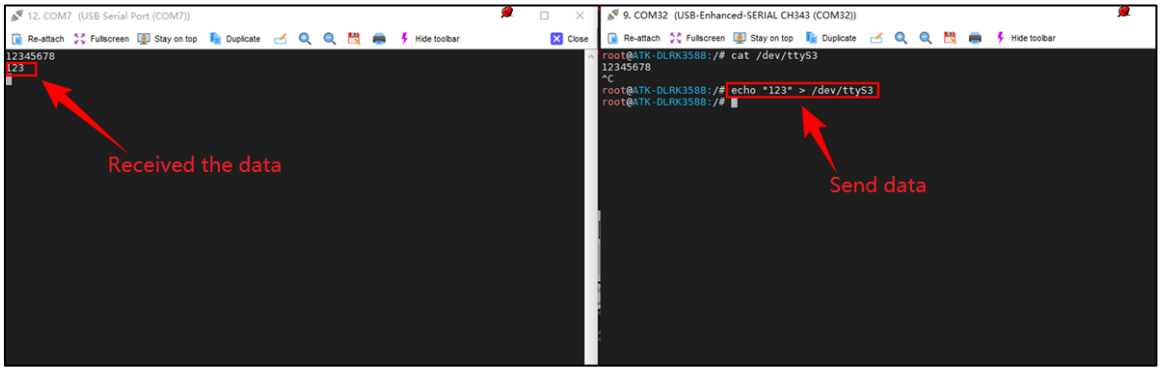
After configuring the same baud rate of 115200 in the upper computer Mobaxterm and selecting the corresponding COM port, as shown in the following figure. Here, UART3 is the serial port we need to test, and UART2 is the debugging serial port. The debugging serial port UART2 requires the input of the following instructions and then waits for UART3 to send.

```
cat /dev/ttyS3
```



Meanwhile, when we are debugging the serial port and sending data to UART3, then execute the following instructions.

```
echo "123" > /dev/ttyS3
```



If the data is received, it indicates that both the receiving and transmitting functions of UART3 are functioning properly.

3.19.3 UART3 (RS485) Test

Similarly, the test steps for UART4 are the same as those in the previous subsection 3.19.2. At this time, we need an RS485 to serial port module. This module is available for sale at the ALIENTEK store. Of course, users can also test it themselves if they have other tools for testing 485. Note that 485 is half-duplex and cannot transmit and receive simultaneously.

ALIENTEK store also sells a USB serial port converter three-in-one (supporting 485 testing), as shown in the following picture. Connect the DuPont wires, connect A to A, B to B to the RS485 interface of the development board. Follow the test steps in the previous subsection 3.19.2 to test whether the RS485 interface is working properly.

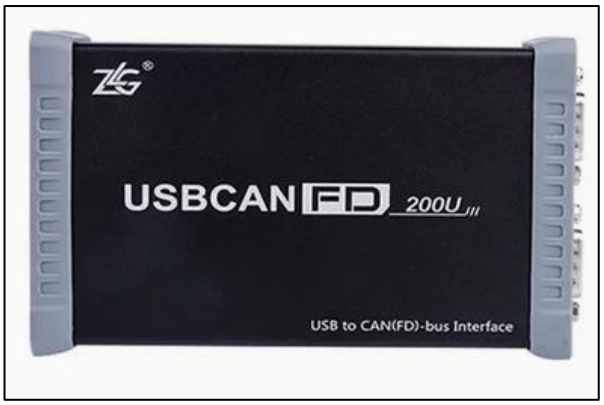
3.20 CAN Test

CAN (Controller Area Network, Controller Area Network) is a serial communication protocol used for real-time control systems and network communication. It was originally developed by the German company Bosch in the 1980s. It has been widely used in the automotive electronics field, connecting and communicating various electronic control units (ECU) in vehicles, such as engine control units, sensors, actuators, etc.

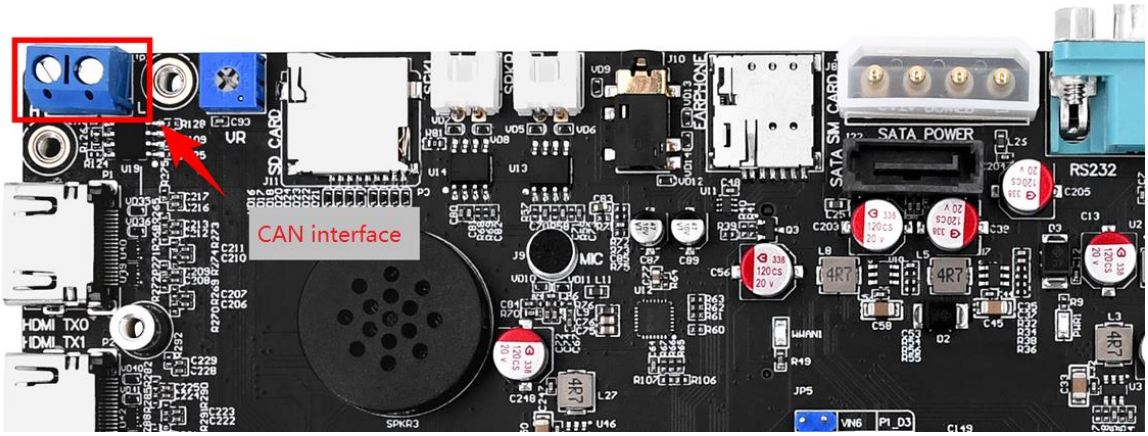
To test CAN, users need to have a USB CAN (FD) analyzer at hand, such as Zhou Ligong's CAN (FD) analyzer, etc. If the user is familiar with CAN technology, then this user should have a CAN (FD) analyzer. If you are a beginner, you need to purchase a CAN (FD) analyzer to test CAN. To test CAN, you need to purchase a CAN (FD) analyzer. Or you can also test by connecting the CAN interfaces of two development boards. If you have the conditions or requirements, you can purchase a CAN (FD) analyzer. However, USB CAN (FD) analyzers are often expensive, more expensive than a complete board set.

Zhou Ligong's USBCANFD 200U can test CAN or CANFD. Note that CAN is half-duplex and cannot transmit and receive simultaneously.

In this experiment, Zhou Ligong's USB CANFD analyzer was used, as shown in the figure below. Please consult the vendor for the use of the CANFD analyzer. This experiment does not explain the use of the USB CANFD analyzer.



The location of the CAN interface on the development board is as follows.



Connect one channel of the USB CANFD analyzer to the CAN interface of the development board at the H position, and connect the CANL at the development board's CAN interface at the L position. Start the CAN host computer.

Enter the following instructions to configure CAN.

```
ip link set can0 up type can bitrate 1000000
```

Instruction Explanation:

1. ip link set can0 up: Set the CAN0 interface to enabled state.
2. type can: Set the CAN bus type to CAN.
3. bitrate 1000000: Set the bit rate of the CAN bus to 1 Mbps (1000000 bits per second).

Note: The CAN bus typically supports bit rates ranging from 1 kbps (kilobits per second) to 1 Mbps (megabits per second), with common values being 800k, 500k, 250k, 125k, 100k and 50k.

```
root@ATK-DLRK3588:/# ip link set can0 up type can bitrate 1000000
[ 3951.727388] rockchip_canfd fea60000.can can0: bitrate error 1.0%
[ 3951.727801] IPv6: ADDRCONF(NETDEV_CHANGE): can0: link becomes ready
root@ATK-DLRK3588:/#
```

Execute the following instruction to print the information of CAN.

```
ip -details link show can0
```

```
root@ATK-DLRK3588:/# ip -details link show can0
2: can0: <NOARP,UP,LOWER_UP,ECHO> mtu 16 qdisc pfifo_fast state UP mode DEFAULT group default qlen 10
    link/can  promiscuity 0 minmtu 0 maxmtu 0
    can state ERROR-ACTIVE (berr-counter tx 0 rx 0) restart-ms 1
          bitrate 990000 sample-point 0.740
          tq 20 prop-seg 18 phase-seg1 18 phase-seg2 13 sjw 1
          rockchip_canfd: tseg1 1..128 tseg2 1..128 sjw 1..128 brp 1..256 brp-inc 2
          clock 99000000 numtxqueues 1 numrxqueues 1 gso_max_size 65536 gso_max_segs 65535
root@ATK-DLRK3588:/#
```

CAN transmission:

Send (Standard Frame, Data Frame, ID: 123, data: DEADBEEF):

```
cansend can0 123#DEADBEEF
```

Send (Standard Frame, Remote Frame, ID: 123):

```
cansend can0 123#R
```

Send (Standard Frame, Remote Frame, ID: 123): Send (Extended Frame, Data Frame, ID: 00000123, data: 12345678):

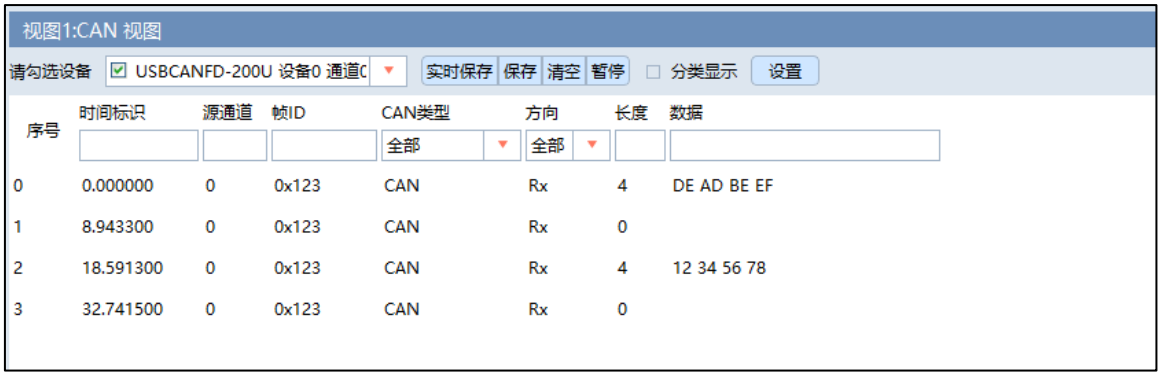
```
cansend can0 00000123#12345678
```

Send (Extended Frame, Remote Frame, ID: 00000123):

```
cansend can0 00000123#R
```

```
root@ATK-DLRK3588:/# cansend can0 123#DEADBEEF
root@ATK-DLRK3588:/# cansend can0 123#R
root@ATK-DLRK3588:/# cansend can0 00000123#12345678
root@ATK-DLRK3588:/# cansend can0 00000123#R
root@ATK-DLRK3588:/#
```

The upper computer on CAN received the following data.

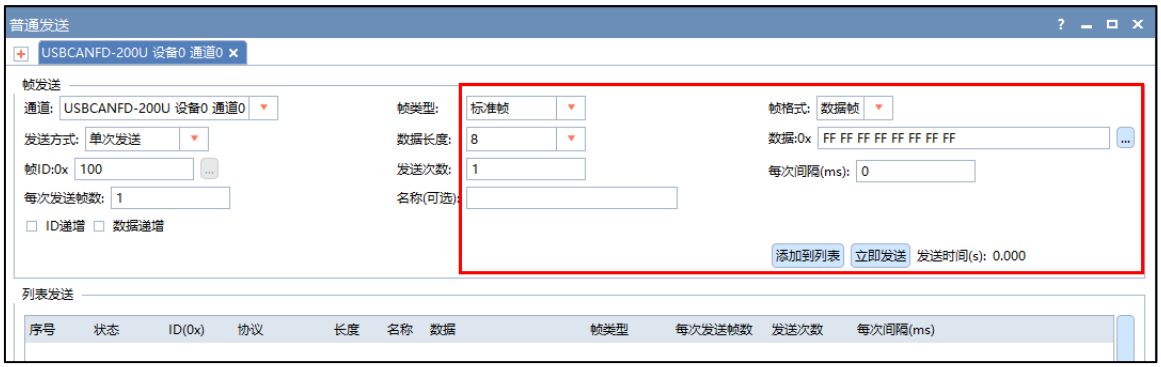


序号	时间标识	源通道	帧ID	CAN类型	方向	长度	数据
				全部	全部		
0	0.000000	0	0x123	CAN	Rx	4	DE AD BE EF
1	8.943300	0	0x123	CAN	Rx	0	
2	18.591300	0	0x123	CAN	Rx	4	12 34 56 78
3	32.741500	0	0x123	CAN	Rx	0	

CAN receives data. Input the following instruction to wait for reception:

```
candump can0
```

The data sent by the CAN host computer is as shown in the figure below. This time, we sent 8 bytes.



The development board received 8 bytes as shown in the figure below.

```
root@ATK-DLRK3588:/# candump can0
  can0  100   [8]  FF FF FF FF FF FF 00 00
```

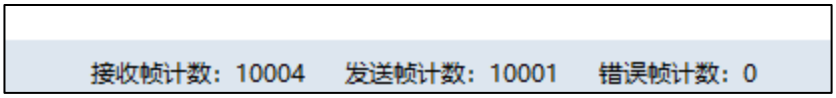
CAN randomly generates data and sends it, with 8 bytes each time. The script is as follows.

```
#!/bin/bash

# Send repeatedly for 10,000 times
for ((i = 1; i <= 10000; i++)); do
    # Generate random 8-byte data (16 hexadecimal digits)
    random_data=$(od -A n -t x8 -N 8 /dev/urandom | tr -d ' ')

    # Build the cansend command
    cansend_cmd="cansend can0 123#${random_data}"

    # Execute the "cansend" command
```

```
        eval $cansend_cmd

        echo "Sent CAN frame $i with data: $random_data"

        # ait for a period of time to control the sending rate (unit: seconds)
        sleep 0.01
done
```

The upper-level script sent data at a 10000-frame interval of 100ms for testing. After the test was completed, the upper computer also sent 10000 frames of data at an interval of 10ms to the development board via CAN.

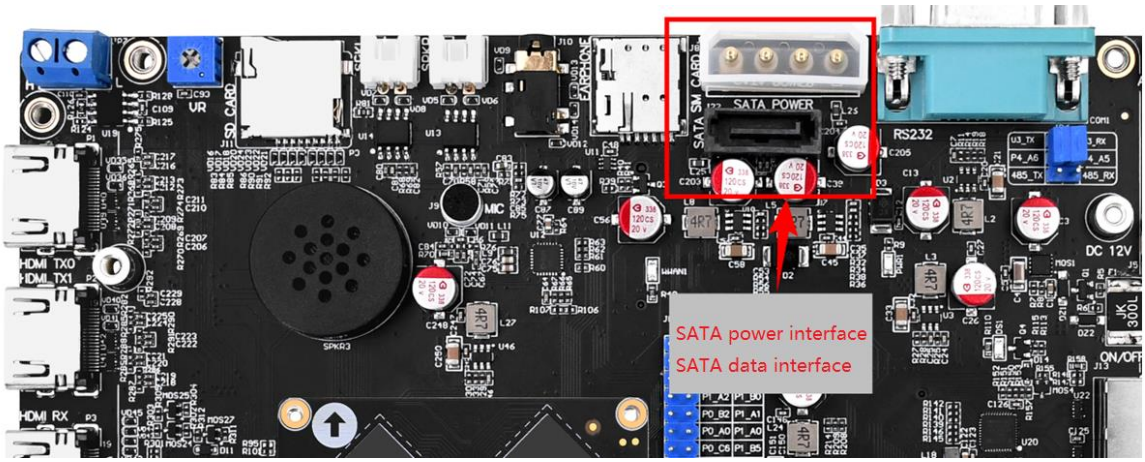
The test results of the error frames are as follows. The error rate is 0%.



3.21 SATA Interface Test

In some cases, when the eMMC storage capacity is insufficient, users need to expand the storage space. One low-cost way to increase storage capacity is to use an external hard drive.

The SATA interface of the development board is located as shown in the figure below.



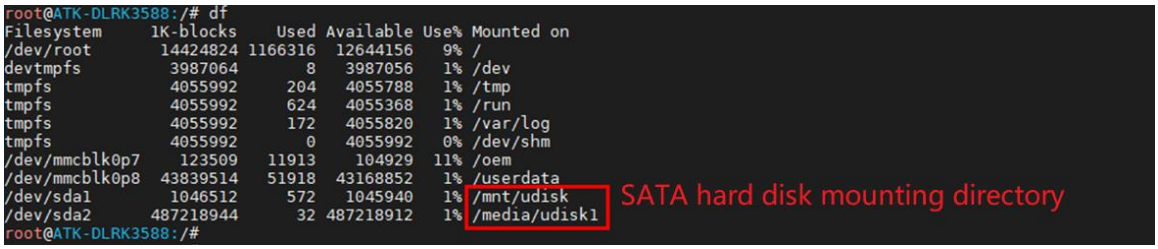
Connect the SATA interface hard drive with the SATA power interface cable and the SATA data interface cable (the SATA data cable and power cable are non-developer board accessories, please prepare them yourself). It can be hot-swapped. However, to protect your hard drive, it is recommended to turn off the power first, then plug it in, and finally turn on the computer.

After starting the computer, use the df command to check if the hard drive has been mounted.

Note: If the automatic mounting fails, please check if the disk is properly inserted! Also, make sure if the /dev/sd* node has been generated. If this node has been generated, it indicates that the hard disk has not been recognized. If a node has been generated but not mounted, then your hard disk might not have been partitioned or the partition is damaged. In this case, please format it and mount it yourself! (The formatting operation can be performed by inserting the hard disk into the PC, or using the command for formatting in Linux. Refer to the Linux Driver Basics documentation.)

```
df
```

As shown in the following figure, the directory where this SATA hard drive is mounted by the author is /media/usdisk1 and /mnt/udisk. Since this hard drive of the author has two partitions, one is the sda1 partition and the other is the sda2 partition, there are two directories for the mount.



Then we can perform read and write operations on this directory.

3.22 M.2 Interface Test

The M.2 interface uses the PCI Express (PCIe) bus. PCIe is a high-speed serial bus used to connect internal components of a computer, such as graphics cards, network cards, storage devices, etc. The M.2 interface provides high-speed data transmission through the PCIe bus and supports a transmission speed of up to 4 GB/s.

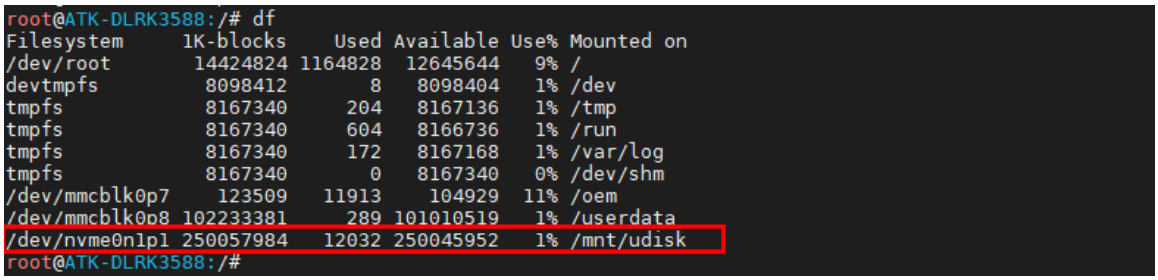
The ATK-DLRK3588 M.2 interface is on the back of the development board. Please install the solid-state drive with the M.2 interface on the back of the development board (please prepare your own M.2 screws, most users who purchase M.2 hard drives will be given or check if there are corresponding screws in the accessory package), then power on and start.

After powering on and starting up, the M.2 hard drive will automatically be mounted.

Note: If it doesn't automatically mount, please check if it is properly inserted! At the same time, make sure if the /dev/nvme0n1 node has been generated. If this node has been generated, it means the hard drive has not been recognized. If a node has been generated but not mounted, then your hard drive might not have been partitioned or the partition is damaged. Please format it yourself and then mount it! (The formatting operation can be performed by inserting the hard drive into the PC, or using the command for formatting in Linux. Refer to the Linux driver basic documentation.)

Use the df -h command to check the mounted node of the M.2 hard drive.

```
df -h
```



It can be seen that the M.2 solid-state drive has been mounted under the path /mnt/udisk*. Now we can access and write files in the path /media/udisk*.

3.23 5G Module Interface Test

3.23.1 5G Module Internet Access Test

Please prepare the Fibocom_FG132 5G module and connect it to the antenna. Refer to the following picture. It is already supported on the ALIENTEK ATK-DLRK3588 development board. Note that it must be this type of 5G module! The 5G module interface is on the back of the development board, and the SIM card interface is on the front of the development board.

After powering on and logging into the terminal, wait for about 20 seconds more, and the terminal will print the following information, indicating that the 5G module has been recognized and loaded successfully.

```
root@ATK-DLRK3588:/# timed out
[   38.236955] usb 2-1.1: new high-speed USB device number 3 using ehci-platform
[   38.357506] usb 2-1.1: New USB device found, idVendor=2cb7, idProduct=0112, bcdDevice= 5.15
[   38.357517] usb 2-1.1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[   38.357523] usb 2-1.1: Product: Fibocom Module
[   38.357527] usb 2-1.1: Manufacturer: Fibocom Wireless Inc.
[   38.357532] usb 2-1.1: SerialNumber: 9cbbabf9
[   38.382506] qmi_wwan 2-1.1:1.0: cdc-wdm0: USB WDM device
[   38.383092] qmi_wwan 2-1.1:1.0 wwan0: register 'qmi_wwan' at usb-fc880000.usb-1.1, WWAN/QMI device, 0a:1b:f9:68:a5:c7
[   38.383258] option 2-1.1:1.1: GSM modem (1-port) converter detected
[   38.383461] usb 2-1.1: GSM modem (1-port) converter now attached to ttyUSB0
[   38.383985] option 2-1.1:1.2: GSM modem (1-port) converter detected
[   38.384179] usb 2-1.1: GSM modem (1-port) converter now attached to ttyUSB1
[   38.384353] option 2-1.1:1.3: GSM modem (1-port) converter detected
[   38.384522] usb 2-1.1: GSM modem (1-port) converter now attached to ttyUSB2

root@ATK-DLRK3588:/#
```

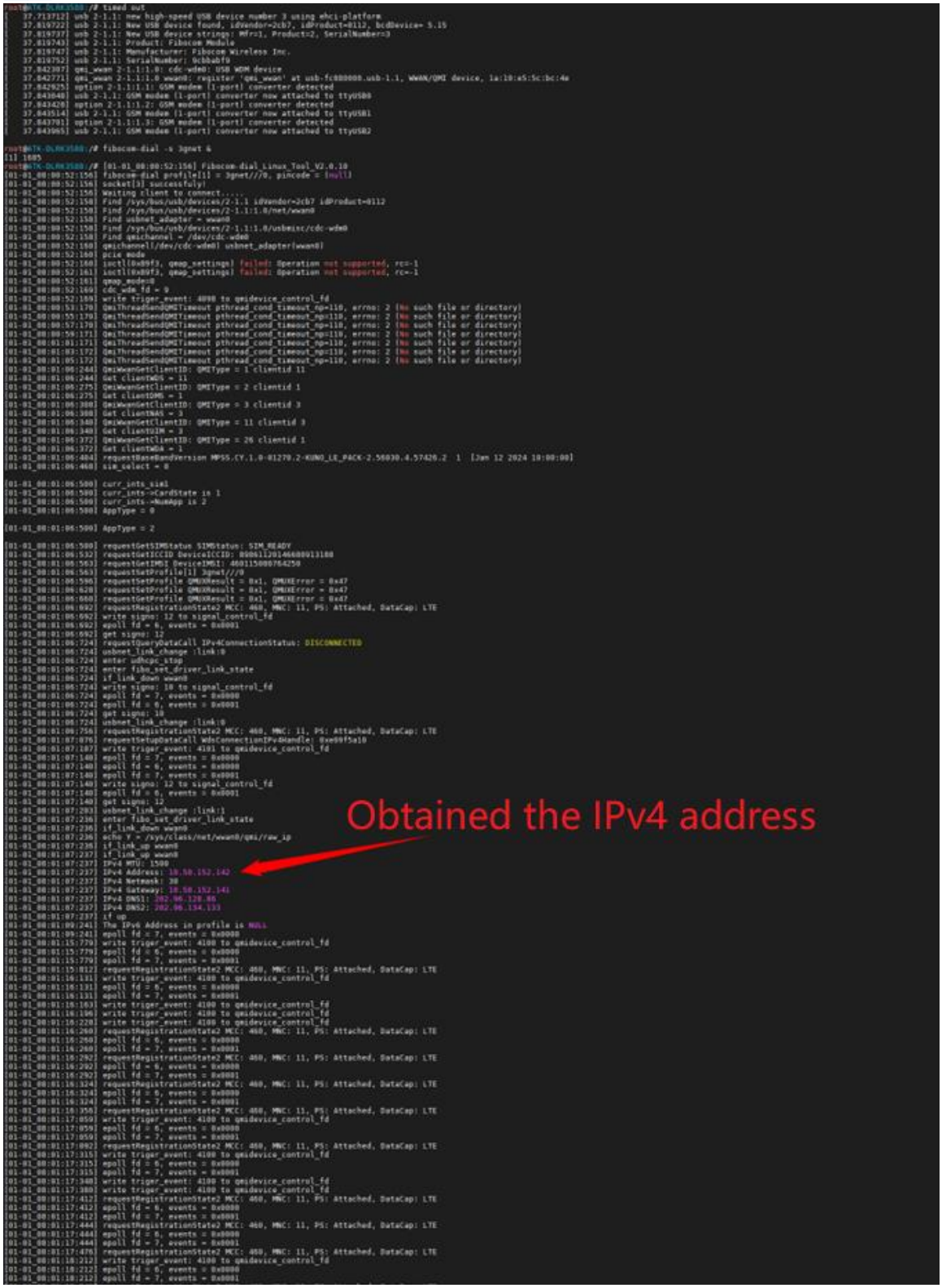
3.23.1.1 Dialing test

The ATK-DLRK3588 development board already integrates the dialing tools from Guanghe Tong and Ruiwei. Simply execute the following command to make the call.

```
fibocom-dial -s 3gnet &
// Guanghetong uses this command for dialing. Note that this command will print a large number of logs generated by fibocom-dial, causing the command line to constantly refresh.
quectel-CM &                    // Miotong used this command for dialing.
```

Note: You can redirect the logs to /dev/null using `fibocom-dial -s 3gnet >/dev/null 2&>1 &` or `quectel-CM > /dev/null 2&>1 &`. This means to suppress the logs.

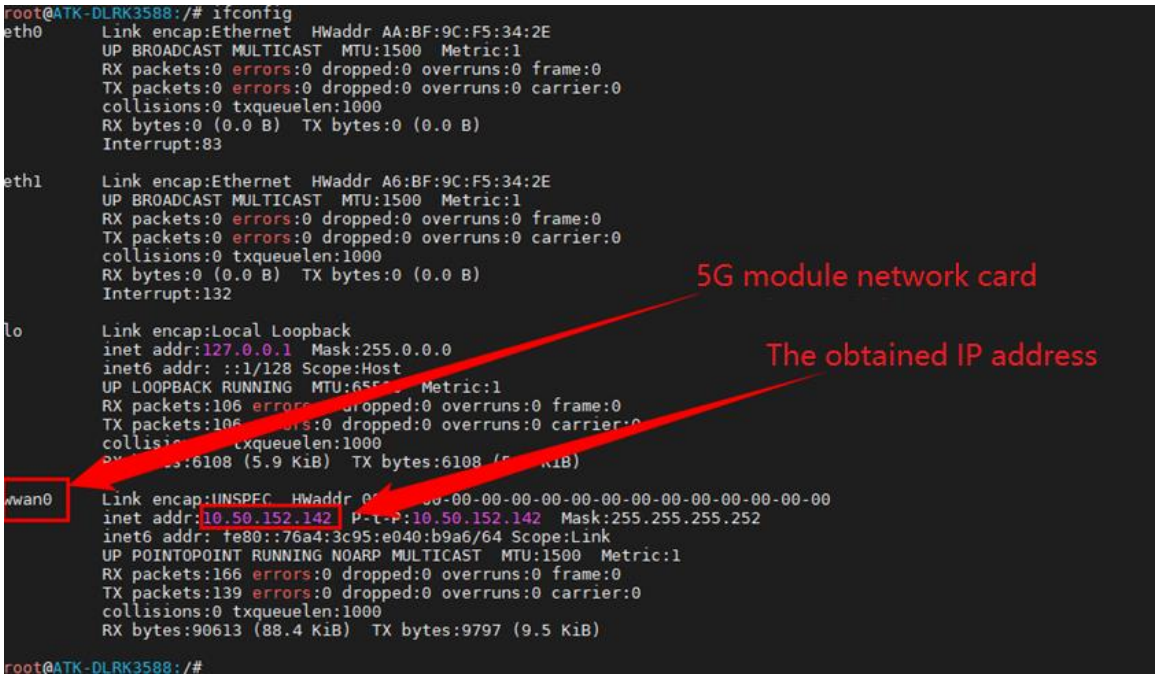
After executing the command, as shown in the following figure, the dialing program fibocom-dial will print the log (similar to the errno log). About 8 seconds later, it will start to obtain the IP. This phenomenon is normal. Once the IP is obtained, it indicates that the dialing is successful. In my test of using this module, it seems quite simple. Just executing a dialing program enables you to access the internet. During the dialing process, fibocom-dial automatically opens the wwan0 network card (which is the 5G module network card).



3.23.1.2 Internet testing

Enter the "ifconfig" command to view the obtained IP address.

```
ifconfig
```



To check if the device can access the internet, it is definitely done by pinging the external network. Execute the command "ping baidu". - The "I" option refers to the specific network card device.

```
ping www.baidu.com -I wwan0
```

```
root@ATK-DLRK3588:/# ping www.baidu.com -I wwan0
PING www.a.shifen.com (183.2.172.185) from 10.50.152.142 wwan0: 56(84) bytes of data.
64 bytes from 183.2.172.185 (183.2.172.185): icmp_seq=1 ttl=52 time=25.0 ms
64 bytes from 183.2.172.185 (183.2.172.185): icmp_seq=2 ttl=52 time=26.9 ms
64 bytes from 183.2.172.185 (183.2.172.185): icmp_seq=3 ttl=52 time=26.9 ms
[03-08_10:32:26:903] write triger_event: 4100 to qmidevice_control_fd
[03-08_10:32:26:904] epoll fd = 6, events = 0x0000
[03-08_10:32:26:904] epoll fd = 7, events = 0x0001
[03-08_10:32:26:936] requestRegistrationState2 MCC: 460, MNC: 11, PS: Attached, DataCap: LTE
[03-08_10:32:27:096] write triger_event: 4100 to qmidevice_control_fd
[03-08_10:32:27:096] epoll fd = 6, events = 0x0000
[03-08_10:32:27:096] epoll fd = 7, events = 0x0001
[03-08_10:32:27:127] write triger_event: 4100 to qmidevice_control_fd
[03-08_10:32:27:159] write triger_event: 4100 to qmidevice_control_fd
[03-08_10:32:27:192] requestRegistrationState2 MCC: 460, MNC: 11, PS: Attached, DataCap: LTE
[03-08_10:32:27:192] epoll fd = 6, events = 0x0000
[03-08_10:32:27:192] epoll fd = 7, events = 0x0001
[03-08_10:32:27:224] requestRegistrationState2 MCC: 460, MNC: 11, PS: Attached, DataCap: LTE
[03-08_10:32:27:224] epoll fd = 6, events = 0x0000
[03-08_10:32:27:224] epoll fd = 7, events = 0x0001
64 bytes from 183.2.172.185 (183.2.172.185): icmp_seq=4 ttl=52 time=338 ms
[03-08_10:32:27:256] requestRegistrationState2 MCC: 460, MNC: 11, PS: Attached, DataCap: LTE
64 bytes from 183.2.172.185 (183.2.172.185): icmp_seq=5 ttl=52 time=30.3 ms
64 bytes from 183.2.172.185 (183.2.172.185): icmp_seq=6 ttl=52 time=25.0 ms
^C
--- www.a.shifen.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5005ms
rtt min/avg/max/mdev = 25.026/78.731/338.209/116.055 ms
root@ATK-DLRK3588:/#
```

When data reply is received, it indicates that the 5G module can access the internet.

3.24 View NPU information

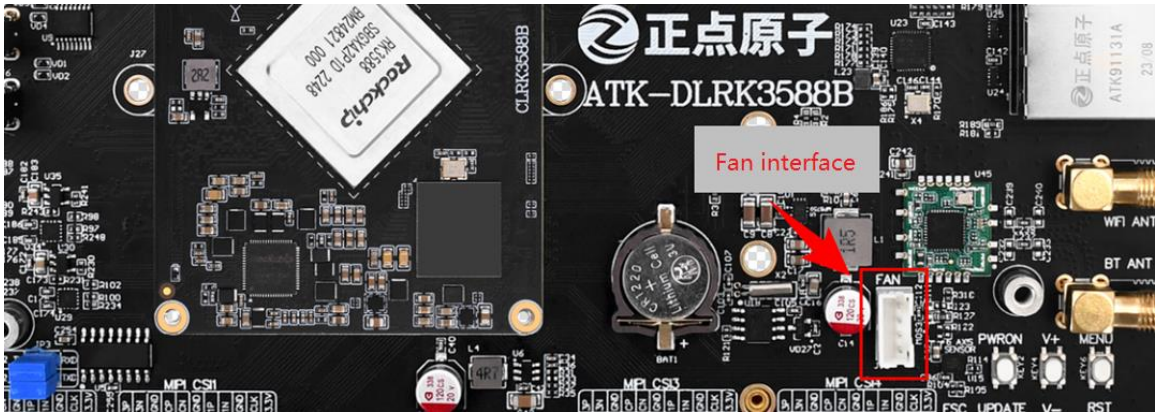
Check the NPU usage rate. Note: The usage rate will change only when the routine that calls the NPU is executed.

```
cat /sys/kernel/debug/rknpu/load
```

```
root@ATK-DLRK3588:/# cat /sys/kernel/debug/rknpu/load
NPU load:  Core0:  0%, Core1:  0%, Core2:  0%,
root@ATK-DLRK3588:/#
```

3.25 CPU cooling fan test

Please install the fan on the motherboard. Just align the two self-locking seats of the fan with the two reserved holes on the core and install them. To remove it, you need to remove the motherboard. Unbolt the self-locking seats from the back of the motherboard and connect the fan connector to the following position on the development board. For detailed instructions, please refer to the fan installation guide on the Taobao page.



When the ATK-DLRK3588 development board is powered on normally, the CPU cooling fan will rotate for a while. After a while, when the temperature control system is properly initialized, the fan will stop. Later, the fan will start to cool the CPU when the temperature exceeds 32 degrees. The purpose of this is to reduce the power consumption of the entire development board. The device tree settings can be modified to adjust the trigger threshold for the fan.

During a cold winter, if the development board has few external peripherals connected, it may be difficult to reach 32 degrees. In such cases, the fan will not start. To make the fan start, the CPU load state needs to be increased.

3.26 PCIE WIFI & BT interface test

Since we need to connect specific types of PCIE modules, we have another document written by us - the ATK-DLRK3588 PCIE WIFI module user manual. The specific location is: Development Board CD-ROM A Disk - Basic Materials -> 10_user_manual -> 03, Supplementary Documents "28 [ALIENTEK] ATK-DLRK3588_PCIe_WiFi Module User Manual V1.0.pdf".

If the image file used is from the B drive of the development board disc - Development Environment and SDK -> 02, ATK-DLRK3588 development board SDK -> 06, linux_r8_sdk -> atk-rk3588_linux_release_R8_v1.0_20250104.tgz, the functions listed below may not work properly:

The Bluetooth driver of the RTL8723BE module cannot function normally

The Bluetooth driver of the RTL8822BE module cannot function normally

Chapter 4. Common Function Settings of the Buildroot System

Here are the functions that regular ALIENTEK users often need in the system.

4.1 Mount point information

After the system starts up, the mount points are as follows.

```
root@ATK-DLRK3588:/# df
Filesystem     1K-blocks    Used Available Use% Mounted on
/dev/root       14422864 1071532  12737316   8% /
devtmpfs         4044396       8   4044388   1% /dev
tmpfs            4055980     220   4055760   1% /tmp
tmpfs            4055980     756   4055224   1% /run
tmpfs            4055980     228   4055752   1% /var/log
tmpfs            4055980       0   4055980   0% /dev/shm
/dev/mmcblk0p7    123509   11913    104929  11% /oem
/dev/mmcblk0p8  43839514    2244  43218526   1% /userdata
root@ATK-DLRK3588:/#
```

The important mount points are as follows:

/: Root directory, which is the common root directory we usually encounter, containing common folders/files such as /dev/ and /usr/lib.

/oem: Manufacturer folder, where the test files/programs provided by the manufacturer are stored.

/userdata: User data folder. It is recommended that users save their data to this directory because this folder has more space than the root directory. In the future when users create products, it is also recommended to save the data generated by the App to this folder to prevent the root partition from being filled up and affecting startup.

4.2 Common Software Versions of the System

Software version	Supported functions
nginx 1.20.1	High-performance web applications and services
ffmpeg 4.4.1	Multimedia processing tools
Qt5 5.15.8	Support for qml, QWidget, QChart, etc. Support for sqlite, mysql drivers, etc.
gstreamer 1.22.2	Multimedia framework
opencv 4.5.4	Computer vision library
python 3.10.5	Advanced programming language, facilitating development
weston 12.0.1	Based on the Wayland framework, window manager and synthesizer
……	