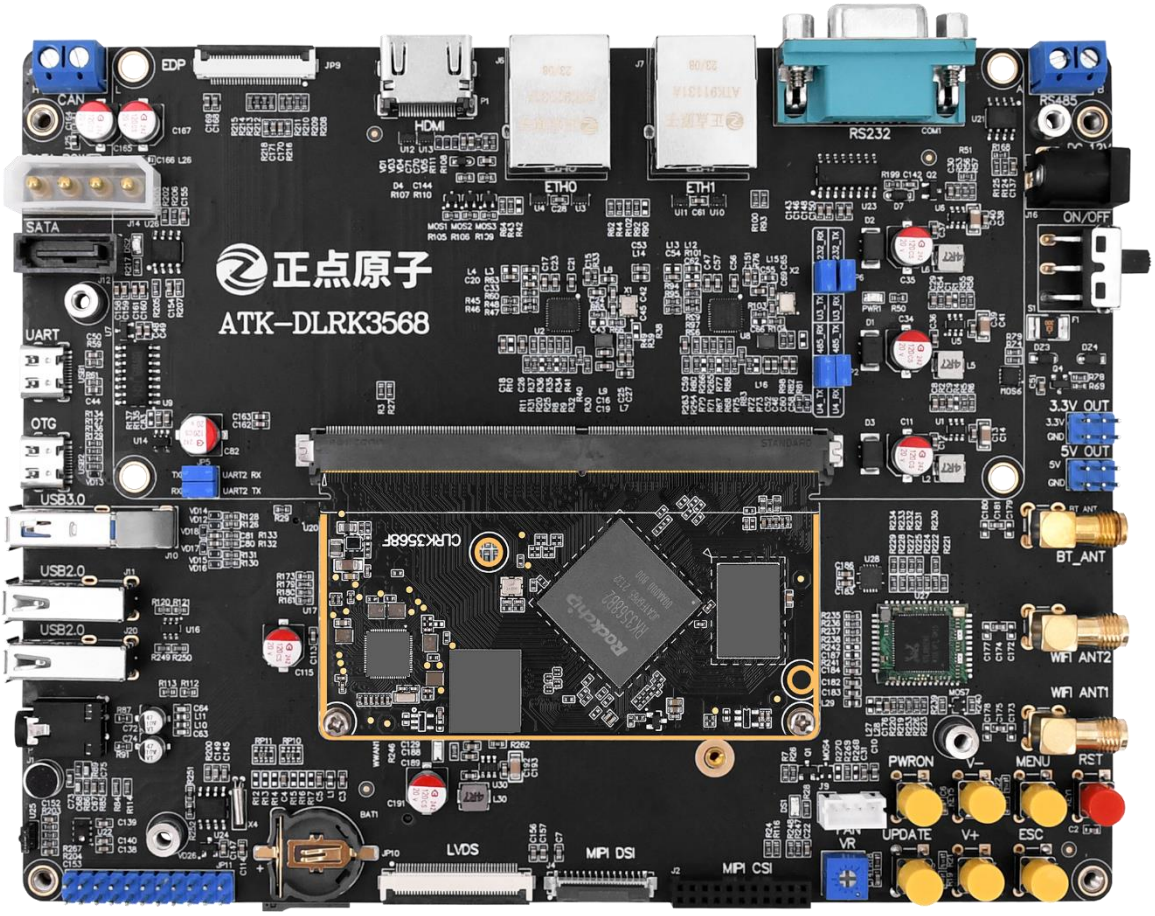
ATK-DLRK3568

Buildroot System Quick test manual

V1.0



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. The copyright of this document belongs to Guangzhou Xingyi Electronic Technology Co., LTD. Without the written permission of the company, any unit or individual shall not be used for profit-making purposes in any way of dissemination.

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	2022.07.03

Brief

This document is a quick experience guide for the ALIENTEK ATK-DLRK3568 development board. It can be regarded as a hardware testing document. Here, we can learn about the usage methods of the development board and the testing methods for peripheral devices. We can also understand the peripheral resources on the board. Therefore, this document is quite important.

The environment used in this document:

- Windows 10 64-bit, it is 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: Please use 20.04 for Ubuntu. If you have used a new version of Ubuntu and the installation environment is different, it may cause errors. Please solve it by yourself!
- The readers are required to be able to use FileZilla, WinSCP and Windows Git to transfer files between Ubuntu and Windows.

This document is the Buildroot system test document for the ALIENTEK ATK-DLRK3568 development board.

Catalogue

Brief 5
Chapter 1. Preparation for using the ATK-DLRK3568 development board 1
1.1 Precautions before powering on 1
1.2 Serial Port Software Installation 1
1.3 Connecting the development board to the PC 1
1.4 SSH login password 2
Chapter 2. Burn the Android system image 3
Chapter 3. Function Testing of ATK-DLRK3568 Development Board 4
3.1 LED Testing 5
3.2 Key Test 5
3.3 Display and Touch Testing 6
3.3.1 Touch Test 6
3.3.2 Multi-touch Test 6
3.3.3 MIPI backlight test 8
3.3.4 LVDS and EDP screen backlight test 8
3.3.5 HDMI Test 9
3.4 RTC Clock Test 9
3.5 Audio Test 10
3.5.1 Playing Audio Test 10
3.5.2 Audio test 10
3.6 Camera Test 11
3.7 SH3001 Six-Axis Sensor Test 12
3.7.1 Acceleration Test 13
3.7.2 Angular velocity 14
3.8 USB test 15
3.8.1 USB HOST test 15
3.8.1.1 USB2.0 Test 16
3.8.1.2 USB2.0 Test 17
3.8.2 OTG Slave Device Test 17
3.8.3 OTG Host Device Test 17
3.9 Network Test 18
3.10 Onboard SDIO WIFI test 20
3.10.1 Station mode 20
3.10.2 SoftAP mode 21
3.11 Built-in Bluetooth testing 24
3.11.1 Bluetooth Initialization 25
3.11.2 Bluetooth rfcomm 28
3.11.3 Bluetooth Pairing 28
3.11.4 Bluetooth communication test 31
3.12 TF(SD) Card & eMMC Test 34
3.12.1 SD Card Test 34

3.12.2 EMMC Test....35
3.13 ADC Test....36
3.14 CPU temperature....37
3.15 CPU frequency....37
3.16 Infrared test....38
3.16.1 Testing for Multiple Infrared Remotes....39
3.16.2 Applicable to ALIENTEK Infrared Remote Control Module....39
3.17 Headphone Insertion/Extraction Test....40
3.18 Light sensor AP3216C test....40
3.19 Serial port test....42
3.19.1 UART2 Test....42
3.19.2 UART3 Test....43
3.19.3 UART4 Test....44
3.20 CANFD Test....45
3.21 SATA Interface Test....47
3.21.1 Partition the hard disk....48
3.21.2 Format the partition....50
3.22 M.2 Interface Test....51
3.23 5G Module Interface Test....52
3.23.1 5G Module Internet Access Test....52
3.23.1.1 Configuration of driver type....53
3.23.1.2 Configure dial-up mode....54
3.23.1.3 Internet Test....55
3.23.2 4G Module Internet Test....56
3.23.2.1 Dial Test....57
3.23.2.2 Internet test....58
3.24 View NPU information....58
3.25 CPU Cooling Fan Test....58
Chapter 4. Common Function Settings of Buildroot System....60
4.1 Mount point information....61
4.2 Set static IP....61
4.3 Create a startup program....63
Chapter 5. System software related information....66
5.1 Common system software versions....67

Chapter 1. Preparation for using the ATK-DLRK3568 development board

Here we will introduce the matters that need to be paid attention to when using the development board. Please read carefully. Because if you don't pay attention to these matters, it may lead to the failure of your experiment or damage to the development board, etc. So, please be sure to pay attention.

1.1 Precautions before powering on

- Firstly, check if the power supply is plugged in. If it is, then check if the power plug switch is turned on and if the power supply is normal.
- Before leaving the factory, the entire board was made of acrylic and covered with a screen protector. However, in some cases, users may remove the acrylic board or the screen. At this time, it is necessary to observe that the bottom board should not be placed on the clutter table. A protective film should be placed under the bottom board, otherwise it is easy to come into contact with metal objects, which may cause the development board to short circuit; in addition, it is also necessary to pay attention to waterproofing, moisture-proofing, and dust-proofing.

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 software. No detailed installation tutorials will be provided as it is relatively basic.

- Install the CH340 USB serial port driver (for the computer to communicate with the development board's serial port, we need to install this driver)

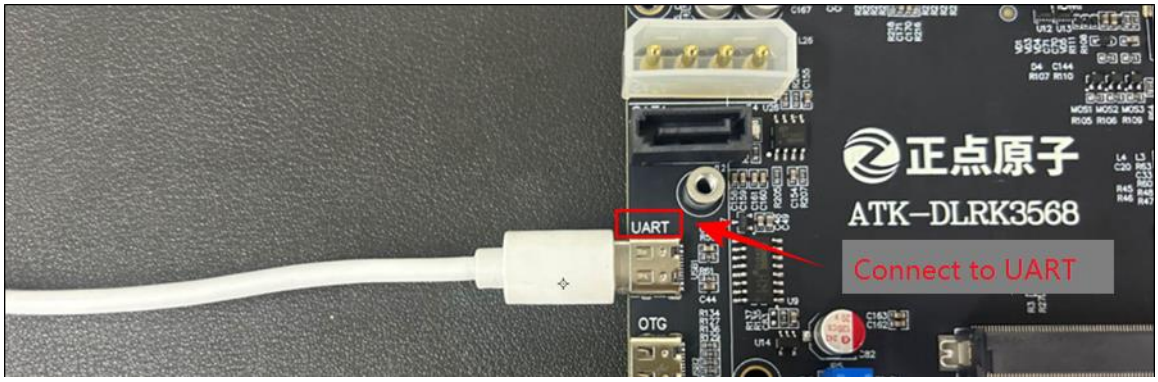
On the development board CD-ROM A - Basic Materials -> 4_softwares -> CH340SER.EXE (actually, it is also applicable to Windows 10. If you have already installed it, you don't need to install again). Double-click to run it, and then the pop-up window, directly click to install, wait for the pre-installation or the successful installation window to appear. Note: The default baud rate for debugging the development board's serial port is 1500000 (1.5M).

- Install the MobaXterm terminal software (or install Xshell, SecureCRT, etc. as terminal software) This article takes MobaXterm as an example. The installation package of MobaXterm is located at: development board CD-ROM A - Basic Materials -> 4_softwares -> MobaXterm_Installer_v12.3.zip. Double-click to open this compressed file, after decompression, directly double-click MobaXterm_installer_12.3.msi to install. The MobaXterm installation program will guide the user to install.

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-DLRK3568 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, and then connect the USB Type-C connection cable, and the other end connects to the USB interface of the computer host. The connection diagram is as follows.



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

1.4 SSH login password

The password for the SSH root user login to the ATK-DLRK3568 device 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 logs in automatically. 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 it and want to restore it to the factory state, then you can follow the instructions in the second chapter.

Chapter 2. Burn the Android system image

The default system pre-installed on the ATK-DLRK3568 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 "[ALIENTEK] ATK-DLRK3568 Factory Image Flashing Guide", from Chapter 1 to Chapter 3, to flash the Buildroot Linux system.

Chapter 3. Function Testing of ATK-DLRK3568 Development Board

This chapter will introduce how to test the ATK-DLRK3568 development board. It explains the usage method and testing method of the development board. It is very helpful for development. Note that many tests require self-provided testing tools. Please complete the testing steps according to your personal situation. Most of the hardware has already been tested when it was manufactured.

3.1 LED Testing

The ALIENTEK ATK-DLRK3568 development board is equipped with a user LED. The relationship between them and the hardware pins on the baseboard is as shown in the following table. In the factory system, we use the LED as a heartbeat indicator, and it is generally used to indicate whether the system is operating normally.

Pin name	PI0
ALIENTEK ATK-DLRK3568	GPIO0_C0

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 supported trigger modes of LED0, the default 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, it is in a constant-on state at this time, the value ranges from 1 to 255
echo 0 > /sys/class/leds/work/brightness  # Turn off LED0, it is in a off state at this time, the value is 0
```

3.2 Key Test

The ATK-DLRK3568 development board of the ALIENTEK has 7 keys equipped. Among them, 4 keys are simulated using ADC0. When the keys are pressed, ADC0 has different voltage changes.

	KEY4	KEY5	KEY6	KEY7
Name of the button	V+	V-	MENU	ESC
Value of ADC0	9	235	457	683

Run the following command in the serial terminal of the development board for testing, and then input the number "7" because the key event is event7. If you need 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-DLRK356X:~# evtest
No device specified, trying to scan all of /dev/input/event*
Available devices:
/dev/input/event0:      fe6e0030.pwm
/dev/input/event1:      rk805 pwrkey
/dev/input/event2:      goodix-ts
/dev/input/event3:      lightsensor-level
/dev/input/event4:      proximity
/dev/input/event5:      gsensor
/dev/input/event6:      gyro
/dev/input/event7:      adc-keys
/dev/input/event8:      rockchip,rk809-codec Headphones
/dev/input/event9:      rockchip,hdmi rockchip,hdmi
Select the device event number [0-9]: 7
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 1 (KEY_ESC)
    Event code 114 (KEY_VOLUMEDOWN)
    Event code 115 (KEY_VOLUMEUP)
    Event code 139 (KEY_MENU)
Properties:
Testing ... (interrupt to exit)
Event: time 1680834779.518398, type 1 (EV_KEY), code 1 (KEY_ESC), value 1
Event: time 1680834779.518398, -------------- SYN_REPORT ------------
Event: time 1680834779.724699, type 1 (EV_KEY), code 1 (KEY_ESC), value 0
Event: time 1680834779.724699, -------------- SYN_REPORT ------------
Event: time 1680834780.551599, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), value 1
Event: time 1680834780.551599, -------------- SYN_REPORT ------------
Event: time 1680834780.861345, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), value 0
Event: time 1680834780.861345, -------------- SYN_REPORT ------------
Event: time 1680834783.135029, type 1 (EV_KEY), code 114 (KEY_VOLUMEDOWN), value 1
Event: time 1680834783.135029, -------------- SYN_REPORT ------------
Event: time 1680834783.341355, type 1 (EV_KEY), code 114 (KEY_VOLUMEDOWN), value 0
```

3.3 Display and Touch Testing

In the factory system, since the screen wiring of the ALIENTEK has ADC pins connected to the CPU, the factory system can recognize and load different screen resolutions. The MIPI screens used by ALIENTEK come in two types, 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	340

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 is also acceptable for normal touch.

3.3.2 Multi-touch Test

The MIPI screen supports a maximum of 5 points of touch. Although Qt supports multi-touch, the controls we use are all single-point. Multi-touch requires the use of 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 can use tslib for testing (the default file system is not configured). Similarly, 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.

```
root@ATK-DLRK356X:~# evtest
No device specified, trying to scan all of /dev/input/event*
Available devices:
/dev/input/event0:      fe6e0030.pwm
/dev/input/event1:      rk805 pwrkey
/dev/input/event2:      goodix-ts
/dev/input/event3:      lightsensor-level
/dev/input/event4:      proximity
/dev/input/event5:      gsensor
/dev/input/event6:      gyro
/dev/input/event7:      adc-keys
/dev/input/event8:      rockchip,rk809-codec Headphones
/dev/input/event9:      rockchip,hdmi rockchip,hdmi
Select the device event number [0-9]: 2
Input driver version is 1.0.1
Input device ID: bus 0x18 vendor 0xdead product 0xbeef version 0x28bb
Input device name: "goodix-ts"
Supported events:
  Event type 0 (EV_SYN)
  Event type 1 (EV_KEY)
    Event code 330 (BTN_TOUCH)
  Event type 3 (EV_ABS)
    Event code 0 (ABS_X)
      Value      0
      Min        0
      Max      720
    Event code 1 (ABS_Y)
      Value      0
      Min        0
      Max     1280
    Event code 47 (ABS_MT_SLOT)
      Value      0
      Min        0
      Max       15
    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      720
    Event code 54 (ABS_MT_POSITION_Y)
      Value      0
      Min        0
      Max     1280
    Event code 57 (ABS_MT_TRACKING_ID)
      Value      0
      Min        0
      Max      255
Properties:
  Property type 1 (INPUT_PROP_DIRECT)
Testing ... (interrupt to exit)
```

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 levels range from 0 to 255, with the default being 255. The larger the value, the brighter the screen.

```
cat /sys/class/backlight/backlight1/max_brightness
# Check the maximum brightness level of the LCD.
cat /sys/class/backlight/backlight1/brightness          # Check the current brightness level
echo 255 > /sys/class/backlight/backlight1/brightness
# Modify the brightness level of the current observation screen and adjust the brightness changes.
cat /sys/class/backlight/backlight1/brightness          # Check the current brightness level again
```

3.3.4 LVDS and EDP screen backlight test

On the hardware side, since there are not many PWNs available for the LVDS and EDP screens to use, they share a single PWN path. The same as in the previous section. If you have an EDP or LVDS screen connected, then you can use the following commands to adjust the backlight of 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 255 > /sys/class/backlight/backlight/brightness
# Modify the brightness level of the current observation screen and adjust the brightness changes.
cat /sys/class/backlight/backlight/brightness
# Check the current brightness level again
```

3.3.5 HDMI Test

The default factory system supports HDMI. Please connect the display with an HDMI cable to test the display. The HDMI display has been rotated by 270 degrees by default, just to adapt to the factory's Qt interface. Note: The factory Qt interface is adapted for screens with resolutions of 1080*1920, 800*1280 and 720*1280. Other resolutions are not adapted. If you need to restore the landscape display, please enter the following command (for more screen-related settings, please refer to the official RK documentation Rockchip_Developer_Guide_Buildroot_Weston_CN.pdf).

```
echo "output:HDMI-A-1:rotate0" > /tmp/.weston_drm.conf
```

3.4 RTC Clock Test

The ATK-DLRK3568 development board has two hardware clocks. One is the RTC clock inside the RK809 on the main board, and the other is the RTC clock chip PCF8563 on the base board, both of which are external RTC clocks for the chips. If you need to improve the clock accuracy, a high-precision crystal oscillator is required.

Please check if there is an RTC button battery installed on the base board of the development board. You can also use a multimeter to check if the RTC battery is powered. It should be around 3.3v for normal operation. Prevent the system from losing time due to a dead RTC battery. (Note: Batteries are consumables. If they are dead, replace the button battery.)

The Linux system has two clocks, system time (software clock) and 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 powered 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 check 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 synchronize with the network time. However, this synchronized clock will not be written to the hardware clock. Therefore, you may see two different times using date and hwclock.

```
date
```

Set the current system clock

```
date -s "2023-05-20 00:00:00"
```

Check the system clock

```
date
```

Use hwclock to write to the hardware clock

```
hwclock -w
```

Check the hardware clock and verify if it is the clock set by the command "hwclock -w" above.

```
hwclock
```

```
root@ATK-DLRK356X:/# date -s "2023-05-20 00:00:00"
Sat May 20 12:00:00 AM CST 2023
root@ATK-DLRK356X:/# date
Sat May 20 12:00:06 AM CST 2023
root@ATK-DLRK356X:/# hwclock -w
root@ATK-DLRK356X:/# hwclock
Sat May 20 00:00:16 2023  0.000000 seconds
root@ATK-DLRK356X:/#
```

The above operation will only write to RTC0, and RTC0 is the internal hardware RTC clock on the core board, while RTC1 is the PCF8563. If you need to operate RTC1, follow the below steps.

Write the current time to RTC1. Note that before reading RTC, you need to write to it once first.

```
hwclock -w -f /dev/rtc1
```

Read the time of rtc1.

```
hwclock -f /dev/rtc1 --show
```

```
root@ATK-DLRK356X:/# hwclock -f /dev/rtc1 --show
Mon Jul 10 14:29:16 2023  0.000000 seconds
root@ATK-DLRK356X:/#
```

3.5 Audio Test

The power management chip rk809 integrated in the ATR-DLRK3568 core board handles audio. The onboard microphone can record audio, and a 2-ohm 8-watt speaker is connected on the back of the base board, facilitating users to play music. Note: The speaker and headphones cannot be switched; they can only play music simultaneously.

3.5.1 Playing Audio Test

Audio files can be played 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.

Note: Do not open the music or video player on the factory interface to play audio or video, otherwise it will occupy the audio device. The reason why the computer can play multiple audio simultaneously is due to software support and algorithm optimization.

```
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 /userdata/apps/src/audio/0.mp3
# Use gst-play-1.0 to play. It supports wav and other formats such as mp3.
```

3.5.2 Audio test

```
amixer cset name='Capture MIC Path' 'Main Mic'      # Use the Mic on the base plate
arecord -r 44100 -f S16_LE -d 10 record.wav         # Start recording
```

Explanation of command parameters for arecord:

- -f S16_LE: Sample in S16_LE format.
- -r 44100: Sampling rate 44.1K

- -d 10: Recording duration 10 seconds
- record.wav: Audio file generated from the recording

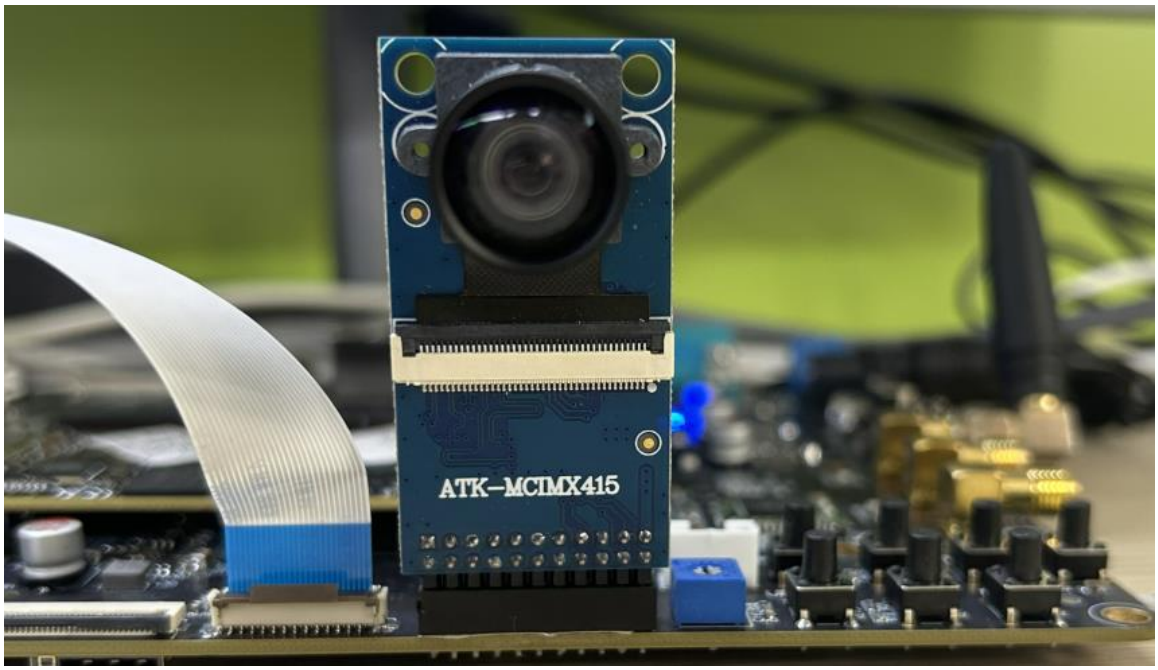
To play the audio file recorded above, note: The size of the generated file is related to the format set by the user and the duration of the recording.

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

```
root@ATK-DLRK356X:~# amixer cset name='Capture MIC Path' 'Main Mic'
numid=3,iface=MIXER,name='Capture MIC Path'
  ; type=ENUMERATED,access=rw------,values=1,items=4
  ; Item #0 'MIC OFF'
  ; Item #1 'Main Mic'
  ; Item #2 'Hands Free Mic'
  ; Item #3 'BT Sco Mic'
  : values=1
root@ATK-DLRK356X:~# arecord -r 44100 -f S16_LE -d 10 record.wav
Recording WAVE 'record.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Mono
root@ATK-DLRK356X:~# aplay record.wav
Playing WAVE 'record.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Mono
root@ATK-DLRK356X:~#
```

3.6 Camera Test

ATK-DLRK3568 supports MIPI CSI cameras. It can support the maximum 14M (4416x3312) pixel camera. The development board is equipped with two cameras specially developed by ALIENTEK, namely IMX415 and IMX335. The IMX415 is an 8M pixel camera that can record 4K (3840x2160) videos, while the IMX335 is a 5M pixel camera that can record 2K (2592x1944) videos. The installation method is that the camera lens should face outward from the development board, as shown in the following figure (taking IMX415 as an example).



The IMX415 can support the maximum 4K (3840x2160) playback. Since the MIPI screen of ALIENTEK has a resolution of 720P (720*1280) or 1080P (1080*1920), the data end has collected data exceeding the screen resolution. The kmssink will automatically adapt to the size of the screen and display it on the screen, but the data is still the size captured by the camera.

Because the default factory kernel configuration has two channels for simultaneous display (MIPI and HDMI), to ensure that the kmssink can display on your MIPI screen, you need to turn off the HDMI display channel. Execute the following command to turn off the other two channels first.

```
echo off > /sys/class/drm/card0-HDMI-A-1/status
```

Of course, you can also click on the camera app on the factory interface to test the camera. Execute the following instructions to capture a video and display it on the MIPI screen. Note: The camera app and the instructions for capturing the video cannot be used simultaneously.

```
gst-launch-1.0 v4l2src device=/dev/video0 ! video/x-raw,format=NV12,width=3840,height=2160,\
framerate=30/1 ! kmssink
```

The IMX335 can support the maximum playback of 2K (2592x1944). Since the MIPI screen of ALIENTEK has a resolution of 720P (720*1280) or 1080P (1080*1920), the data end has collected data exceeding the screen resolution. The kmssink will automatically adapt to the size of the screen and display it on the screen, but the data is still the size captured by the camera.

Collect the video and display it on the MIPI screen.

```
gst-launch-1.0 v4l2src device=/dev/video0 ! video/x-raw,format=NV12,width=2592,height=1944,\
framerate=30/1 ! kmssink
```

3.7 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, positioning and navigation; TWS headphones for posture detection, 3D sound effects; automatic counting and motion trajectory recording for smart sports equipment such as rope skipping, pull rope, abdominal wheel, etc.; smart electric toothbrush, smart water cup 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-Pin 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: 2.5×3.0×0.9mm³

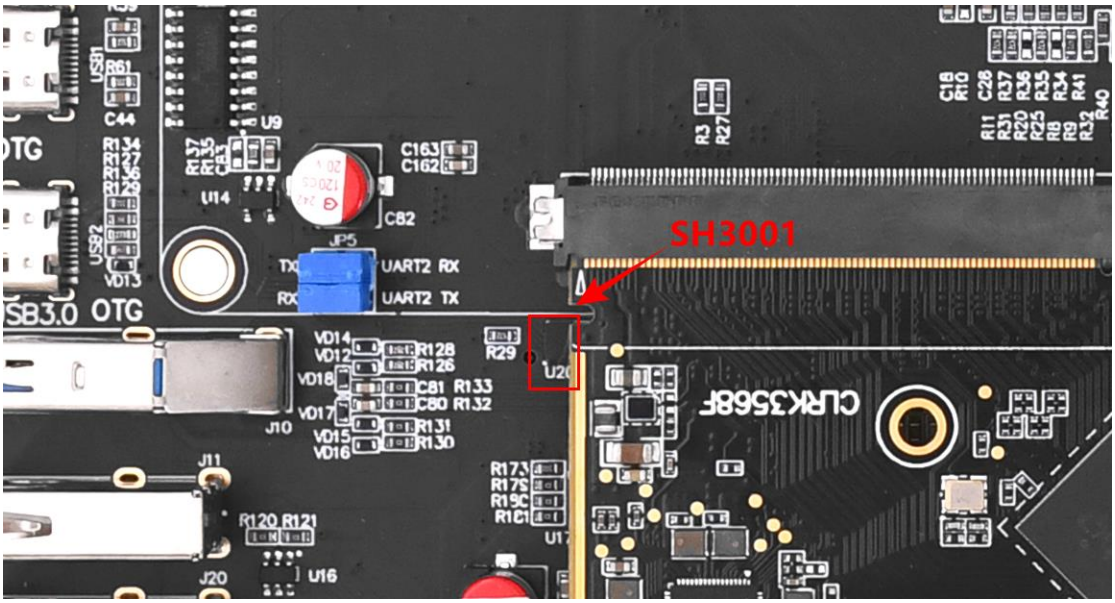


Figure 3.7-1 At the location of the development board

An 6-axis sensor SH3001 was connected to the ATK-DLRK3568 baseboard using the I2C5 interface. 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 on the A disc of the development board - Basic Data -> 01_codes -> Peripheral Test Source Code. The test method is as follows. The executable program path is located under /userdata/apps/src/pcba. The programs in the pcba directory are used as the factory test program to detect whether the hardware functions normally.

3.7.1 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.

```
/userdata/apps/src/pcba/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. Press Ctrl + c to end the test program.

```
root@ATK-DLRK356X:/# /userdata/apps/src/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 <<<
[11438.165758] gsensor_sh3001 5-0036: 0 times, read accel data is -518, 3149, 19386
[11438.203083] gsensor_sh3001 5-0036: 1 times, read accel data is -539, 3227, 19436
[11438.243389] gsensor_sh3001 5-0036: 2 times, read accel data is -556, 3201, 19480
[11438.279980] gsensor_sh3001 5-0036: 3 times, read accel data is -578, 3186, 19431
[11438.316556] gsensor_sh3001 5-0036: 4 times, read accel data is -525, 3198, 19359
[11438.353348] gsensor_sh3001 5-0036: 5 times, read accel data is -530, 3235, 19436
[11438.389839] gsensor_sh3001 5-0036: 6 times, read accel data is -587, 3211, 19435
[11438.426668] gsensor_sh3001 5-0036: 7 times, read accel data is -538, 3212, 19417
[11438.463477] gsensor_sh3001 5-0036: 8 times, read accel data is -525, 3237, 19388
[11438.500105] gsensor_sh3001 5-0036: 9 times, read accel data is -528, 3216, 19390
[11438.536729] gsensor_sh3001 5-0036: 10 times, read accel data is -536, 3254, 19346
[11438.573459] gsensor_sh3001 5-0036: 11 times, read accel data is -519, 3192, 19367
[11438.610211] gsensor_sh3001 5-0036: 12 times, read accel data is -545, 3225, 19451
[11438.646676] gsensor_sh3001 5-0036: 13 times, read accel data is -563, 3186, 19400
[11438.683476] gsensor_sh3001 5-0036: 14 times, read accel data is -553, 3120, 19421
[11438.720292] gsensor_sh3001 5-0036: 15 times, read accel data is -524, 3215, 19493
[11438.756716] gsensor_sh3001 5-0036: 16 times, read accel data is -516, 3172, 19443
[11438.793504] gsensor_sh3001 5-0036: 17 times, read accel data is -547, 3217, 19459
[11438.829987] gsensor_sh3001 5-0036: 18 times, read accel data is -546, 3243, 19496
[11438.866763] gsensor_sh3001 5-0036: 19 times, read accel data is -535, 3244, 19445
[11438.902260] gsensor_sh3001 5-0036: accel offset is -540, 3207, 3039
>>> 校准完成 <<<
offset: [-540, 3207, 3039]
-0.004785       0.013757        9.776672
-0.002991       0.009570        9.826917
0.012561        0.003589        9.775476
-0.030505       0.058020        9.791028
-0.026318       0.070581        9.796411
-0.019739       0.025122        9.795813
-0.025122       0.058618        9.804187
-0.011365       0.061609        9.794019
-0.013757       0.025720        9.822729
-0.007178       -0.012561       9.820935
-0.041272       -0.001196       9.797009
0.008374        0.026917        9.788037
0.009570        0.023926        9.776074
-0.019141       -0.023926       9.798804
0.014954        0.033496        9.820935
0.012561        -0.015552       9.798206
0.012561        -0.000598       9.770691
0.004187        -0.006580       9.802991
0.008972        0.023926        9.774878
0.008972        0.012561        9.764111
-0.029907       -0.008972       9.780261
-0.013159       -0.002393       9.725232
-0.019739       -0.005981       9.719849
^C
root@ATK-DLRK356X:/#
```

Figure 3.7-2 The read-out acceleration data

3.7.2 Angular velocity

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

/userdata/apps/src/pcba/gyro_test

```
root@ATK-DLRK356X:/# /userdata/apps/src/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 <<<
[12279.454076] gyro_sh3001 5-0036-1: 0 times, read gyro data is 27, -52, 35
[12279.491048] gyro_sh3001 5-0036-1: 1 times, read gyro data is 27, -49, 36
[12279.527984] gyro_sh3001 5-0036-1: 2 times, read gyro data is 28, -49, 34
[12279.564757] gyro_sh3001 5-0036-1: 3 times, read gyro data is 28, -53, 33
[12279.601400] gyro_sh3001 5-0036-1: 4 times, read gyro data is 27, -51, 30
[12279.637994] gyro_sh3001 5-0036-1: 5 times, read gyro data is 27, -52, 34
[12279.674702] gyro_sh3001 5-0036-1: 6 times, read gyro data is 29, -51, 33
[12279.711391] gyro_sh3001 5-0036-1: 7 times, read gyro data is 27, -55, 36
[12279.747907] gyro_sh3001 5-0036-1: 8 times, read gyro data is 26, -49, 34
[12279.784413] gyro_sh3001 5-0036-1: 9 times, read gyro data is 27, -54, 34
[12279.822565] gyro_sh3001 5-0036-1: 10 times, read gyro data is 27, -49, 34
[12279.857976] gyro_sh3001 5-0036-1: 11 times, read gyro data is 27, -53, 32
[12279.894577] gyro_sh3001 5-0036-1: 12 times, read gyro data is 25, -51, 34
[12279.931275] gyro_sh3001 5-0036-1: 13 times, read gyro data is 25, -53, 34
[12279.967933] gyro_sh3001 5-0036-1: 14 times, read gyro data is 29, -52, 32
[12280.004605] gyro_sh3001 5-0036-1: 15 times, read gyro data is 26, -56, 33
[12280.041355] gyro_sh3001 5-0036-1: 16 times, read gyro data is 30, -55, 32
[12280.077959] gyro_sh3001 5-0036-1: 17 times, read gyro data is 29, -53, 34
[12280.114683] gyro_sh3001 5-0036-1: 18 times, read gyro data is 27, -52, 37
[12280.151081] gyro_sh3001 5-0036-1: 19 times, read gyro data is 28, -52, 32
[12280.186793] gyro_sh3001 5-0036-1: gyro offset is 27, -52, 33
>>> 校准完成 <<<
offset: [27, -52, 33]
[12280.430330] gyro_sh3001 5-0036-1: sensor on: starting poll sensor data 30ms
0.061035        0.000000        0.061035
-0.122070       0.244141        0.122070
0.000000        0.061035        0.000000
0.000000        0.427246        0.122070
-0.061035       0.122070        0.061035
-0.061035       0.000000        -0.061035
-0.305176       -0.122070       0.061035
0.061035        -0.122070       0.000000
0.000000        0.061035        0.000000
0.000000        0.000000        0.122070
0.061035        -0.061035       0.183105
0.122070        -0.061035       0.000000
0.000000        0.061035        0.061035
0.000000        -0.122070       0.000000
0.061035        -0.244141       0.122070
0.000000        0.000000        0.061035
0.000000        -0.061035       0.122070
0.061035        -0.122070       0.122070
0.061035        -0.183105       0.000000
```

Figure 3.7-3 The read-out angular velocity data

3.8 USB test

3.8.1 USB HOST test

Use a FAT32 formatted USB drive, plug it directly into the USB interface on the ATK-DLRK3568 baseboard (any of the three USB interfaces will work, two of which are USB 2.0, and the blue interface is USB 3.0 (note: the USB 3.0 interface shares the same functionality as OTG. When OTG is plugged in, the USB 3.0 interface cannot be used. USB 3.0 can function as a host or slave, and can be used for burning or as an external USB drive.) As shown in the following figure.

```
root@ATK-DLRK356X:~# [61423.650680] usb 1-1: new high-speed USB device number 4 using ehci-platform
[61423.802542] usb 1-1: New USB device found, idVendor=aaaa, idProduct=8816, bcdDevice=13.08
[61423.802628] usb 1-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[61423.802661] usb 1-1: Product: MXT USB Device
[61423.802689] usb 1-1: Manufacturer: MXTronics
[61423.802715] usb 1-1: SerialNumber: 150101v01
[61423.805316] usb-storage 1-1:1.0: USB Mass Storage device detected
[61423.807537] scsi host1: usb-storage 1-1:1.0
[61424.829370] scsi 1:0:0:0: Direct-Access     MXT-USB  Storage Device   1501 PQ: 0 ANSI: 0 CCS
[61424.834771] sd 1:0:0:0: [sda] 61120512 512-byte logical blocks: (31.3 GB/29.1 GiB)
[61424.835599] sd 1:0:0:0: [sda] Write Protect is off
[61424.836315] sd 1:0:0:0: [sda] No Caching mode page found
[61424.836334] sd 1:0:0:0: [sda] Assuming drive cache: write through
[61424.840829]  sda: sda1
[61424.843703] sd 1:0:0:0: [sda] Attached SCSI removable disk

root@ATK-DLRK356X:~#
root@ATK-DLRK356X:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       5.9G  1.6G  4.1G  29% /
devtmpfs        977M     0  977M   0% /dev
tmpfs           986M     0  986M   0% /dev/shm
tmpfs           986M  1.2M  985M   1% /tmp
tmpfs           986M  348K  986M   1% /run
/dev/mmcblk0p7  123M   13M  104M  11% /oem
/dev/mmcblk0p8   23G   25K   23G   1% /userdata
/dev/sda1        30G   64M   30G   1% /media/udisk0
root@ATK-DLRK356X:~#
```

Capacity of USB flash drive

"sda" represents the node name, and "sda1" represents a partition of this node.

Mount the partition sda1 to the directory /media/udisk0

Figure 3.8-1 Information about USB drive mounting

We can directly enter the "/media/usb0" directory to perform read and write file operations.

Read speed test:

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

```
[root@ATK-RV1126:/]# hdparm -t /dev/sda1

/dev/sda1:
Timing buffered disk reads:   64 MB in 3.01 seconds = 21741 kB/s
[root@ATK-RV1126:/]#
```

Figure 3.8-2 Read speed evaluation

Write speed test:

```
time dd if=/dev/zero of=/media/udisk0/test bs=1024k count=100 conv=fdatasync
rm /media/udisk0/test
```

Here represents the writing of a 100MiB file, with a writing speed of 14.5 MB/s. In fact, the larger the written file, the closer the average writing speed is to the actual value.

3.8.1.1 USB2.0 Test

Note: The test of read and write speeds is related to the maximum read and write speed of the personal card reader or USB/TF card.

Using the hdparm command for read speed test, here it took 3.05 seconds to read 47MB. The speed may not be very accurate because the sampling is too small.

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

```
root@ATK-DLRK356X:~# hdparm -t /dev/sda1

/dev/sda1:
Timing buffered disk reads:   47 MB in 3.05 seconds = 15734 kB/s
root@ATK-DLRK356X:~#
```

Write a speed test.

```
time dd if=/dev/zero of=/media/udisk0/test bs=1024k count=100 conv=fdatasync
# Note that the instruction is too long, resulting in abnormal display of the instruction, but it does not affect the operation.
```

```
root@ATK-DLRK356X:~#
100+0 records inX:~# time dd if=/dev/zero of=/media/udisk0/test bs=1024k count=100 conv=fdatasync
100+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 10.5146 s, 10.0 MB/s

real    0m10.529s
user    0m0.000s
sys     0m0.235s
root@ATK-DLRK356X:~#
```

Use the "dd" command for reading. Due to the Linux system mechanism, usually 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=/media/udisk0/test of=/dev/null bs=1024k  # Read speed test
rm /media/udisk0/test                                # Delete this file after reading it.
```

```
root@ATK-DLRK356X:~# echo 3 > /proc/sys/vm/drop_caches
[ 1006.666602] sh (893): drop_caches: 3
root@ATK-DLRK356X:~# time dd if=/media/udisk0/test of=/dev/null bs=1024k
100+0 records in
100+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 6.37117 s, 16.5 MB/s

real    0m6.401s
user    0m0.004s
sys     0m0.425s
root@ATK-DLRK356X:~#
```

3.8.1.2 USB2.0 Test

As mentioned earlier, the OTG interface is compatible with the USB3.0 interface. When we want to use the USB3.0 interface as the host, the OTG connection cable does not need to be connected.

Similarly, the testing method is the same as the testing steps in 3.8.1.1. Note: The read/write speed is most closely related to the speed of the USB flash drive.

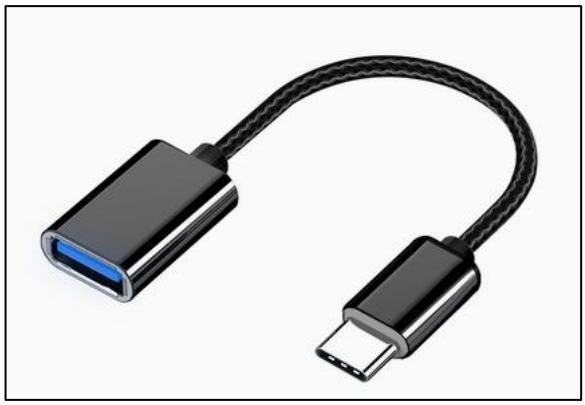
3.8.2 OTG Slave Device Test

The OTG interface on the ATK-DLRK3568 board can be used as ADB. We connect the OTG interface to the PC (computer) using a USB Type-C connection cable. By using adb commands, communication with the development board can be achieved.

3.8.3 OTG Host Device Test

Connect a USB to Type-C adapter (if the user doesn't have this adapter, this experiment can be skipped) to the USB port, and insert it into the OTG interface. The test is conducted in the same way as in 3.8.1.1 section.

The USB to Type-C adapter is as follows. Please purchase it as needed by yourself.



3.9 Network Test

The ATK-DLRK3568 baseboard is equipped with two gigabit network chips, which are adaptive to 10/100/1000M. To test the network using the file system, use the following command. Plug the network cable into one of the two network ports and ensure that the network cable can access the internet. Check the IP obtained from the network.

```
ifconfig
```

```
root@ATK-DLRK356X:~# ifconfig
eth0      Link encap:Ethernet  HWaddr F6:CB:4E:B0:6E:6F
          inet addr:192.168.6.243  Bcast:192.168.6.255  Mask:255.255.255.0
          inet6 addr: fe80::f4cb:4eff:feb0:6e6f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5916 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1479 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:641715 (626.6 KiB)  TX bytes:99820 (97.4 KiB)
          Interrupt:38

eth1      Link encap:Ethernet  HWaddr F2:CB:4E:B0:6E:6F
          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:50

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:128 errors:0 dropped:0 overruns:0 frame:0
          TX packets:128 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:10241 (10.0 KiB)  TX bytes:10241 (10.0 KiB)

root@ATK-DLRK356X:~#
```

Figure 3.9-1 eth0 has obtained the IP address.

Check if the network can access the internet. 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-DLRK356X:~# ping www.baidu.com
PING www.a.shifen.com (14.119.104.189) 56(84) bytes of data.
64 bytes from 14.119.104.189 (14.119.104.189): icmp_seq=1 ttl=54 time=5.83 ms
64 bytes from 14.119.104.189 (14.119.104.189): icmp_seq=2 ttl=54 time=5.87 ms
64 bytes from 14.119.104.189 (14.119.104.189): icmp_seq=3 ttl=54 time=6.24 ms
64 bytes from 14.119.104.189 (14.119.104.189): icmp_seq=4 ttl=54 time=6.37 ms
64 bytes from 14.119.104.189 (14.119.104.189): icmp_seq=5 ttl=54 time=5.83 ms
64 bytes from 14.119.104.189 (14.119.104.189): icmp_seq=6 ttl=54 time=5.91 ms
^C
--- www.a.shifen.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5009ms
rtt min/avg/max/mdev = 5.828/6.006/6.368/0.215 ms
root@ATK-DLRK356X:~#
```

Figure 3.9-2 ping Baidu

At the same time, the other network port is also tested in the same way. Plug the network cable into the other network 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, seeing "1Gbps/Full" indicates it is a gigabit network.

```
root@ATK-DLRK356X:~# [ 3945.431471] rk_gmac-dwmac fe2a0000.ethernet eth1: Link is Up - 1Gbps/Full - flow control rx/tx
[ 3945.431640] IPv6: ADDRCONF(NETDEV_CHANGE): eth1: link becomes ready
```

Figure 3.9-3 Identified as a gigabit network

The connection speed of the gigabit network can be tested using the iperf3 command

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

View Ubuntu IP

ifconfig

```
alientek@ubuntu:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.6.152  netmask 255.255.255.0  broadcast 192.168.6.255
        inet6 fe80::3521:4027:7e42:43f6  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:f1:b3:b6  txqueuelen 1000  (以太网)
        RX packets 1959437  bytes 221638999 (221.6 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1874615  bytes 2626772140 (2.6 GB)
        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 2585  bytes 312137 (312.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2585  bytes 312137 (312.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

alientek@ubuntu:~$
```

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

iperf3 -s

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

ATK-DLRK3568 is the client. By using the iperf3 command, simply input the IP address of the server (Ubuntu) to proceed.

iperf3 -c 192.168.6.152 -i 1

Command parsing:

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

```
root@ATK-DLRK356X:~# iperf3 -c 192.168.6.152 -i 1
Connecting to host 192.168.6.152, port 5201
[  5] local 192.168.6.131 port 53258 connected to 192.168.6.152 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec   113 MBytes   949 Mbits/sec    0    515 KBytes
[  5]   1.00-2.00   sec   112 MBytes   942 Mbits/sec    0    540 KBytes
[  5]   2.00-3.00   sec   112 MBytes   939 Mbits/sec    0    540 KBytes
[  5]   3.00-4.00   sec   112 MBytes   942 Mbits/sec    0    540 KBytes
[  5]   4.00-5.00   sec   112 MBytes   942 Mbits/sec    0    567 KBytes
[  5]   5.00-6.00   sec   112 MBytes   939 Mbits/sec    0    598 KBytes
[  5]   6.00-7.00   sec   112 MBytes   936 Mbits/sec    0    634 KBytes
[  5]   7.00-8.00   sec   112 MBytes   938 Mbits/sec    0    634 KBytes
[  5]   8.00-9.00   sec   112 MBytes   941 Mbits/sec    0    634 KBytes
[  5]   9.00-10.00  sec   112 MBytes   943 Mbits/sec    0    634 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec  1.10 GBytes   941 Mbits/sec    0             sender
[  5]   0.00-10.00  sec  1.09 GBytes   940 Mbits/sec                  receiver

iperf Done.
root@ATK-DLRK356X:~#
```

Figure 3.9-4 iperf3 test results

The Bitrate received and sent by the test are approximately 900 Mbits/sec, which is the speed of a gigabit network (please do not run other 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. None of the devices can be of the hundred-megabit type; otherwise, the test result will be a hundred-megabit network.

3.10 Onboard SDIO WIFI test

The ATK-DLRK3568 board is equipped with an RTL8852bs chip, which is a dual-function chip for WIFI and Bluetooth. Run the following command to test WIFI. If you have connected a MIPI screen or an HMDI screen, you can click on the "Settings App" > "Wireless Local Area Network Test" on the interface.

3.10.1 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.

We can connect to WiFi using the following instructions.

```
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 the WIFI scan function. Multiple scans are available.
connmanctl> services        # List the scanned WIFI list
```

The command execution result is as follows:

```
root@ATK-DLRK356X:~# connmanctl
connmanctl> enable wifi
Error wifi: Already enabled
connmanctl> agent on
Agent registered
connmanctl> scan wifi
Scan completed for wifi
connmanctl> services
*AR Wired                ethernet_f2cb4eb06e6f_cable
    Linux-WIFI6_5G       wifi_b4d65cbd3d64_4c696e75782d57494649365f3547_managed_psk
    Linux-WIFI6          wifi_b4d65cbd3d64_4c696e75782d5749464936_managed_psk
    ALIENTEK-Linux_Wi-Fi5 wifi_b4d65cbd3d64_414c49454e54454b2d4c696e75785f57692d466935_managed_psk
    ALIENTEK-Linux       wifi_b4d65cbd3d64_414c49454e54454b2d4c696e7578_managed_psk
                         wifi_b4d65cbd3d64_hidden_managed_psk
    RICH901-5G           wifi_b4d65cbd3d64_524943483930312d3547_managed_psk
    ALIENTEK-WX          wifi_b4d65cbd3d64_414c49454e54454b2d5758_managed_psk
    ChinaNet-ymyk-5G     wifi_b4d65cbd3d64_4368696e614e65742d796d796b2d3547_managed_psk
    RICH901              wifi_b4d65cbd3d64_52494348393031_managed_psk
    ChinaNet-ymyk        wifi_b4d65cbd3d64_4368696e614e65742d796d796b_managed_psk
    GZXH HUAWEI          wifi_b4d65cbd3d64_475a584820485541574549_managed_psk
    GZXH HUAWEI_Wi-Fi5   wifi_b4d65cbd3d64_475a584820485541574549 5f57692d466935_managed_psk
    ALIENTEK-7机房     wifi_b4d65cbd3d64_414c49454e54454b2d37e69cbae688bf_managed_psk
    ALIENTEK-SYS         wifi_b4d65cbd3d64_414c49454e54454b2d535953_managed_psk
    ATK-HUAWEI           wifi_b4d65cbd3d64_41544b2d485541574549_managed_psk
    Tenda_480398         wifi_b4d65cbd3d64_54656e64615f343830333938_managed_psk
    ALIENTEK-SYS_5G      wifi_b4d65cbd3d64_414c49454e54454b2d5359535f3547_managed_psk
    ALIENTEK-XHYS        wifi_b4d65cbd3d64_414c49454e54454b2d58485953_managed_psk
    ChinaNet-kss2        wifi_b4d65cbd3d64_4368696e614e65742d6b737332_managed_psk
connmanctl>
```

Figure 3.10-1 List the scanned WIFI

From the list of scanned WIFI, use "connect XXXX" to connect, then enter the password and press Enter to confirm. For example, if I want to connect to "ALIENTEK-Linux", (note that due to the compatibility issue of IOS mobile phone hotspots, it may not be able to connect to the hotspot sent by the IOS mobile phone) the result of the operation is as follows:

connect wifi_b4d65cbd3d64_414c49454e54454b2d4c696e7578_managed_psk # connect

exit # Exit the terminal

```
connmanctl> connect wifi_b4d65cbd3d64_414c49454e54454b2d4c696e7578_managed_psk
Agent RequestInput wifi_b4d65cbd3d64_414c49454e54454b2d4c696e7578_managed_psk
  Passphrase = [ Type=psk, Requirement=mandatory, Alternates=[ WPS ] ]
  WPS = [ Type=wpspin, Requirement=alternate ]
Passphrase? 15902020203
connmanctl> [  836.882968] IPv6: ADDRCONF(NETDEV_CHANGE): wlan0: link becomes ready
Connected wifi_b4d65cbd3d64_414c49454e54454b2d4c696e7578_managed_psk
connmanctl> exit
root@ATK-DLRK356X:~#
```

Press the Enter key after entering the password.

Seeing this message indicates that a connection has been established.

Figure 3.10-2 Connection successful - WIFI

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

```
root@ATK-DLRK356X:~# ifconfig
eth0      Link encap:Ethernet  HWaddr F6:CB:4E:B0:6E:6F
          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:38

eth1      Link encap:Ethernet  HWaddr F2:CB:4E:B0:6E:6F
          inet addr:192.168.6.131  Bcast:192.168.6.255  Mask:255.255.255.0
          inet6 addr: fe80::f0cb:4eff:feb0:6e6f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1152 errors:0 dropped:0 overruns:0 frame:0
          TX packets:183 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:185743 (181.3 KiB)  TX bytes:15685 (15.3 KiB)
          Interrupt:50

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:136 errors:0 dropped:0 overruns:0 frame:0
          TX packets:136 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:10504 (10.2 KiB)  TX bytes:10504 (10.2 KiB)

wlan0     Link encap:Ethernet  HWaddr B4:D6:5C:BD:3D:64
          inet addr:192.168.6.199  Bcast:192.168.6.255  Mask:255.255.255.0
          inet6 addr: fe80::b6d6:5cff:febd:3d64/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:60 overruns:0 frame:0
          TX packets:56 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:96107 (93.8 KiB)  TX bytes:5194 (5.0 KiB)

root@ATK-DLRK356X:~#
```

Figure 3.10-3 Check the obtained IP address.

Ping Baidu to test connectivity, or you can ping the gateway to test the connectivity of the WIFI.

ping www.baidu.com -I wlan0 # Ping Baidu test

ping 192.168.6.1 -I wlan0

Change the ping gateway to your own gateway, such as 192.168.x.x.

```
root@ATK-DLRK356X:~# ping www.baidu.com -I wlan0
PING www.baidu.com (14.119.104.254) from 192.168.6.199 wlan0: 56(84) bytes of data.
64 bytes from 14.119.104.254 (14.119.104.254): icmp_seq=1 ttl=54 time=6.71 ms
64 bytes from 14.119.104.254 (14.119.104.254): icmp_seq=2 ttl=54 time=31.7 ms
64 bytes from 14.119.104.254 (14.119.104.254): icmp_seq=3 ttl=54 time=9.22 ms
64 bytes from 14.119.104.254 (14.119.104.254): icmp_seq=4 ttl=54 time=8.78 ms
64 bytes from 14.119.104.254 (14.119.104.254): icmp_seq=5 ttl=54 time=7.79 ms
64 bytes from 14.119.104.254 (14.119.104.254): icmp_seq=6 ttl=54 time=11.4 ms
64 bytes from 14.119.104.254 (14.119.104.254): icmp_seq=7 ttl=54 time=11.5 ms
64 bytes from 14.119.104.254 (14.119.104.254): icmp_seq=8 ttl=54 time=12.6 ms
64 bytes from 14.119.104.254 (14.119.104.254): icmp_seq=9 ttl=54 time=9.37 ms
^C
--- www.baidu.com ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 9397ms
rtt min/avg/max/mdev = 6.707/12.107/31.678/7.140 ms
root@ATK-DLRK356X:~#
```

Figure 3.10-4 Ping Baidu test

3.10.2 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 for turning on the hotspot are all placed in the /root/shell/wifi directory, as shown in the following figure.

```
root@ATK-DLRK356X:/root/shell/wifi# ls
autoconnect.sh  dnsmasq.conf  hostapd.conf  wifi_apmode.sh
root@ATK-DLRK356X:/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=111
```

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-DLRK356X:/root/shell/wifi# ./wifi_apmode.sh
[  144.873989] IPv6: ADDRCONF(NETDEV_UP): wlan0: link is not ready
Disabled wifi
[  146.502302] [BT_RFKILL]: rfkill_rk_set_power: set bt wake_host high!
[  146.555707] [BT_RFKILL]: ENABLE UART_RTS
[  146.662360] [BT_RFKILL]: DISABLE UART_RTS
[  146.662576] [BT_RFKILL]: bt turn on power
[  146.662665] [BT_RFKILL]: Request irq for bt wakeup host
[  146.662752] [BT_RFKILL]: ** disable irq
[  147.691703] IPv6: ADDRCONF(NETDEV_UP): wlan0: link is not ready
Configuration file: hostapd.conf
Using interface wlan0 with hwaddr c0:8e:da:2c:fb:38 and ssid "alientek_softap"
[  149.067081] IPv6: ADDRCONF(NETDEV_CHANGE): wlan0: link becomes ready
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
root@ATK-DLRK356X:/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.

3.11 Built-in Bluetooth testing

Bluetooth development relies on Buildroot. The files such as hci_uart.ko, which are Bluetooth driver modules, 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 Port Protocol, maps the data transmission of serial port devices to 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 to transmit 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 to transmit high-quality audio data between Bluetooth devices.
- AVRCP: Audio/Video Remote Control Protocol, used to transmit media control commands between Bluetooth devices.

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

The Buildroot system path contains the onboard Bluetooth driver module /system/lib/modules/hci_uart.ko.

3.11.1 Bluetooth Initialization

The Bluetooth initialization script bt_init.sh has been placed in the /root/shell/bluetooth directory in advance. The content of bt_init.sh is as follows. In fact, Rockchip also has a bt_init.sh script located in the /usr/bin directory. The bt_init.sh script written by me has been optimized a little.

```
#!/bin/bash
# Kill the process to prevent the serial port from being occupied
if [[ "$(pidof rtk_hciattach)" != "" ]]
then
                kill -9 $(pidof rtk_hciattach)
fi
echo 0 > /sys/class/rfkill/rfkill0/state # Power off
echo 0 > /proc/bluetooth/sleep/btwrite
sleep 1
echo 1 > /sys/class/rfkill/rfkill0/state # Power on
echo 1 > /proc/bluetooth/sleep/btwrite
sleep 1
! test -d /sys/module/hci_uart && insmod /system/lib/modules/hci_uart.ko

rtk_hciattach -n -s 115200 /dev/ttyS8 rtk_h5 &
sleep 1
/usr/libexec/bluetooth/bluetoothd -C &
hciconfig hci0 piscan
```

In the above script, Bluetooth uses serial port communication. The serial port used is ttyS8, and the Bluetooth initialization is performed using rtk_hciattach.

Simply input "bt_init.sh" in the serial port terminal to initialize Bluetooth. After seeing the content in the red box in the following picture, it indicates that the Bluetooth initialization is successful.

/root/shell/bluetooth/bt_init.sh

```
root@ATK-DLRK356X:/# /root/shell/bluetooth/bt_init.sh
[   38.121199] [BT_RFKILL]: bt shut off power
[   40.131863] [BT_RFKILL]: rfkill_rk_set_power: set bt wake_host high!
[   40.184949] [BT_RFKILL]: ENABLE UART_RTS
[   40.291730] [BT_RFKILL]: DISABLE UART_RTS
[   40.291900] [BT_RFKILL]: bt turn on power
[   40.291998] [BT_RFKILL]: Request irq for bt wakeup host
[   40.292152] [BT_RFKILL]: ** disable irq
[   42.331709] hci_uart: loading out-of-tree module taints kernel.
[   42.335038] Bluetooth: HCI UART driver ver 2.2.fdfa5dd.20220110-123627
[   42.335089] Bluetooth: HCI H4 protocol initialized
[   42.335098] Bluetooth: HCI Realtek H5 protocol initialized
[   42.335109] rtk_btcoex: rtk_btcoex_init: version: 1.2
[   42.335116] rtk_btcoex: create workqueue
[   42.335597] rtk_btcoex: alloc buffers 1792, 2432 for ev and l2
[   42.345769] of_dma_request_slave_channel: dma-names property of node '/serial@fe6c0000' missing or empty
[   42.345893] ttyS8 - failed to request DMA, use interrupt mode
Realtek Bluetooth :Realtek Bluetooth init uart with init speed:115200, type:HCI UART H5
Realtek Bluetooth :Realtek hciattach version 3.1.2c27de6.20220110-123628

Realtek Bluetooth :Use epoll
Realtek Bluetooth :[SYNC] Get SYNC Resp Pkt
Realtek Bluetooth :[CONFIG] Get SYNC pkt
Realtek Bluetooth :[CONFIG] Get CONFG pkt
Realtek Bluetooth :[CONFIG] Get CONFG resp pkt
Realtek Bluetooth :dic is 1, cfg field 0x14
Realtek Bluetooth :H5 init finished

Realtek Bluetooth :Realtek H5 IC
Realtek Bluetooth :Receive cmd complete event of command: 1001
Realtek Bluetooth :HCI Version 0x0b
Realtek Bluetooth :HCI Revision 0x000b
Realtek Bluetooth :LMP Subversion 0x8852
Realtek Bluetooth :Receive cmd complete event of command: fc6d
Realtek Bluetooth :Read ROM version 01
Realtek Bluetooth :LMP Subversion 0x8852
Realtek Bluetooth :EVersion 1
Realtek Bluetooth :IC: RTL8852BS
Realtek Bluetooth :Firmware/config: rtl8852bs_fw, rtl8852bs_config
Realtek Bluetooth :Couldnt open extra config /opt/rtk_btconfig.txt, No such file or directory
Realtek Bluetooth :Couldnt access customer BT MAC file /opt/bdaddr
Realtek Bluetooth :Origin cfg len 33
Realtek Bluetooth :55 ab 23 87 1b 00 0c 00 10 02 80 92 04 50 c5 ea
Realtek Bluetooth :19 e1 1b fd af 5f 01 a4 0b 8d 00 01 fa 8f 00 01
Realtek Bluetooth :bf
Realtek Bluetooth :Config baudrate: 04928002
Realtek Bluetooth :uart flow ctrl: 1
Realtek Bluetooth :Vendor baud from Config file: 04928002
Realtek Bluetooth :New cfg len 33
Realtek Bluetooth :55 ab 23 87 1b 00 0c 00 10 02 80 92 04 50 c5 ea
Realtek Bluetooth :19 e1 1b fd af 5f 01 a4 0b 8d 00 01 fa 8f 00 01
Realtek Bluetooth :bf
Realtek Bluetooth :Load FW /lib/firmware/rtlbt/rtl8852bs_fw OK, size 49852
Realtek Bluetooth :rtb_get_fw_project_id: opcode 0, len 1, data 20
Realtek Bluetooth :FW version 0xddb743b1, Patch num 2
Realtek Bluetooth :Chip id 0x0001
Realtek Bluetooth :Chip id 0x0002
Realtek Bluetooth :Patch length 0x83b5
Realtek Bluetooth :Start offset 0x00003ec0
Realtek Bluetooth :Svn version: 3985952765
Realtek Bluetooth :Coexistence: BTCOEX_20210331-0606

Realtek Bluetooth :FW  exists, Config file  exists
Realtek Bluetooth :Total len 33750 for fwc
Realtek Bluetooth :baudrate in change speed command: 0x02 0x80 0x92 0x04
Realtek Bluetooth :Receive cmd complete event of command: fc17
Realtek Bluetooth :Received cc of vendor change baud
Realtek Bluetooth :Final speed 1500000
Realtek Bluetooth :end_idx: 133, lp_len: 234, additional pkts: 6

Realtek Bluetooth :Start downloading...
Realtek Bluetooth :Send additional packet 7
Realtek Bluetooth :Send additional packet 8
Realtek Bluetooth :Send additional packet 9
Realtek Bluetooth :Send additional packet 10
Realtek Bluetooth :Send additional packet 11
Realtek Bluetooth :Last packet 140
Realtek Bluetooth :Send last pkt
Realtek Bluetooth :Enable host hw flow control
Realtek Bluetooth :h5_hci_reset: Issue hci reset cmd
Realtek Bluetooth :Receive cmd complete event of command: 0c03
Realtek Bluetooth :Received cc of hci reset cmd
Realtek Bluetooth :Init Process finished
[   42.969017] Bluetooth: h5_open
[   42.969113] Bluetooth: hci_uart_register_dev
Realtek Bluetooth :Realtek Bluetooth post process
Realtek Bluetooth :Device setup colmplete
[   42.971920] rtk_btcoex: Open BTCOEX
[   42.974514] rtk_btcoex: BTCOEX hci_rev 0xddb7
[   42.974600] rtk_btcoex: BTCOEX lmp_subver 0x43b1
[   45.071801] Bluetooth: __hci_uart_flush: hdev 000000002b30d156 tty 0000000091635b4b
[   45.071953] Bluetooth: hci_uart_close: hdev 000000002b30d156
[   45.071984] Bluetooth: __hci_uart_flush: hdev 000000002b30d156 tty 0000000091635b4b
[   45.072013] rtk_btcoex: Close BTCOEX
[   45.072048] rtk_btcoex: -x
[   47.361601] rtk_btcoex: Open BTCOEX
[   47.363800] rtk_btcoex: BTCOEX hci_rev 0xddb7
[   47.363825] rtk_btcoex: BTCOEX lmp_subver 0x43b1
root@ATK-DLRK356X:/# _
```

Figure 3.11-1 Bluetooth Initialization

3.11.2 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 approach 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 approach is suitable for scenarios requiring serial port communication, such as wireless controllers, wireless sensors, etc.

The author has written a script for enabling Bluetooth RFCOMM to establish an RFCOMM channel, 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-DLRK356X:/# /root/shell/bluetooth/rfcomm_init.sh
[ 1050.698799] Bluetooth: hu 00000000229fa317 retransmitting 1 pkts
rfcomm init finished! please use bluetoothctl cmd to pair your device and then connect it!
root@ATK-DLRK356X:/# Waiting for connection on channel 1

root@ATK-DLRK356X:/#
```

Figure 3.11-2 Start RFCOMM

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

3.11.3 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	# Start the agent
discoverable on	# Enable detectability
scan on	# Activate scanning of devices

```
root@ATK-DLRK356X:/# bluetoothctl
Agent registered
[bluetooth]# power on
Changing power on succeeded
[bluetooth]# agent on
Agent is already registered
[bluetooth]# discoverable on
[bluetooth]# [ 2227.438287] Bluetooth: hu 00000000229fa317 retransmitting 1 pkts
Changing discoverable on succeeded
[CHG] Controller FF:FF:FF:FF:FF:94 Discoverable: yes
[bluetooth]# scan on
[bluetooth]# [ 2239.091149] Bluetooth: hu 00000000229fa317 retransmitting 1 pkts
[ 2239.095535] rtk_btcoex: hci (periodic)inq start
Discovery started
[CHG] Controller FF:FF:FF:FF:FF:94 Discovering: yes
[bluetooth]# [ 2239.572432] Bluetooth: Unknown advertising packet type: 0x100
[NEW] Device A6:D7:3C:42:4A:D3 A6-D7-3C-42-4A-D3
[bluetooth]# [ 2240.922025] Bluetooth: Unknown advertising packet type: 0x100
[NEW] Device 75:48:E6:E2:C1:45 75-48-E6-E2-C1-45
[CHG] Device A6:D7:3C:42:4A:D3 EirData:
  02 01 06 06 ff 40 00 00 6d 02 11 06 59 c1 4a 4a  .....@..m...Y.JJ
  ed 2b 73 b5 59 4e f4 4e 00 00 2a 80 00 00 00 00  .+s.YN.N..*.....
  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

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

scan off

As shown in the figure below, the author scanned the Bluetooth MAC address of the Android phone and, after confirmation, the pairing process could be initiated.

```
[bluetooth]# [  992.739140] rtk_btcoex: hci (periodic)inq cancel/exit
Discovery stopped
[CHG] Controller 48:8F:4C:10:2E:BD Discovering: no
[CHG] Device 55:EF:13:8A:AC:5C RSSI is nil
[CHG] Device D2:32:E0:5D:64:3F RSSI is nil
[CHG] Device 6C:B3:EC:CF:F8:1D TxPower is nil
[CHG] Device 6C:B3:EC:CF:F8:1D RSSI is nil
[CHG] Device 75:A5:9E:4A:4F:01 TxPower is nil
[CHG] Device 75:A5:9E:4A:4F:01 RSSI is nil
[CHG] Device 73:BE:5F:23:0A:E8 TxPower is nil
[CHG] Device 73:BE:5F:23:0A:E8 RSSI is nil
[CHG] Device E9:E6:5E:39:31:13 RSSI is nil
[CHG] Device 4C:D8:BC:5F:3E:4D TxPower is nil
[CHG] Device 4C:D8:BC:5F:3E:4D RSSI is nil
[CHG] Device 7E:AA:B1:1E:11:CE TxPower is nil
[CHG] Device 7E:AA:B1:1E:11:CE RSSI is nil
[CHG] Device E7:E8:E6:3F:18:81 RSSI is nil
[CHG] Device 0C:AE:B0:FB:5B:0E RSSI is nil
[CHG] Device 44:A9:D1:ED:86:82 RSSI is nil
[CHG] Device AC:D6:18:5C:C2:28 RSSI is nil
[CHG] Device 74:D2:85:88:BD:59 RSSI is nil
[CHG] Device 64:68:76:9D:A5:71 RSSI is nil
[CHG] Device 22:56:BD:A8:CB:63 RSSI is nil
[CHG] Device 5F:31:3F:31:78:9D RSSI is nil
[CHG] Device 53:DE:A9:F4:3A:10 TxPower is nil
[CHG] Device 53:DE:A9:F4:3A:10 RSSI is nil
[CHG] Device 56:D2:E0:53:93:43 TxPower is nil
[CHG] Device 56:D2:E0:53:93:43 RSSI is nil
[CHG] Device 5A:78:8A:55:04:7F TxPower is nil
[CHG] Device 5A:78:8A:55:04:7F RSSI is nil
[CHG] Device 7C:2A:DB:03:DF:77 RSSI is nil
[CHG] Device 00:13:EF:A0:04:F1 TxPower is nil
[CHG] Device 00:13:EF:A0:04:F1 RSSI is nil
[CHG] Device 6F:2D:97:F5:77:30 RSSI is nil
[CHG] Device 5C:D5:55:04:3A:B0 TxPower is nil
[CHG] Device 5C:D5:55:04:3A:B0 RSSI is nil
[CHG] Device 42:42:D1:55:64:5D TxPower is nil
[CHG] Device 42:42:D1:55:64:5D RSSI is nil
[CHG] Device 60:AB:67:65:93:E8 RSSI is nil
[CHG] Device 43:32:ED:80:11:D7 TxPower is nil
```

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]# [ 2922.905337] rtk_btcoex: hci create connection, start paging
[ 2923.925376] rtk_btcoex: connected, handle 0001, status 0x00
[ 2923.925474] rtk_btcoex: Page success
[ 2923.985388] rtk_btcoex: io capability request
[CHG] Device 60:AB:67:65:93:E8 Connected: yes
Request confirmation
[agent] Confirm passkey 445388 (yes/no): yes
[蓝牙测试]# [ 2937.012081] rtk_btcoex: link key notify
[ 2937.055204] rtk_btcoex: l2cap op 2, len 16, out 1
[ 2937.055389] rtk_btcoex: TX l2cap conn req, hndl 0x0001, PSM 0x0001, scid 0x0040
[ 2937.055541] rtk_btcoex: PSM(0x0001) do not need parse
[ 2937.081083] rtk_btcoex: l2cap op 3, len 20, out 0
[ 2937.081293] rtk_btcoex: RX l2cap conn rsp, hndl 0x0001, dcid 0x0040, scid 0x0040, result 0x0000
[ 2937.171138] rtk_btcoex: l2cap op 2, len 16, out 0
[ 2937.171366] rtk_btcoex: RX l2cap conn req, hndl 0x0001, PSM 0x0001, scid 0x0041
[ 2937.171399] rtk_btcoex: PSM(0x0001) do not need parse
[ 2937.171639] rtk_btcoex: l2cap op 3, len 20, out 1
[ 2937.171780] rtk_btcoex: TX l2cap conn rsp, hndl 0x0001, dcid 0x0041, scid 0x0041, result 0x0000
[ 2937.219562] rtk_btcoex: l2cap op 6, len 16, out 0
[ 2937.219802] rtk_btcoex: RX l2cap disconn req, hndl 0x0001, dcid 0x0041, scid 0x0041
[ 2937.219900] rtk_btcoex: handle_l2cap_discon_req: handle 0x0001, dcid 0x0041, scid 0x0041, dir 0
[ 2937.229641] rtk_btcoex: l2cap op 2, len 16, out 0
[ 2937.229885] rtk_btcoex: RX l2cap conn req, hndl 0x0001, PSM 0x0001, scid 0x0042
[ 2937.229942] rtk_btcoex: PSM(0x0001) do not need parse
[ 2937.230167] rtk_btcoex: l2cap op 3, len 20, out 1
[ 2937.230317] rtk_btcoex: TX l2cap conn rsp, hndl 0x0001, dcid 0x0041, scid 0x0042, result 0x0000
[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]# [ 2937.269555] rtk_btcoex: l2cap op 6, len 16, out 0
[ 2937.269642] rtk_btcoex: RX l2cap disconn req, hndl 0x0001, dcid 0x0041, scid 0x0042
[ 2937.269653] rtk_btcoex: handle_l2cap_discon_req: handle 0x0001, dcid 0x0041, scid 0x0042, dir 0
[ 2939.823379] rtk_btcoex: l2cap op 6, len 16, out 1
[ 2939.823587] rtk_btcoex: TX l2cap disconn req, hndl 0x0001, dcid 0x0040, scid 0x0040
[ 2939.823622] rtk_btcoex: handle_l2cap_discon_req: handle 0x0001, dcid 0x0040, scid 0x0040, dir 1
[ 2942.059038] rtk_btcoex: HCI Disconnect, handle 0001, reason 0x13
[ 2942.143796] rtk_btcoex: disconn cmpl evt: status 0x00, handle 0001, reason 0x16
[ 2942.143900] rtk_btcoex: process disconn complete event.
[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]#
```

3.11.4 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 Tool. 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.

```
cat /dev/rfcomm0
```

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

The data received by the development board is as follows.

```
root@ATK-DLRK356X:/root/shell/bluetooth# cat /dev/rfcomm0
正点原子[ 1104.807434] rtk_btcoex: count_pan_packet_timeout: pan_packet_count 15
[ 1104.807563] rtk_btcoex: timeout_handler: pan idle->busy!
[ 1104.807595] rtk_btcoex: update_profile_state: btrtl_coex.profie_bitmap = 8
[ 1104.807619] rtk_btcoex: update_profile_state: btrtl_coex.profile_status = 8
[ 1104.807651] rtk_btcoex: rtk_notify_profileinfo_to_fw: BufferSize 5
[ 1104.807676] rtk_btcoex: rtk_notify_profileinfo_to_fw: NumberOfHandles 1
[ 1104.807744] rtk_btcoex: rtk_notify_profileinfo_to_fw: handle 0x0001
[ 1104.807786] rtk_btcoex: rtk_notify_profileinfo_to_fw: profile_bitmap 0x08
[ 1104.807808] rtk_btcoex: rtk_notify_profileinfo_to_fw: profile_status 0x08
[ 1104.807853] rtk_btcoex: rtk_vendor_cmd_to_fw: opcode 0xfc19
```

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

```
echo "正点原子" > /dev/rfcomm0
```

```
root@ATK-DLRK356X:/root/shell/bluetooth# echo "正点原子" > /dev/rfcomm0
root@ATK-DLRK356X:/root/shell/bluetooth# [ 1410.825165] rtk_btcoex: count_pan_packet_timeout: pan_packet_count 2
[ 1410.938557] rtk_btcoex: update_hid_active_state: handle 0x0001, interval 0
[ 1410.938664] rtk_btcoex: HID not connected, nothing to be down
```

3.12 TF(SD) Card & eMMC Test

Instruction Prompt:

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.

3.12.1 SD Card Test

After the development board starts, insert the TF card into the card slot on the bottom board of the development board, and print the following information. The development board will automatically mount the TF card. (Note: A FAT32 format TF card must be used, not an NTFS format.)

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

```
root@ATK-DLRK356X:~# [  988.800100] mmc_host mmc1: Bus speed (slot 0) = 148500000Hz (slot req 150000000Hz, actual 148500000HZ div = 0)
[  988.879499] dwmmc_rockchip fe2b0000.dwmmc: Successfully tuned phase to 106
[  988.879601] mmc1: new ultra high speed SDR104 SDHC card at address 0001
[  988.883015] mmcblk1: mmc1:0001 SD32G 29.1 GiB
[  988.885176]  mmcblk1: p1

root@ATK-DLRK356X:~# df
Filesystem     1K-blocks    Used Available Use% Mounted on
/dev/root        6127168 1639520   4214092  29% /
devtmpfs         1000424       0   1000424   0% /dev
tmpfs            1009384       0   1009384   0% /dev/shm
tmpfs            1009384    1028   1008356   1% /tmp
tmpfs            1009384     360   1009024   1% /run
/dev/mmcblk0p7    125331   12362    106419  11% /oem
/dev/mmcblk0p8  23547102      25  23547077   1% /userdata
/dev/mmcblk1p1  30556156  167440  30388716   1% /media/sdcard3
root@ATK-DLRK356X:~#
```

Read speed test is often used to test the reading speed of hard drives. The commands for this are 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 is for reference only.

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

```
root@ATK-DLRK356X:~# hdparm -t /dev/mmcblk1p1

/dev/mmcblk1p1:
Timing buffered disk reads:  188 MB in 3.02 seconds = 63737 kB/s
root@ATK-DLRK356X:~#
```

Figure 3.12-1 Read Speed Evaluation

It can be seen that the read speed is 63737 KB/s.

Write speed test

```
time dd if=/dev/zero of=/media/sdcard3/test bs=1024k count=50 conv=fdatasync
# Note that the instruction is too long, resulting in abnormal display of the instruction, but it does not affect the operation.
```

```
root@ATK-DLRK356X:~#
50+0 records in6X:~# time dd if=/dev/zero of=/media/sdcard3/test bs=1024k count=50 conv=fdatasync
50+0 records out
52428800 bytes (52 MB, 50 MiB) copied, 3.48937 s, 15.0 MB/s

real    0m3.501s
user    0m0.002s
sys     0m0.126s
root@ATK-DLRK356X:~#
```

Use the "dd" command for reading. Due to the Linux system mechanism, usually 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=/media/sdcard3/test of=/dev/null bs=1024k  # Read speed test
rm /media/sdcard3/test                               # Delete this file after reading it.
```

```
root@ATK-DLRK356X:~# echo 3 > /proc/sys/vm/drop_caches
[ 1831.926739] sh (894): drop_caches: 3
root@ATK-DLRK356X:~# time dd if=/media/sdcard3/test of=/dev/null bs=1024k
50+0 records in
50+0 records out
52428800 bytes (52 MB, 50 MiB) copied, 0.995962 s, 52.6 MB/s

real    0m1.030s
user    0m0.000s
sys     0m0.228s
root@ATK-DLRK356X:~#
root@ATK-DLRK356X:~#
root@ATK-DLRK356X:~# rm /media/sdcard3/test
```

It can be seen that the reading speed is 52.6MB/s. This speed is for reference only. The larger the file being read, the more accurate the speed will be.

3.12.2 EMMC Test

Using the df -h command to determine the partition where the EMMC is mounted. The results are as follows. In fact, the eMMC storage is divided into many partitions, and the partition mounted under the root directory "/ " is also an eMMC partition. Here, we take the partition mmcblk0p8 mounted as the directory /userdata as an example.

```
root@ATK-DLRK356X:~# df
Filesystem     1K-blocks    Used Available Use% Mounted on
/dev/root        6127168 1639520   4214092  29% /
devtmpfs         1000424       0   1000424   0% /dev
tmpfs            1009384       0   1009384   0% /dev/shm
tmpfs            1009384    1176   1008208   1% /tmp
tmpfs            1009384     352   1009032   1% /run
/dev/mmcblk0p7    125331   12362    106419  11% /oem
/dev/mmcblk0p8  23547102      25  23547077   1% /userdata
root@ATK-DLRK356X:~#
```

Read speed test

hdparm -t /dev/mmcblk0p8

```
root@ATK-DLRK356X:~# hdparm -t /dev/mmcblk0p8

/dev/mmcblk0p8:
Timing buffered disk reads:  482 MB in 3.00 seconds = 164456 kB/s
root@ATK-DLRK356X:~#
```

It can be seen that the reading speed is 164456 KB/s.

Write speed test

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

Note that the instruction is too long, resulting in abnormal display of the instruction, but it does not affect the operation.

```
root@ATK-DLRK356X:~#
50+0 records in6X:~# time dd if=/dev/zero of=/userdata/test bs=1024k count=50 conv=fdatasync
50+0 records out
52428800 bytes (52 MB, 50 MiB) copied, 1.22247 s, 42.9 MB/s

real    0m1.277s
user    0m0.004s
sys     0m0.713s
root@ATK-DLRK356X:~#
```

It can be seen that the writing speed is 42.9MB/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-DLRK356X:~# echo 3 > /proc/sys/vm/drop_caches
[ 2925.649122] sh (894): drop_caches: 3
root@ATK-DLRK356X:~# 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, 0.316789 s, 166 MB/s

real    0m0.355s
user    0m0.001s
sys     0m0.147s
root@ATK-DLRK356X:~#
```

It can be seen that the reading speed is 166MB/s. The file being read is too small and this speed is for reference only.

3.13 ADC Test

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

There is an ADC circuit marked as VR on the ATK-DLRK3568 baseboard for users to test. The potentiometer is connected to ADC3 interface, as shown in the following figure:

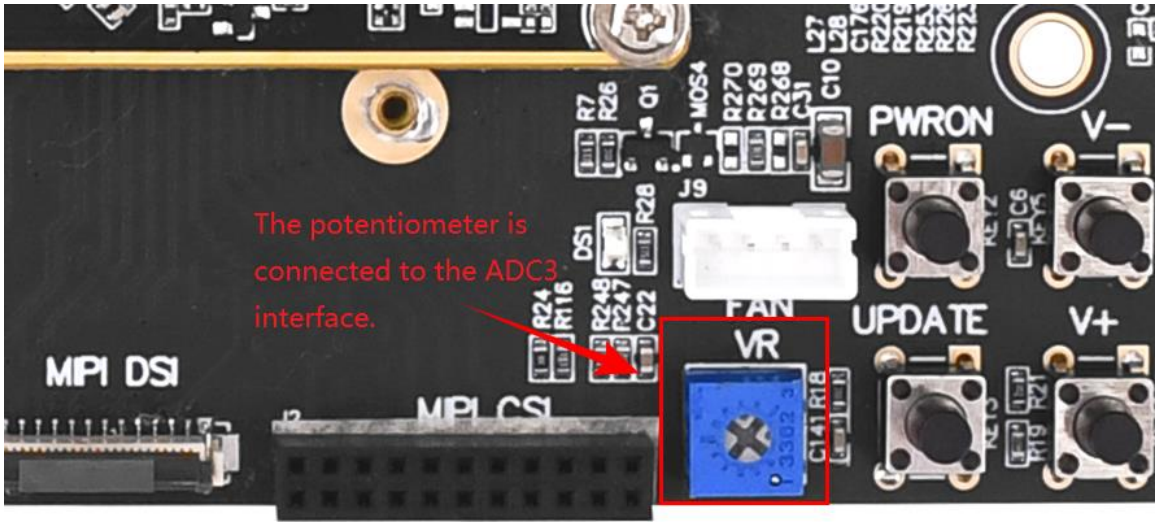


Figure 3.13-1 Potentiometer diagram

In the Linux system, ADC sampling falls under the IIO subsystem. It can be accessed through the interface provided by SYSFS and uses 10-bit precision. Use the following command to view the raw data collected by the ADC:

```
cat /sys/bus/iio/devices/iio\:device0/in_voltage3_raw
```

```
root@ATK-DLRK356X:/# cat /sys/bus/iio/devices/iio\:device0/in_voltage3_raw
979
root@ATK-DLRK356X:/#
```

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 indicates the number of bits for AD conversion, raw represents the original data collected by AD, and Vresult is the voltage that the user needs to collect.

Vresult = (1800mv * 979) /1023 = 1722.58mv

By using the formula, we can calculate that it is approximately 1.7V.

3.14 CPU temperature

To check the CPU temperature, use the CAT command. The user layer provides two interfaces, and the unit of the displayed value is millikelvin, as shown in the following figure.

```
cat /sys/class/thermal/thermal_zone0/temp    # Check the temperature of CPU1
cat /sys/class/thermal/thermal_zone1/temp    # Check the temperature of CPU2
```

```
root@ATK-DLRK356X:/# cat /sys/class/thermal/thermal_zone0/temp
43750
root@ATK-DLRK356X:/# cat /sys/class/thermal/thermal_zone1/temp
41875
root@ATK-DLRK356X:/#
```

41875 represents 41.875 degrees.

3.15 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 frequency

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

Enable the CPU performance mode and run at the highest frequency.

echo performance > /sys/devices/system/cpu/cpufreq/policy0/scaling_governor

Switch the governor to userspace

echo userspace > /sys/devices/system/cpu/cpufreq/policy0/scaling_governor

To set the CPU frequency, you need to first switch to the userspace.

echo 1992000 > /sys/devices/system/cpu/cpufreq/policy0/scaling_setspeed

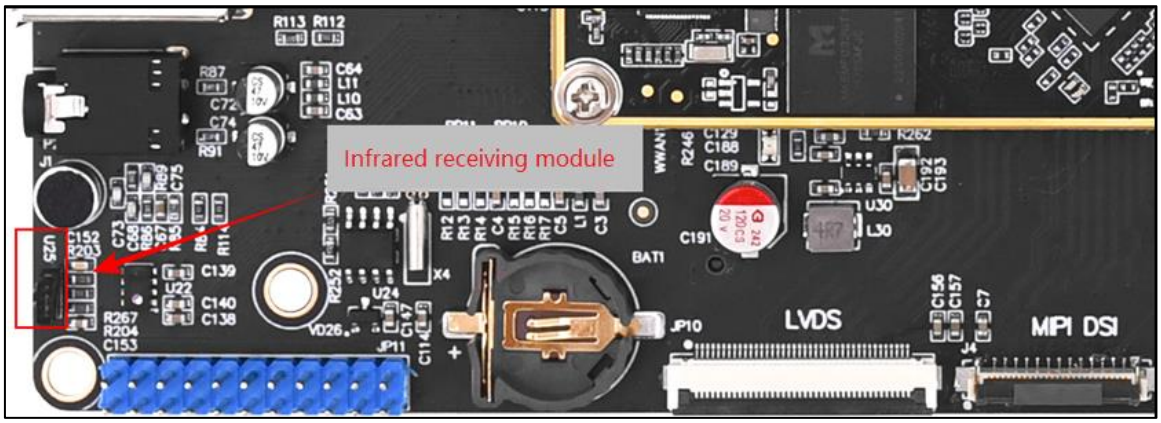
Check the current running CPU frequency again

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

```
root@ATK-DLRK356X:~# cat /sys/devices/system/cpu/cpu*/cpufreq/cpuinfo_cur_freq
1416000
1416000
1416000
1416000
root@ATK-DLRK356X:~# cat /sys/devices/system/cpu/cpu*/cpufreq/scaling_cur_freq
1416000
1416000
1416000
1416000
root@ATK-DLRK356X:~# cat /sys/devices/system/cpu/cpufreq/policy0/scaling_available_frequencies
408000 600000 816000 1104000 1416000 1608000 1800000 1992000
root@ATK-DLRK356X:~# echo performance > /sys/devices/system/cpu/cpufreq/policy0/scaling_governor
root@ATK-DLRK356X:~# echo userspace > /sys/devices/system/cpu/cpufreq/policy0/scaling_governor
root@ATK-DLRK356X:~# echo 1992000 > /sys/devices/system/cpu/cpufreq/policy0/scaling_setspeed
root@ATK-DLRK356X:~# cat /sys/devices/system/cpu/cpu*/cpufreq/cpuinfo_cur_freq
1992000
1992000
1992000
1992000
root@ATK-DLRK356X:~#
```

3.16 Infrared test

The PCB board with the onboard infrared module is marked with the position U25 on the silkscreen, as shown in the following figure.



We need to find an infrared remote control (the infrared remote control is an accessory for the non-developer board, please prepare it yourself), such as the infrared remote module from the ALIENTEK store. If you have any other infrared remote controls at hand, you can also test them. Note that the codes

of different remote controls may be different. If you use the remote control from the ALIENTEK store, then you input the number "1", and the development board will decode it as "1", and so on.

3.16.1 Testing for Multiple Infrared Remotes

If you have an infrared remote control, such as for an air conditioner, please proceed. Execute the following instructions to enable the infrared information printing. This time, we are testing with the Gree air conditioner infrared remote control and the ALIENTEK infrared remote control module.

```
echo 1 > sys/module/rockchip_pwm_remotectl/parameters/code_print
```

```
root@ATK-DLRK356X:/# echo 1 > sys/module/rockchip_pwm_remotectl/parameters/code_print
root@ATK-DLRK356X:/# [ 3674.313651] USERCODE=0x929
[ 3674.336307] RMC_GETDATA=50
[ 3674.481504] USERCODE=0x929
[ 3674.505210] RMC_GETDATA=70
[ 3678.208119] USERCODE=0xff00
[ 3678.235263] RMC_GETDATA=f7
[ 3679.150112] USERCODE=0xff00
[ 3679.177272] RMC_GETDATA=f7
[ 3679.641107] USERCODE=0xff00
[ 3679.668245] RMC_GETDATA=f7
[ 3680.539443] USERCODE=0xff00
[ 3680.566584] RMC_GETDATA=e3
[ 3682.080559] USERCODE=0xff00
[ 3682.107688] RMC_GETDATA=a5
```

It can be seen that without decoding, we have no idea what the data on the above remote control is.

3.16.2 Applicable to ALIENTEK Infrared Remote Control Module

As shown in the figure below, this module is an infrared remote control for ALIENTEK stores, commonly used on STM32 development boards. Our ATK-DLRK3568 development board has already converted its infrared encoding and decoding into corresponding keys or function keys. For example, inputting the number "1" can be recognized as the number "1".

The test program is located at /userdata/apps/src/pcba/infrared_test. The path of this test program is the development board CD-ROM A disc - Basic Data -> 01_codes -> Peripheral Test Source Code.

Now we will execute the following instructions to test this remote control.

First, turn off the printing of information. Then execute the test program.

echo 0 > sys/module/rockchip_pwm_remotectl/parameters/code_print

/userdata/apps/src/pcba/infrared_test

```
root@ATK-DLRK356X:/# echo 0 > sys/module/rockchip_pwm_remotectl/parameters/code_print
root@ATK-DLRK356X:/# /userdata/apps/src/pcba/infrared_test
===  等待红外遥控按键数字0~9按下  ===
KEY_1: press
KEY_1: release
KEY_2: press
KEY_2: release
KEY_3: press
KEY_3: release
KEY_4: press
KEY_4: release
```

It can be seen that the corresponding keys have been decoded into the corresponding values.

3.17 Headphone Insertion/Extraction Test

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

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

```
root@ATK-DLRK356X:/# evtest
No device specified, trying to scan all of /dev/input/event*
Available devices:
/dev/input/event0:      fe6e0030.pwm
/dev/input/event1:      rk805 pwrkey
/dev/input/event2:      goodix-ts
/dev/input/event3:      lightsensor-level
/dev/input/event4:      proximity
/dev/input/event5:      gsensor
/dev/input/event6:      gyro
/dev/input/event7:      adc-keys
/dev/input/event8:      rockchip,rk809-codec Headphones
/dev/input/event9:      rockchip,hdmi rockchip,hdmi
Select the device event number [0-9]: 8
Input driver version is 1.0.1
Input device ID: bus 0x0 vendor 0x0 product 0x0 version 0x0
Input device name: "rockchip,rk809-codec Headphones"
Supported events:
  Event type 0 (EV_SYN)
  Event type 5 (EV_SW)
    Event code 2 (SW_HEADPHONE_INSERT) state 0
Properties:
Testing ... (interrupt to exit)
[ 5175.365792] asoc-simple-card rk809-sound: ASoC: DAPM unknown pin Headphones
Event: time 1686278329.338309, type 5 (EV_SW), code 2 (SW_HEADPHONE_INSERT), value 1
Event: time 1686278329.338309, -------------- SYN_REPORT ------------
[ 5176.859105] asoc-simple-card rk809-sound: ASoC: DAPM unknown pin Headphones
Event: time 1686278330.831620, type 5 (EV_SW), code 2 (SW_HEADPHONE_INSERT), value 0
Event: time 1686278330.831620, -------------- SYN_REPORT ------------
```

Input the number 8

Alternatively, we can also use a program to detect the insertion and removal of the headphones. The testing program is located at /userdata/apps/src/pcba/hp_det_test. The source code path of this testing program is the development board CD-ROM A disk - Basic Data→01_ codes→Peripheral Test Source Code.

Execute the following command to conduct the headphone insertion and removal test.

```
root@ATK-DLRK356X:/# /userdata/apps/src/pcba/hp_det_test
=== 耳机插拔测试 ===
[ 5946.766010] asoc-simple-card rk809-sound: ASoC: DAPM unknown pin Headphones
耳机插入
[ 5948.552609] asoc-simple-card rk809-sound: ASoC: DAPM unknown pin Headphones
耳机拔出
```

3.18 Light sensor AP3216C test

AP3216C introduction:

An integrated environmental sensor, AP3216C, was connected via I2C5 on the ATK-DLRK3568 development board. This sensor is produced by Dunan and is capable of detecting three environmental parameters: ambient light intensity (ALS), proximity distance (PS), and infrared intensity (IR). The features of AP3216C are as follows:

①, The I2C interface allows a baud rate of up to 400Kbit/S in fast mode.

②, Multiple working modes are available: ALS, PS + IR, ALS + PS + IR, PD, etc.

③, Built-in temperature compensation circuit.

④, Wide operating temperature range (-30°C to +80°C).

⑤, Ultra-small package, 4.1mm x 2.4mm x 1.35mm.

⑥, The ambient light sensor has a 16-bit resolution.

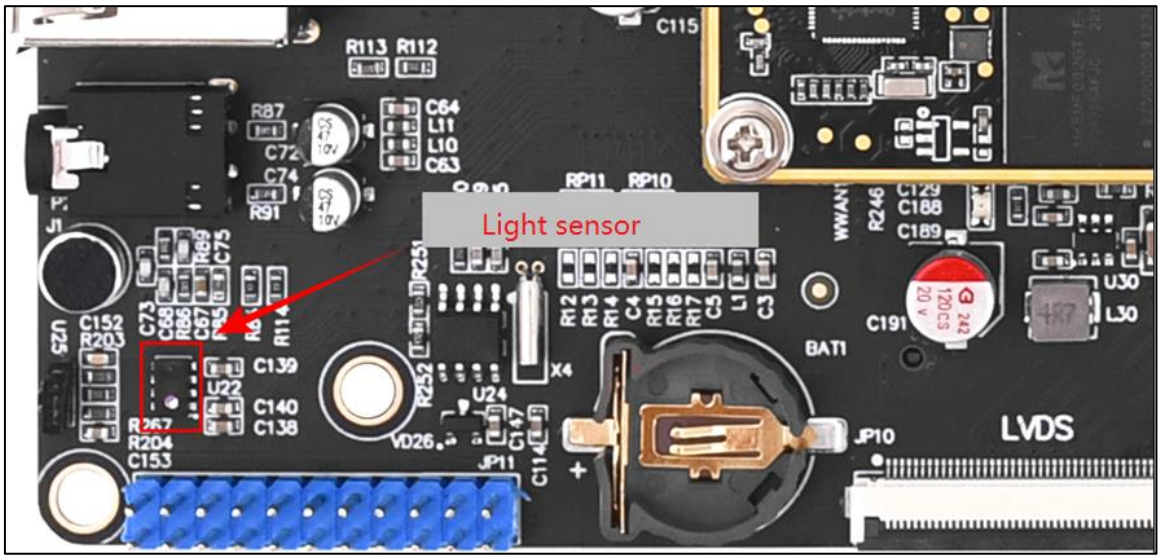
⑦, The proximity sensor and infrared sensor have a 10-bit resolution.

The AP3216C is commonly used in mobile phones, tablets, navigation devices, etc. Its built-in proximity sensor can be used to detect whether an object is approaching, such as detecting whether the ear is touching the earpiece on a mobile phone. If it detects that the ear is touching, it indicates that a call is in progress, and the phone will turn off the screen to save power. The ambient light sensor can also be used to detect the light intensity, enabling automatic backlight brightness adjustment.

To read the environmental sensor's environmental parameter values, execute the following instructions in the development board file system. Depending on the environment of the development board, the environmental parameter values will vary. First, use the following instructions to read the environmental parameter values once, and then bring the hand close to the AP3216C sensor.

ALS & PS are commonly used on the ALIENTEK 1000 ATK-DLRK3568 development board to measure the light environment intensity and the proximity distance of objects.

The position of the sensor is as shown in the following figure.



The test program is located at /userdata/apps/src/pcba/ap3216c_test. This test program path is on the A disk of the development board CD-ROM - Basic Data -> 01_codes -> Peripheral Test Source Code.

Execute the following command to test whether the AP3216C is working properly. Bring your finger close to the sensor and it will print the information as shown in the figure below.

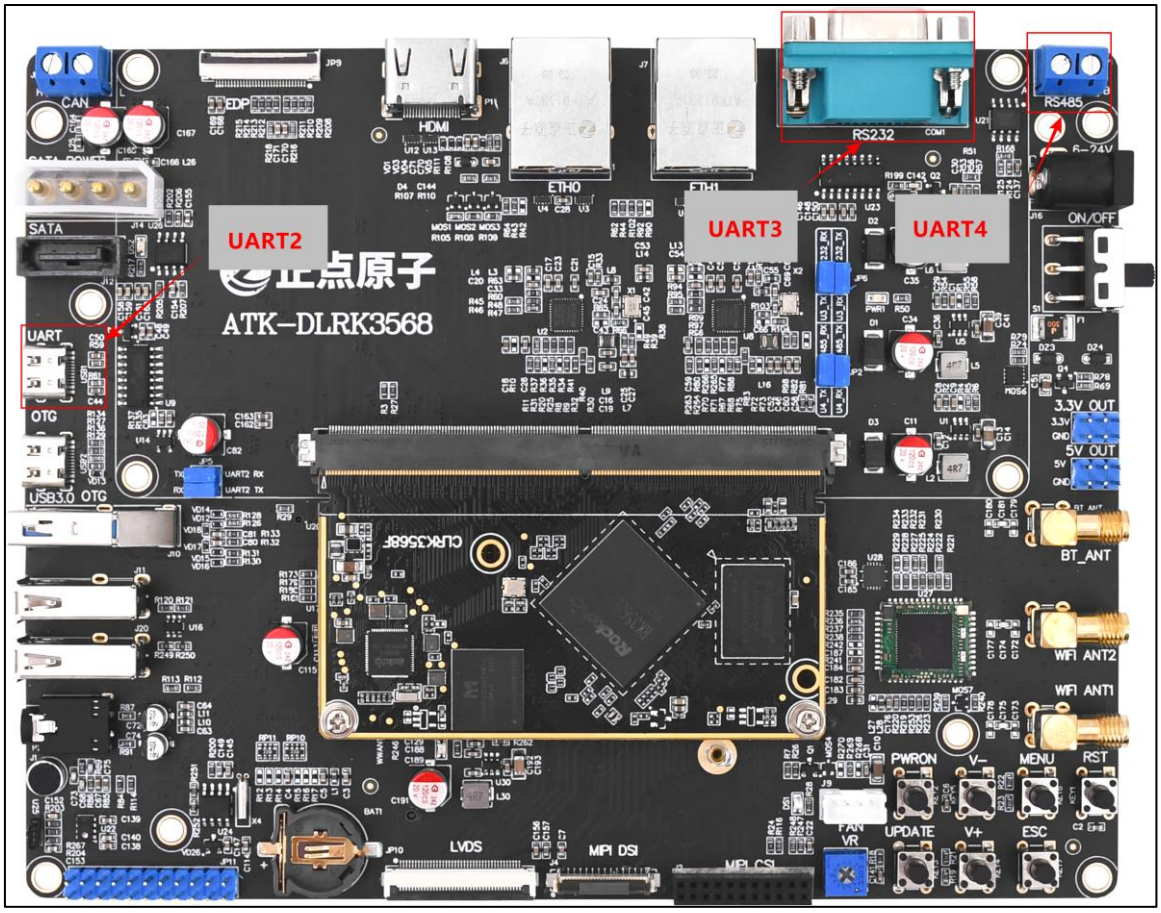
/userdata/apps/src/pcba/ap3216c_test

```
root@ATK-DLRK356X:/# /userdata/apps/src/pcba/ap3216c_test
===  ap3216c testing  ===[  747.952608] light_ap321xx 5-001e: sensor on: starting poll sensor data 30ms
[  747.954713] proximity_ap321xx 5-001e-1: sensor on: starting poll sens光强度等级0～7：当前为1
or data 30ms
距离状态：接近
光强度等级0～7：当前为0
光强度等级0～7：当前为1
光强度等级0～7：当前为0
光强度等级0～7：当前为1
距离状态：远离
光强度等级0～7：当前为0
距离状态：接近
光强度等级0～7：当前为1
距离状态：远离
```

3.19 Serial port test

The ALIENTEK ATK-DLRK3568 development board introduces three serial ports. Of course, some of these ports are multiplexed functions. The maximum number of serial ports supported by the RK3568 is 10.

The serial ports onboard the development board are as shown in the following figure (excluding the multiplexed (the exposed IO pin) or used serial ports such as the one used by Bluetooth for UART8).



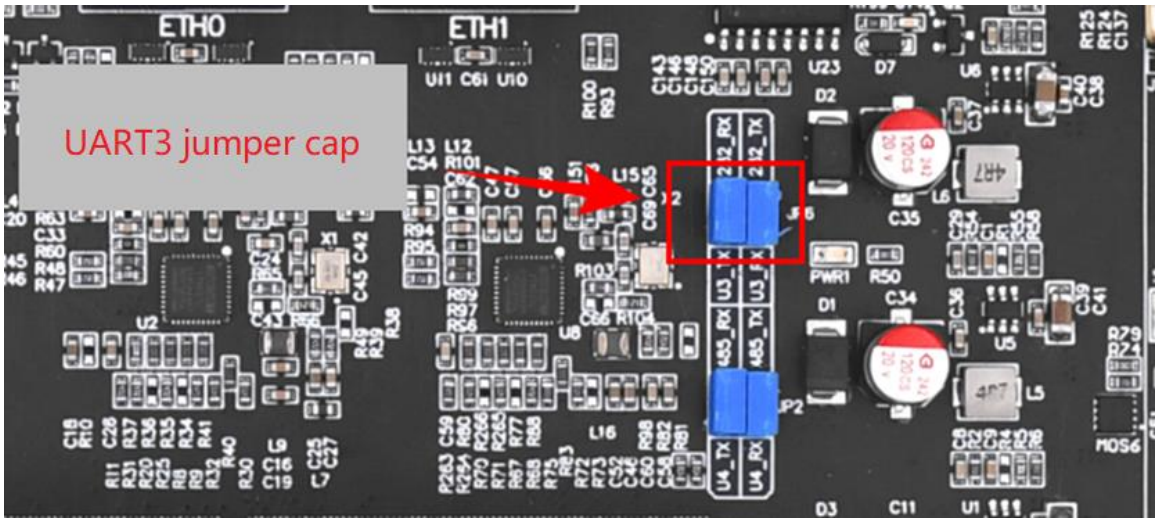
3.19.1 UART2 Test

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 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 prepare it yourself), and it should be of the male head type (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.



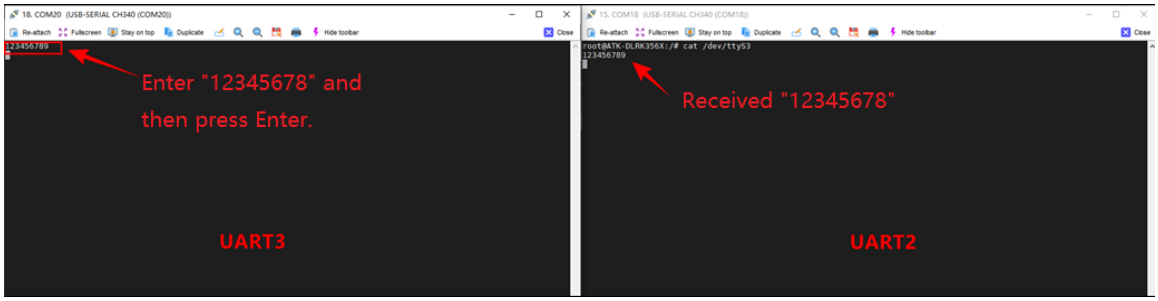
Set the serial port baud rate, connect the USB to RS232 serial cable to the development board, then connect it to the USB interface of the computer, and open the upper-level machine (Mobaxterm).

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

```
root@ATK-DLRK356X:/# stty -F /dev/ttyS3 115200
[13604.357135] of_dma_request_slave_channel: dma-names property of node '/serial@fe670000' missing or empty
[13604.357215] ttyS3 - failed to request DMA, use interrupt mode
root@ATK-DLRK356X:/#
```

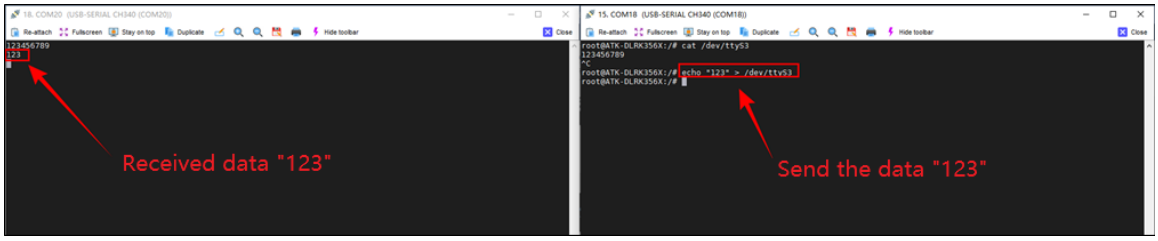
After configuring the same baud rate of 115200 in the upper computer Mobaxterm and selecting the corresponding COM port, as shown in the following figure. Here, UART3 is the serial port we need to test, and UART2 is the debugging serial port. The debugging serial port UART2 requires the input of the 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 data is received, it indicates that both the receiving and transmitting functions of UART3 are normal.

3.19.3 UART4 Test

Similarly, the test steps for UART4 are the same as those in the previous subsection 3.19.2. At this point, 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.

The 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.

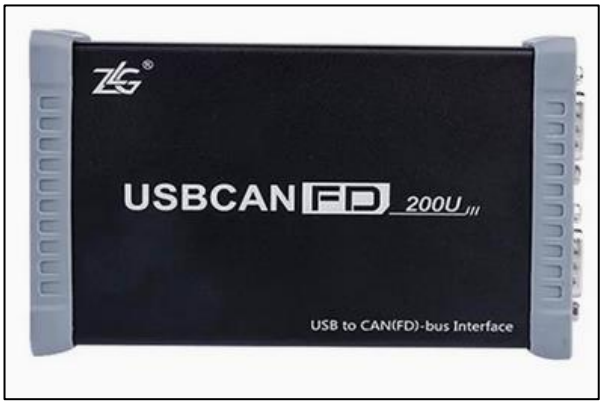
3.20 CANFD Test

CAN FD (Controller Area Network Flexible Data Rate) is a new CAN bus standard. It has made significant improvements in transmission rate and data frame length compared to the traditional CAN protocol, and can better meet the demand for transmitting large amounts of data in modern automotive and industrial applications. Compared to the traditional CAN protocol, CAN FD can provide higher data transmission rate and larger data frame length, while maintaining backward compatibility with the traditional CAN protocol, making it easy to integrate with existing CAN networks.

Testing CANFD requires the user to have a USB CANFD analyzer, such as Zhou Liguang's CANFD analyzer, etc. If the user is involved in CAN-related work, then this user should have a CANFD analyzer. If you are a beginner, you need to purchase a CANFD analyzer and check if it supports CANFD, because there is also ordinary CAN. To test CANFD, you need to purchase a CANFD analyzer. Or you can also test by connecting the CANFD interfaces of two development boards. If you have the conditions or requirements, you can purchase a CANFD analyzer. However, USB CANFD analyzers are often expensive, more expensive than a complete board set.

Note that CAN is half-duplex and cannot transmit and receive simultaneously.

This experiment uses Zhou Liguang's USB CANFD analyzer, as shown in the figure below. Note that for the use of CANFD analyzer, please consult the merchant. This experiment does not explain the use of USB CANFD analyzer.



The location of the CANFD 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 CANFD.

```
ip link set can0 up type can bitrate 1000000 dbitrate 5000000 fd on
```

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 arbitration segment communication rate of the CAN bus to 1 Mbps (1000000 bits/sec).
4. dbitrate 5000000: Set the data segment communication rate of CAN FD to 5 Mbps (5000000 bits/sec).
5. fd on: Enable CAN FD protocol, which means enabling the Flexible Data Rate (FDR) function of the CAN bus. Note that this option is only available on hardware that supports CAN FD.

```
root@ATK-DLRK356X:/#
[   49.998628] rockchip_canfd fe580000.can can0: bitrate error 0.3%bitrate 5000000 fd on
[   49.998731] rockchip_canfd fe580000.can can0: bitrate error 1.0%
[   49.999763] IPv6: ADDRCONF(NETDEV_UP): can0: link is not ready
[   50.000042] IPv6: ADDRCONF(NETDEV_CHANGE): can0: link becomes ready
root@ATK-DLRK356X:/#
```

Execute the following instruction to print the information of CAN.

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

```
root@ATK-DLRK356X:/# ip -details link show can0
2: can0: <NOARP,UP,LOWER_UP,ECHO> mtu 72 qdisc pfifo_fast state UP mode DEFAULT group default qlen 10
    link/can  promiscuity 0 minmtu 0 maxmtu 0
    can <FD> state ERROR-ACTIVE (berr-counter tx 0 rx 0) restart-ms 0
          bitrate 1003378 sample-point 0.743
          tq 13 prop-seg 27 phase-seg1 27 phase-seg2 19 sjw 1 brp 2
          rockchip_canfd: tseg1 1..128 tseg2 1..128 sjw 1..128 brp 1..256 brp_inc 2
          dbitrate 4950000 dsample-point 0.733
          dtq 13 dprop-seg 5 dphase-seg1 5 dphase-seg2 4 dsjw 1 dbrp 2
          rockchip_canfd: dtseg1 1..32 dtseg2 1..16 dsjw 1..16 dbrp 1..256 dbrp_inc 2
          clock 148500000 numtxqueues 1 numrxqueues 1 gso_max_size 65536 gso_max_segs 65535
root@ATK-DLRK356X:/#
```

CAN FD transmission:

Transmitting (standard frame, data frame, ID: 123, data: DEADBEEF):

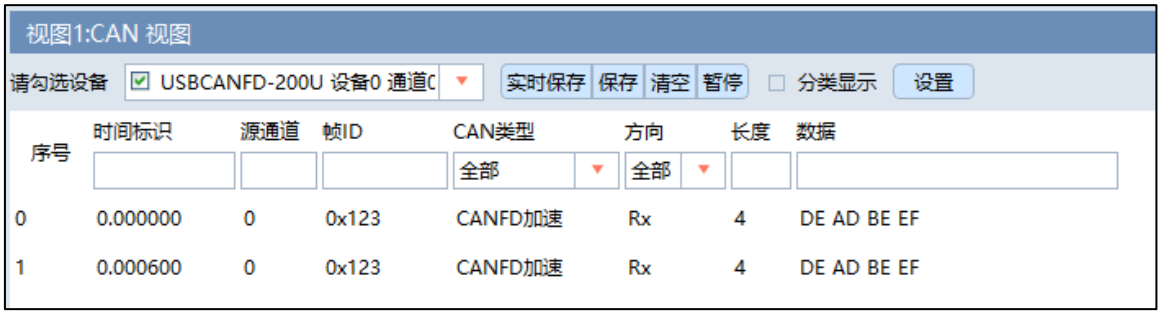
```
cansend can0 123##1DEADBEEF
```

Send (Extended Frame, Data Frame, ID: 00000123, data: DEADBEEF):

```
cansend can0 00000123##1DEADBEEF
```

```
root@ATK-DLRK356X:/# cansend can0 123##1DEADBEEF
root@ATK-DLRK356X:/# cansend can0 00000123##1DEADBEEF
root@ATK-DLRK356X:/#
```

The upper computer on CAN received the following data.



CANFD receives data. Input the following command to wait for reception:

```
candump can0
```

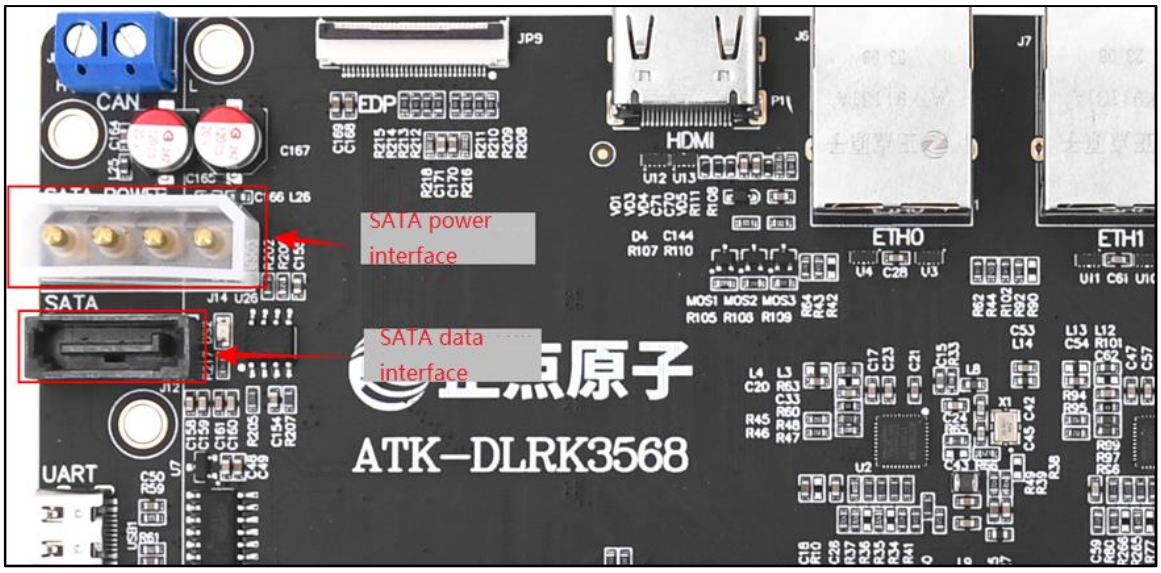
The data sent by the CAN host computer is shown in the following figure. This time, we sent 64 bytes.

The development board received 64 bytes as shown in the figure below.

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.



If you have a MIPI screen, you need to remove the screen. The interface is located below the screen. 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, but to protect your hard drive, it is recommended to turn off the power first, then plug it in, and then turn on the computer.

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

```
df
```

As shown in the following figure, SATA hard drives will be mounted under the "/media/" directory. For example, the SATA-mounted directory on my computer is "/media/usdisk0".

```
root@ATK-DLRK356X:/# df
Filesystem     1K-blocks    Used Available Use% Mounted on
/dev/root        6127168  523324   5329664   9% /
devtmpfs         1989880       0   1989880   0% /dev
tmpfs            1998840       0   1998840   0% /dev/shm
tmpfs            1998840     916   1997924   1% /tmp
tmpfs            1998840     340   1998500   1% /run
/dev/sda1      488267264      64 488267200   1% /media/udisk0
/dev/mmcblk0p8  53803957 3473352  48570476   7% /userdata
/dev/mmcblk0p7    129861   16892    108932  14% /oem
root@ATK-DLRK356X:/#
```

3.21.1 Partition the hard disk

Use the fdisk command to view the nodes of the hard disk.

fdisk -l // "-l" stands for -L, not the number "1".

```
root@ATK-DLRK356X:/# fdisk -l
Disk /dev/sda: 466 GB, 500107862016 bytes, 976773168 sectors
60801 cylinders, 255 heads, 63 sectors/track
Units: sectors of 1 * 512 = 512 bytes

Device  Boot StartCHS    EndCHS        StartLBA     EndLBA    Sectors  Size Id Type
/dev/sda1    0,1,1       1023,254,63          63  976773167  976773105  465G  c Win95 FAT32 (LBA)
Found valid GPT with protective MBR; using GPT

Disk /dev/mmcblk0: 120832000 sectors, 1656M
Logical sector size: 512
Disk identifier (GUID): ab58e823-b72e-46a9-d1d0-08f80216cd8d
Partition table holds up to 128 entries
First usable sector is 34, last usable sector is 120831966

Number  Start (sector)    End (sector)  Size Name
     1           16384           24575 4096K uboot
     2           24576           32767 4096K misc
     3           32768          163839 64.0M boot
     4          163840          294911 64.0M recovery
     5          294912          360447 32.0M backup
     6          360448        12943359 6144M rootfs
     7        12943360        13205503  128M oem
     8        13205504       120831935 51.3G userdata
root@ATK-DLRK356X:/#
```

A 500GB hard drive

As can be seen from the above picture, the SATA-mounted node is /dev/sda. Now, let's format this hard drive. If your hard drive already has at least two partitions, then there will be sda1 and sda2. What's the difference between sda and sda1? sda is the node of the hard drive, while sda is the first partition of the hard drive. In this case, the author's hard drive has two partitions. If there are other partitions, then there will be sda3, sda4, etc., and so on, following the same pattern.

Since this hard drive of the author is already in the "vfat" format, it indicates that it can be recognized and mounted by Linux. Hard drives that have been mounted need to be unmounted before they can be formatted.

Therefore, the author needs to unmount the mounted hard drive. If your hard drive is not mounted, there is no need to unmount it.

To unmount the hard drive, execute the following command.

umount /dev/sda*

```
root@ATK-DLRK356X:/# umount /dev/sda*
root@ATK-DLRK356X:/#
```

Now we will partition this sda, and execute the following command.

fdisk /dev/sda

```
root@ATK-DLRK356X:/# fdisk /dev/sda

The number of cylinders for this disk is set to 60801.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help):
```

Then you can input 'm' to view the usage instructions.

```
root@ATK-DLRK356X:/# fdisk /dev/sda

The number of cylinders for this disk is set to 60801.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): m        <- Input 'm'
Command Action
a       toggle a bootable flag
b       edit bsd disklabel
c       toggle the dos compatibility flag
d       delete a partition
l       list known partition types
n       add a new partition
o       create a new empty DOS partition table
p       print the partition table
q       quit without saving changes
s       create a new empty Sun disklabel
t       change a partition's system id
u       change display/entry units
v       verify the partition table
w       write table to disk and exit
x       extra functionality (experts only)

Command (m for help):
```

The commonly used options are as follows:

-l: List the partition table information of all disks;

-n: Create a new partition;

-d: Delete a partition;

-t: Modify the type of a partition;

-p: Print the partition table;

-u: Display and operate the partition table in sector units instead of cylinder units;

-m: Use the DOS-compatible mode.

-w: Save the modifications

We input 'n' to create a new partition, then input 'p' and press enter. Since the partition table format of this hard disk is MBR format, the maximum number of primary partitions supported is 4. If we input 'e', it means expanding the partition, and it can be an unlimited number of logical partitions. The default starting sector is 63. This time, we set the size of the first partition to 1GB.

```
Command (m for help): n        <- Create a new partition
Partition type
   p   primary partition (1-4)
   e   extended
p                              <- Enter 'P', primary partition
Partition number (1-4): 1
First sector (63-976773167, default 63):   <- Enter key, the starting sector is 63, default
Using default value 63
Last sector or +size{,K,M,G,T} (63-976773167, default 976773167): +1G   <- +1G indicates the position up to 1GB.

Command (m for help):
```

Now let's move on to the second partition, which is also the main partition. Refer to the steps shown in the following diagram. This time, we have set the starting position as "2097215". Since this position is exactly the end of the previous partition. The last sector serves as the termination position. That is to say, the maximum capacity of the hard drive minus 1GB is used as the size of the second partition. Don't forget to input 'w' to save the modifications. Otherwise, they won't take effect. Of course, if you continue partitioning, according to your own needs, the main partition can only be divided into 4, and the extended partition can have countless ones. Then save the modifications as usual.

```
Command (m for help): n
Partition type
   p   primary partition (1-4)
   e   extended
p
Partition number (1-4): 2
First sector (2097215-976773167, default 2097215):
Using default value 2097215
Last sector or +size{,K,M,G,T} (2097215-976773167, default 976773167):
Using default value 976773167

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table
[  972.152381]  sda: sda1 sda2
root@ATK-DLRK356X:/# [  972.161263]  sda: sda1 sda2
[  972.714547] FAT-fs (sda2): utf8 is not a recommended IO charset for FAT filesystems, filesystem will be case sensitive!
[  972.715036] FAT-fs (sda2): bogus number of reserved sectors
[  972.715062] FAT-fs (sda2): Can't find a valid FAT filesystem

root@ATK-DLRK356X:/#
```

Enter 'n' to continue creating a new partition.
Enter 'p' to continue to set it as a primary partition.
Now it's the second section. Please input '2'.
Enter
Enter
Input 'w' to save the modifications

OK, we have completed the partitioning. Now we need to check if the partitions were successfully created. As you can see in the picture, there are two partitions, sda1 and sda2, indicating that the partitions were successful.

```
ls /dev/sda*
```

```
root@ATK-DLRK356X:/# ls /dev/sda*
/dev/sda  /dev/sda1  /dev/sda2
root@ATK-DLRK356X:/#
```

3.21.2 Format the partition

After partitioning the hard disk, formatting is required. Now we will start formatting the sda partition. We are using the vfat format, which is an extension of the fat format. Here, you can choose the ext4 format, but this format cannot be recognized in Windows.

Of course, you can format it as the ntfs format. The file system supports this. Note that not all Linux file systems support NTFS. Although it can support mounting NTFS file system types, it is not supported in the Linux kernel. In simple terms, the support for NTFS is software-level adaptation, not kernel-level!

```
mkfs.ntfs /dev/sda1 -Q      // Quickly format to NTFS format
mkfs.vfat /dev/sda1
mkfs.ext4 /dev/sda1
```

This time, it will be formatted as vfat format. This way, Windows can access it and Linux can also recognize it. Format the two primary partitions that were separated earlier. If you have divided the disk into multiple partitions, the principle is the same.

```
mkfs.vfat /dev/sda1
mkfs.vfat /dev/sda2
```

```
root@ATK-DLRK356X:/# mkfs.vfat /dev/sda1
mkfs.fat 4.1 (2017-01-24)
root@ATK-DLRK356X:/# mkfs.vfat /dev/sda2
mkfs.fat 4.1 (2017-01-24)
root@ATK-DLRK356X:/#
```

The formatting was completed and the development board was restarted. After the restart, the hard drive has automatically been mounted under /media/udisk*. By re-entering the df command, it can be seen that both partitions have been successfully mounted.

df

```
root@ATK-DLRK356X:/# df
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/root         6127168  523324   5329664   9% /
devtmpfs          1989880       0   1989880   0% /dev
tmpfs             1998840       0   1998840   0% /dev/shm
tmpfs             1998840     920   1997920   1% /tmp
tmpfs             1998840     344   1998496   1% /run
/dev/mmcblk0p8   53803957 3473352  48570476   7% /userdata
/dev/mmcblk0p7     129861   16892    108932  14% /oem
/dev/sda1         1046512       4   1046508   1% /media/udisk0
/dev/sda2       487218944      32 487218912   1% /media/udisk1
root@ATK-DLRK356X:/#
```

Use the blkid command to check the partition disk format. You can see that it is in "vfat" format. Note that if the format is "ntfs", it cannot be seen here because Linux cannot recognize this type.

blkid

```
root@ATK-DLRK356X:/# blkid
/dev/mmcblk0p8: LABEL="userdata" UUID="3efafb17-b53d-4ddb-baf0-44ef9e0eec6a" TYPE="ext2"
/dev/mmcblk0p7: LABEL="oem" UUID="3ee28973-6b3c-4deb-b60f-279320952b48" TYPE="ext2"
/dev/mmcblk0p6: LABEL="rootfs" UUID="e91d8978-a1d9-4a05-b886-87b290052277" TYPE="ext4"
/dev/sda2: UUID="1BFD-2A96" TYPE="vfat"
/dev/sda1: UUID="1BBD-D795" TYPE="vfat"
/dev/sda: UUID="7AB7-7795" TYPE="vfat"
root@ATK-DLRK356X:/#
```

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-RK3568 M.2 interface is on the back of the development board. Please install the M.2 solid-state drive on the back of the development board (please bring your own M.2 screws, and most users who purchase M.2 hard drives will be provided by the merchant), then power on and start.

After powering on and starting, the M.2 hard drive will automatically mount. Use the df -h command to check the node where the M.2 hard drive is mounted.

df -h

```
root@ATK-DLRK356X:/# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       5.9G  502M  5.1G   9% /
devtmpfs        1.9G     0  1.9G   0% /dev
tmpfs           2.0G     0  2.0G   0% /dev/shm
tmpfs           2.0G  1.1M  2.0G   1% /tmp
tmpfs           2.0G  344K  2.0G   1% /run
/dev/nvme0n1p2  238G  5.7G  223G   3% /media/udisk0
/dev/nvme0n1p1  511M   45M  467M   9% /media/udisk1
/dev/mmcblk0p7  127M   17M  107M  14% /oem
/dev/mmcblk0p8   52G  3.4G   47G   7% /userdata
root@ATK-DLRK356X:/#
```

It can be seen that the M.2 solid-state drive has been mounted under the /media/udisk* path. If you need to format and partition it, please refer to Section 3.21. Now you can start reading and writing files in the /media/udisk* path.

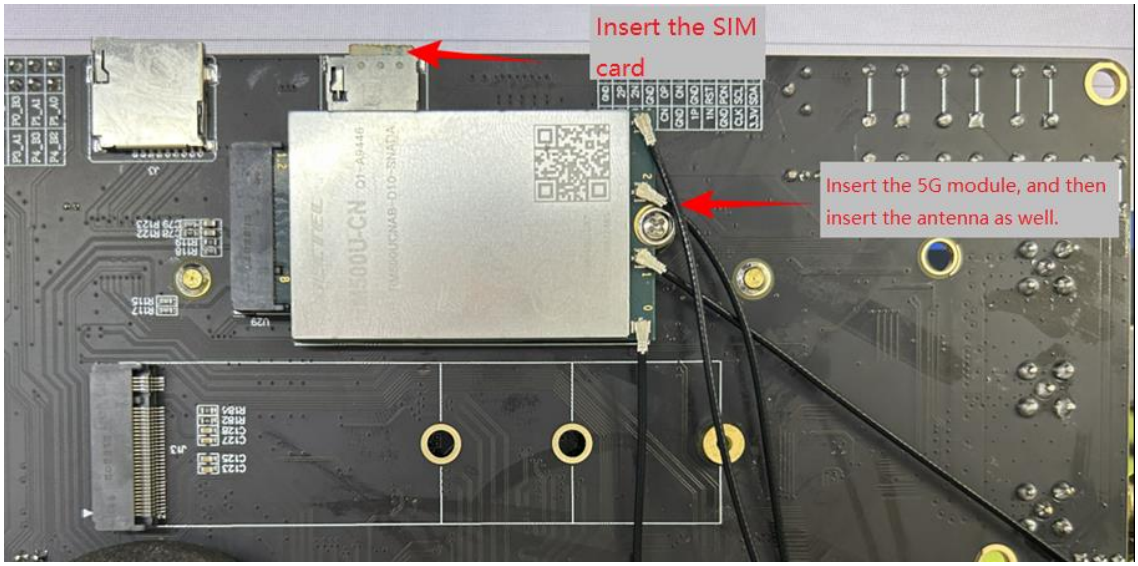
3.23 5G Module Interface Test

3.23.1 5G Module Internet Access Test

Please prepare the RM500U-CN Quectel 5G module and connect an antenna. If necessary, contact Ruiwei for purchase. As shown in the figure below. It is already supported on the ALIENTEK ATK-DLRK3568 development board. Please note that it must be this specific type of 5G module! The 5G module interface is on the back of the development board, and the SIM card interface is also on the back.

The 5G module and antenna are as shown in the following picture.

Please turn off the power supply of the development board, then insert the 5G module and the SIM card (supporting all network connections). As shown in the picture, use a regular SIM card and insert it into the SIM card slot. To fix the 5G module, you need to use the Quectel RM500U-CN 5G module.



After powering on, the terminal prints the following message, indicating that the 5G module driver has been successfully loaded.

```
root@ATK-DLRK356X:/# [   18.655445] usb 6-1: new SuperSpeed Gen 1 USB device number 2 using xhci-hcd
[   18.674810] usb 6-1: New USB device found, idVendor=2c7c, idProduct=0900, bcdDevice= 4.04
[   18.675021] usb 6-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[   18.675056] usb 6-1: Product: RM500U-CN
[   18.675084] usb 6-1: Manufacturer: Quectel
[   18.675188] usb 6-1: SerialNumber: 0123456789ABCDEF
[   18.765638] cdc_ncm 6-1:1.0: MAC-Address: 5e:f2:26:c9:4e:e7
[   18.766795] cdc_ncm 6-1:1.0 usb0: register 'cdc_ncm' at usb-xhci-hcd.0.auto-1, CDC NCM, 5e:f2:26:c9:4e:e7
[   18.768890] IPv6: ADDRCONF(NETDEV_UP): usb0: link is not ready
[   18.768948] usb 6-1: idVendor=2c7c, idProduct=900, bInterfaceNumber=2
[   18.768971] option 6-1:1.2: GSM modem (1-port) converter detected
[   18.770098] usb 6-1: GSM modem (1-port) converter now attached to ttyUSB0
[   18.770792] usb 6-1: idVendor=2c7c, idProduct=900, bInterfaceNumber=3
[   18.770824] option 6-1:1.3: GSM modem (1-port) converter detected
[   18.771719] usb 6-1: GSM modem (1-port) converter now attached to ttyUSB1
[   18.772373] usb 6-1: idVendor=2c7c, idProduct=900, bInterfaceNumber=4
[   18.772408] option 6-1:1.4: GSM modem (1-port) converter detected
[   18.773388] usb 6-1: GSM modem (1-port) converter now attached to ttyUSB2
[   18.773961] usb 6-1: idVendor=2c7c, idProduct=900, bInterfaceNumber=5
[   18.773986] option 6-1:1.5: GSM modem (1-port) converter detected
[   18.774702] cdc_ncm 6-1:1.0 usb0: 851 mbit/s downlink 851 mbit/s uplink
[   18.774979] usb 6-1: GSM modem (1-port) converter now attached to ttyUSB3
[   18.775624] usb 6-1: idVendor=2c7c, idProduct=900, bInterfaceNumber=6
[   18.775659] option 6-1:1.6: GSM modem (1-port) converter detected
[   18.776535] usb 6-1: GSM modem (1-port) converter now attached to ttyUSB4

root@ATK-DLRK356X:/#
```

To view the module ID, use the "lsusb" command.

```
root@ATK-DLRK356X:/# lsusb
Bus 005 Device 001: ID 1d6b:0002
Bus 003 Device 001: ID 1d6b:0001
Bus 001 Device 001: ID 1d6b:0002
Bus 006 Device 001: ID 1d6b:0003
Bus 004 Device 001: ID 1d6b:0001
Bus 002 Device 001: ID 1d6b:0002
Bus 006 Device 002: ID 2c7c:0900
root@ATK-DLRK356X:/#
```

Now, we only need to follow a few simple steps to access the internet using the 5G module. Execute the following command in the command line. Here, ttyUSB2 is the interface provided by the driver, which can be used to issue AT commands to configure the module. If there is no ttyUSB2 node, it indicates that the module has not been initialized.

```
microcom /dev/ttyUSB2 -s 115200
```

```
root@ATK-DLRK356X:/# microcom /dev/ttyUSB2 -s 115200
```

3.23.1.1 Configuration of driver type

There are several supported driver types. We only need to configure one of them. For detailed information, please refer to the relevant documentation. The configuration will be recorded in the module and does not need to be reconfigured next time.

AT+QCFG="usbnet",5 Configures the dialing method and driver type of the network card as NCM;

AT+QCFG="usbnet",1 Configures the dialing method and driver type of the network card as ECM;

AT+QCFG="usbnet",2 Configures the dialing method and driver type of the network card as MBIM;

AT+QCFG="usbnet",3 Configures the dialing method and driver type of the network card as RNDIS.

We can choose the first one, that is, configure it as AT+QCFG="usbnet",5 for the driver type.

The configuration takes effect after a reboot. It is recommended to query the current configuration by entering AT+QCFG="usbnet" when starting up; if the current configuration is different from the expected one, you can configure it using the above method and restart the module.

Enter the AT command to configure the driver type, press Enter. It may return a failure result. We can configure several times until it succeeds.

AT+QCFG="usbnet",5

```
root@ATK-DLRK356X:/# microcom /dev/ttyUSB2 -s 115200
AT+QCFG="usbnet",5
+CME ERROR: 4
AT+QCFG="usbnet",5
+CME ERROR: 4
AT+QCFG="usbnet",5
OK
```

Seeing "OK" indicates that the configuration is successful.

Enter the "AT" command to query the driver type, press Enter, and you can see that it is "5" which is the one we configured, and it is the NCM driver type.

AT+QCFG="usbnet"

```
root@ATK-DLRK356X:/# microcom /dev/ttyUSB2 -s 115200
AT+QCFG="usbnet",5
+CME ERROR: 4
AT+QCFG="usbnet",5
+CME ERROR: 4
AT+QCFG="usbnet",5
OK
AT+QCFG="usbnet"
+QCFG: "usbnet",5

OK
```

View the current configuration of the driver type

3.23.1.2 Configure dial-up mode

There are several dialing modes available, and we need to configure it as the network card mode. The configuration will be recorded in the module and won't need to be reconfigured next time.

AT+QCFG="nat",0 Configures the dialing mode to network card mode;

AT+QCFG="nat",1 Configures the dialing mode to routing mode;

AT+QCFG="nat",2 Configures the dialing mode to bridge mode.

The parameter configuration takes effect after a restart. It is recommended to query the current configuration by entering AT+QCFG="nat" when the device is powered on; if the current configuration is different from the expected one, you can configure it using the above method and restart the module.

Enter the AT command to configure the dialing mode, press Enter. It may return a failure result. We can configure several times until it succeeds.

AT+QCFG="nat",0

```
root@ATK-DLRK356X:/# microcom /dev/ttyUSB2 -s 115200
AT+QCFG="usbnet",5
+CME ERROR: 4
AT+QCFG="usbnet",5
+CME ERROR: 4
AT+QCFG="usbnet",5
OK
AT+QCFG="usbnet"
+QCFG: "usbnet",5

OK
AT+QCFG="nat",0
OK
```

Configured in network card mode

Enter the "AT" command to query the dialing mode, then press Enter. You can see that it is our configured "0", which is the network card mode.

```
root@ATK-DLRK356X:/# microcom /dev/ttyUSB2 -s 115200
AT+QCFG="usbnet",5
+CME ERROR: 4
AT+QCFG="usbnet",5
+CME ERROR: 4
AT+QCFG="usbnet",5
OK
AT+QCFG="usbnet"
+QCFG: "usbnet",5

OK
AT+QCFG="nat",0
OK
[ 2378.746366] xhci-hcd xhci-hcd.0.auto: Failed mapping bounce buffer, not aligning
AT+QCFG="nat"
+QCFG: "nat",0

OK
```

Query dial-up mode

Press the Ctrl + x key combination to exit the interactive mode, and then restart the development board.

3.23.1.3 Internet Test

After the configuration is completed, restart the development board, wait for a moment, and input the "ifconfig" command to check the obtained IP address.

ifconfig

```
root@ATK-DLRK356X:/# [   33.759310] vcc5v0_otg: disabling
[   33.759357] vcc3v3_pcie: disabling
[   35.423589] IPv6: ADDRCONF(NETDEV_CHANGE): usb0: link becomes ready
ifconfig
eth0      Link encap:Ethernet  HWaddr A2:2E:4D:67:C8:80
          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:38

eth1      Link encap:Ethernet  HWaddr 9E:2E:4D:67:C8:80
          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:50

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:392 errors:0 dropped:0 overruns:0 frame:0
          TX packets:392 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:28173 (27.5 KiB)  TX bytes:28173 (27.5 KiB)

usb0      Link encap:Ethernet  HWaddr 1A:A7:66:EE:03:AA
          inet addr:10.86.8.164  Bcast:10.86.8.255  Mask:255.255.255.0
          inet6 addr: fe80::18a7:66ff:feee:3aa/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:17 errors:0 dropped:0 overruns:0 frame:0
          TX packets:31 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3382 (3.3 KiB)  TX bytes:3474 (3.3 KiB)

root@ATK-DLRK356X:/#
```

usb0 is already ready.

Obtained the IP

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

ping www.baidu.com -I usb0

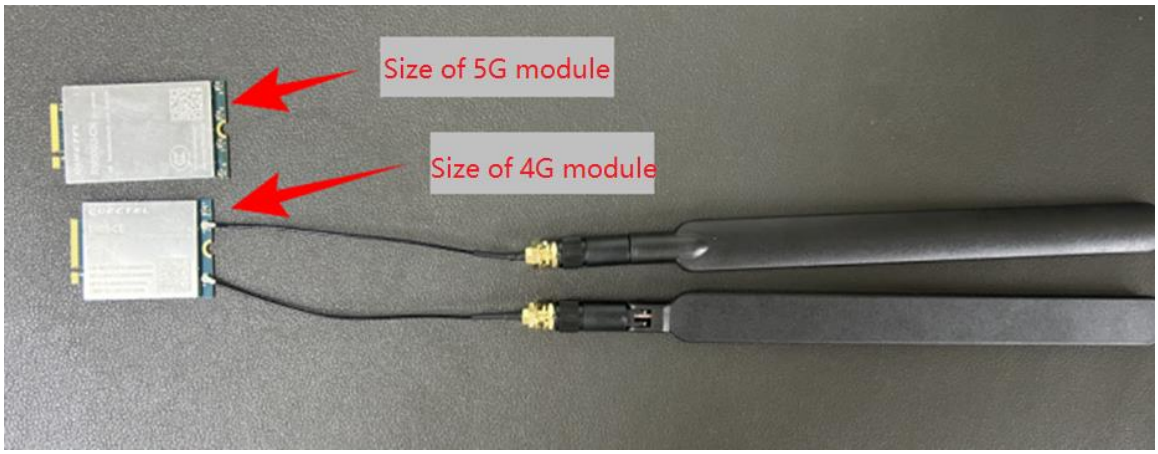
```
root@ATK-DLRK356X:/# ping www.baidu.com -I usb0
PING www.a.shifen.com (120.232.145.144) from 10.86.8.164 usb0: 56(84) bytes of data.
64 bytes from 120.232.145.144 (120.232.145.144): icmp_seq=1 ttl=51 time=25.6 ms
64 bytes from 120.232.145.144 (120.232.145.144): icmp_seq=2 ttl=51 time=38.6 ms
64 bytes from 120.232.145.144 (120.232.145.144): icmp_seq=3 ttl=51 time=37.0 ms
64 bytes from 120.232.145.144 (120.232.145.144): icmp_seq=4 ttl=51 time=42.5 ms
64 bytes from 120.232.145.144 (120.232.145.144): icmp_seq=5 ttl=51 time=41.3 ms
^C64 bytes from 120.232.145.144: icmp_seq=6 ttl=51 time=41.7 ms

--- www.a.shifen.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 50242ms
rtt min/avg/max/mdev = 25.620/37.783/42.474/5.756 ms
root@ATK-DLRK356X:/#
```

Upon seeing the data reply, it indicates that the 5G module can access the internet.

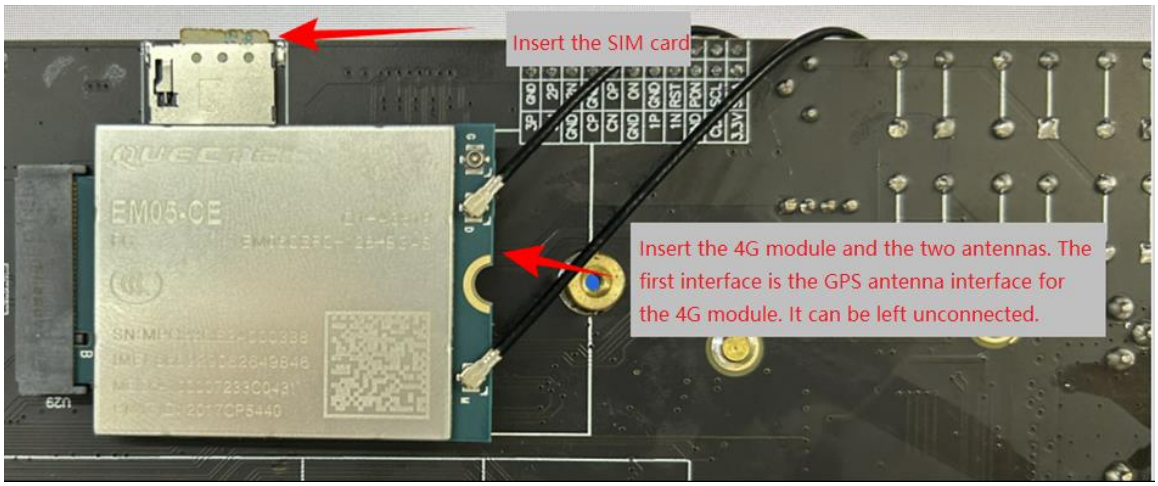
3.23.2 4G Module Internet Test

The 5G module interface is also compatible with the 4G module. Currently, the 4G module that ALIENTEK supports is the Miguang's EM05-CE. Since the EM05-CE 4G module is shorter by a section compared to the 5G module, it is impossible to fix it on the back of the development board. During the test, it can only be stuck with adhesive tape!

The following picture shows the size comparison between the 4G module and the 5G module.



Please turn off the power supply of the development board, then insert the 4G module and the SIM card (supporting all network connections). As shown in the picture, use a regular SIM card and insert it into the SIM card slot. To fix the 4G module, you need to use the EM05-CE 4G module from Quectel.



Similarly, it is necessary to check whether the node /dev/ttyUSB2 is generated under the directory /dev.

```
ls /dev/ttyUSB*
```

```
root@ATK-DLRK356X:/# ls /dev/ttyUSB*
/dev/ttyUSB0  /dev/ttyUSB1  /dev/ttyUSB2  /dev/ttyUSB3
root@ATK-DLRK356X:/#
```

Just like the 5G module, the 4G module also supports many types of drivers. Here, the author will only write about one type. For more driver types, please refer to the 4G module documentation.

3.23.2.1 Dial Test

This test is for the MBIM driver type. The programs needed for this test have been integrated into the /usr/bin directory. Check these programs using the ls command. In the following picture, Quectel* are all the test programs of Spreadtrum.

```
ls /usr/bin/quectel*
```

```
root@ATK-DLRK356X:/# ls /usr/bin/quectel*
/usr/bin/quectel-atc-proxy  /usr/bin/quectel-mbim-proxy
/usr/bin/quectel-CM         /usr/bin/quectel-qmi-proxy
root@ATK-DLRK356X:/#
```

Simply execute "quectel-CM -s ctnet &" and you can start accessing the internet.

```
root@ATK-DLRK356X:/# quectel-CM -s ctnet &
[1] 756
root@ATK-DLRK356X:/# [06-13_17:54:51:908] QConnectManager_Linux_V1.6.4
[06-13_17:54:51:909] Find /sys/bus/usb/devices/5-1 idVendor=0x2c7c idProduct=0x125, bus=0x005, dev=0x003
[06-13_17:54:51:910] Auto find qmichannel = /dev/cdc-wdm0
[06-13_17:54:51:911] Auto find usbnet_adapter = wwan0
[06-13_17:54:51:911] netcard driver = cdc_mbim, driver version = 22-Aug-2005
[06-13_17:54:51:912] Modem works in MBIM mode
[06-13_17:54:51:921] cdc_wdm_fd = 7
[06-13_17:54:51:922] mbim_open_device()
[06-13_17:54:52:712] mbim_device_caps_query()
[06-13_17:54:52:807] DeviceId:     866929062649846
[06-13_17:54:52:807] FirmwareInfo: EM05CEFCR06A04M1G_ND
[06-13_17:54:52:807] HardwareInfo: QUECTEL Mobile Broadband Modul
[06-13_17:54:52:808] mbim_device_services_query()
[06-13_17:54:52:839] mbim_set_radio_state( 1 )
[06-13_17:54:52:870] HwRadioState: 1, SwRadioState: 1
[06-13_17:54:52:870] mbim_subscriber_status_query()
[06-13_17:54:52:903] SubscriberId: 460004513219149
[06-13_17:54:52:903] SimIccId:     89860041191432879899
[06-13_17:54:52:903] SubscriberReadyState NotInitialized -> Initialized
[06-13_17:54:52:904] mbim_register_state_query()
[06-13_17:54:52:935] RegisterState Unknown -> Home
[06-13_17:54:52:935] mbim_packet_service_query()
[06-13_17:54:52:967] PacketServiceState Unknown -> Attached
[06-13_17:54:52:967] mbim_query_connect(sessionID=0)
[06-13_17:54:52:999] ActivationState Unknown -> Deactivated
[06-13_17:54:52:999] ifconfig wwan0 0.0.0.0
[06-13_17:54:53:011] ifconfig wwan0 down
[06-13_17:54:53:019] mbim_set_connect(onoff=1, sessionID=0)
[06-13_17:54:53:670] ActivationState Deactivated -> Activated
[06-13_17:54:53:670] mbim_ip_config(sessionID=0)
[06-13_17:54:53:703] < SessionId = 0
[06-13_17:54:53:703] < IPv4ConfigurationAvailable = 0xf
[06-13_17:54:53:703] < IPv6ConfigurationAvailable = 0x0
[06-13_17:54:53:704] < IPv4AddressCount = 0x1
[06-13_17:54:53:704] < IPv4AddressOffset = 0x3c
[06-13_17:54:53:704] < IPv6AddressCount = 0x0
[06-13_17:54:53:705] < IPv6AddressOffset = 0x0
[06-13_17:54:53:705] < IPv4 = 10.50.53.185/30
[06-13_17:54:53:706] < gw = 10.50.53.186
[06-13_17:54:53:706] < dns1 = 221.179.38.7
[06-13_17:54:53:706] < dns2 = 120.196.165.7
[06-13_17:54:53:707] < ipv4 mtu = 1500
[06-13_17:54:53:735] ifconfig wwan0 up
[06-13_17:54:53:747] ip -4 address flush dev wwan0
[06-13_17:54:53:755] ip -4 address add 10.50.53.185/30 dev wwan0
[06-13_17:54:53:762] ip -4 route add default via 10.50.53.186 dev wwan0
```

Obtained the IP

Use the "ifconfig" command to check the name of the corresponding network card. You can see the following picture: wwan0.

```
root@ATK-DLRK356X:/# ifconfig
eth0      Link encap:Ethernet  HWaddr A2:2E:4D:67:C8:80
          inet addr:192.168.6.211  Bcast:192.168.6.255  Mask:255.255.255.0
          inet6 addr: fe80::a02e:4dff:fe67:c880/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2251 errors:0 dropped:0 overruns:0 frame:0
          TX packets:928 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1368865 (1.3 MiB)  TX bytes:87365 (85.3 KiB)
          Interrupt:38

eth1      Link encap:Ethernet  HWaddr 9E:2E:4D:67:C8:80
          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:50

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:160 errors:0 dropped:0 overruns:0 frame:0
          TX packets:160 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:13556 (13.2 KiB)  TX bytes:13556 (13.2 KiB)

wwan0     Link encap:Ethernet  HWaddr 12:32:25:A2:25:2F
          inet addr:10.50.53.185  Bcast:0.0.0.0  Mask:255.255.255.252
          inet6 addr: fe80::1032:25[illegible]fea2:252f/64 Scope:Link
          UP BROADCAST RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:448 (448.0 B)

root@ATK-DLRK356X:/#
```

Check the corresponding network card

3.23.2.2 Internet test

To check if the internet connection is working, the most common method is to ping 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-DLRK356X:/# ping www.baidu.com -I wwan0
PING www.baidu.com (14.119.104.189) from 10.50.53.185 wwan0: 56(84) bytes of data.
64 bytes from 14.119.104.189 (14.119.104.189): icmp_seq=1 ttl=50 time=143 ms
64 bytes from 14.119.104.189 (14.119.104.189): icmp_seq=2 ttl=50 time=58.8 ms
64 bytes from 14.119.104.189 (14.119.104.189): icmp_seq=3 ttl=50 time=47.9 ms
64 bytes from 14.119.104.189 (14.119.104.189): icmp_seq=4 ttl=50 time=78.4 ms
^C64 bytes from 14.119.104.189: icmp_seq=5 ttl=50 time=78.4 ms

--- www.baidu.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 33604ms
rtt min/avg/max/mdev = 47.941/81.353/143.226/33.077 ms
root@ATK-DLRK356X:/#
```

In the above figure, you can see that there is a data response indicating that the 4G module is working properly.

3.24 View NPU information

Check the NPU usage rate. Note: The usage rate will change only if you run the routine that calls the NPU.

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

```
root@ATK-DLRK356X:/# cat /sys/kernel/debug/rknpu/load
NPU load:  0%
root@ATK-DLRK356X:/#
```

3.25 CPU Cooling Fan Test

The CPU cooling fan is an optional accessory. If you have selected a cooling fan, you can follow the steps below to test whether the fan is working properly.

Install the fan onto the motherboard, and align the two self-locking seats of the fan with the two holes reserved on the core. Then install it. To remove it, you need to remove the motherboard. From the back of the motherboard, remove the self-locking seats.

When the ATK-DLRK3568 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 and will not keep rotating. This is because it has not reached the temperature control threshold. The CPU temperature will only activate the fan for cooling when it exceeds 64 degrees. The purpose of this is to reduce the power consumption of the entire development board.

When will the CPU temperature exceed 64 degrees? In normal temperature conditions, if the development board is connected with few peripherals, it is difficult to reach 64 degrees. The fan will not start. To make the fan start, the CPU load state needs to be close to full operation.

In fact, we can also manually start the fan. First, we need to disable the temperature control strategy and execute the following command.

```
echo disabled > /sys/class/thermal/thermal_zone0/mode
```

View the current fan speed level. By default, it is divided into 22 levels.

```
cat /sys/class/thermal/cooling_device0/cur_state
```

```
root@ATK-DLRK356X:/# cat sys/class/thermal/cooling_device0/cur_state
0
root@ATK-DLRK356X:/#
```

Set the wind force level, set cur_state to 10. Different levels can be set from 0 to 22. The higher the value, the faster the fan speed.

```
echo 10 > /sys/class/thermal/cooling_device0/cur_state
```

```
root@ATK-DLRK356X:/# echo 10 > sys/class/thermal/cooling_device0/cur_state
root@ATK-DLRK356X:/# cat sys/class/thermal/cooling_device0/cur_state
10
root@ATK-DLRK356X:/#
```

In practical applications, it is not recommended to disable the temperature control strategy. This experiment only tests the normal operation of the fan.

Chapter 4. Common Function Settings of Buildroot System

This section summarizes the functions that ALIENTEK users frequently need for the system.

4.1 Mount point information

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

```
root@ATK-DLRK356X:/# df
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/root         6127168  514408   5338580   9% /
devtmpfs          1989884       0   1989884   0% /dev
tmpfs             1998844       0   1998844   0% /dev/shm
tmpfs             1998844     404   1998440   1% /tmp
tmpfs             1998844     328   1998516   1% /run
/dev/mmcblk0p7     129861   16892    108932  14% /oem
/dev/mmcblk0p8   53803957 3473906  48569922   7% /userdata
root@ATK-DLRK356X:/#
```

The important mount points are as follows:

/: Root directory, this is the common root directory where we usually find 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 Set static IP

Setting a static IP is what users need the most. Here we provide a method. Since the ALIENTEK Buildroot system is hosted by connmand, it is only applicable to the ALIENTEK system! connmand automatically obtains the IP. So after we set the static IP, connmand re-obtains the IP again. So this is not feasible. We need to tell connmand: "Don't manage my network, I need to set a static IP!" Just a joke, the actual principle is like this.

Configure the configuration file of connman. By default, the corresponding configuration file has not been created, you need to create the connman folder manually in the etc directory and create the main.conf file. When connman starts, it will automatically read this configuration file.

```
mkdir /etc/connman
vi /etc/connman/main.conf
sync
```

Write the following content in the main.conf file. For example, if I add eth1 to the blacklist, then eth1 will not be managed.

```
[General]
NetworkInterfaceBlacklist = eth1
```

After saving, restart the development board. After restarting the development board, you will notice that the eth1 network card has stopped working because it is not managed by connmand!

```
root@ATK-DLRK356X:/# ifconfig
eth0      Link encap:Ethernet  HWaddr A2:2E:4D:67:C8:80
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:15 errors:0 dropped:0 overruns:0 frame:0
          TX packets:19 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:900 (900.0 B)  TX bytes:2530 (2.4 KiB)
          Interrupt:38

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:560 errors:0 dropped:0 overruns:0 frame:0
          TX packets:560 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:39328 (38.4 KiB)  TX bytes:39328 (38.4 KiB)

root@ATK-DLRK356X:/#
```

Now let's set the static IP. Edit the /etc/network/interfaces file.

```
vi /etc/network/interfaces
```

Add the following content: Set the static IP to 192.168.9.11. Please modify it according to your own network segment.

```
auto  eth1
iface  eth1  inet  static
        address  192.168.9.11
        netmask  255.255.255.0
        network  192.168.9.0
        gateway  192.168.9.1
```

```
# interface file auto-generated by buildroot

auto lo
iface lo inet loopback

auto eth1
iface eth1 inet static
        address 192.168.9.11
        netmask 255.255.255.0
        network 192.168.9.0
        gateway 192.168.9.1
```

Modification completed. Save!

```
sync
```

Restart the development board again! Check if the set static IP is effective! You can see in the picture that eth1 has started up and a static IP of 192.168.9.11 has been set.

```
ifconfig
```

```
root@ATK-DLRK356X:/# ifconfig
eth0      Link encap:Ethernet  HWaddr A2:2E:4D:67:C8:80
          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:38

eth1      Link encap:Ethernet  HWaddr 9E:2E:4D:67:C8:80
          inet addr:192.168.9.11  Bcast:0.0.0.0  Mask:255.255.255.0
          inet6 addr: fe80::9c2e:4dff:fe67:c880/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:43 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3522 (3.4 KiB)  TX bytes:656 (656.0 B)
          Interrupt:50

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:416 errors:0 dropped:0 overruns:0 frame:0
          TX packets:416 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:29248 (28.5 KiB)  TX bytes:29248 (28.5 KiB)

root@ATK-DLRK356X:/#
```

4.3 Create a startup program

In the Linux SDK of Rockchip, Buildroot uses sysvinit. Systemd has become popular on Linux distributions such as Ubuntu and Debian, and even the high-version Yocto project uses Systemd to manage the system. Systemd is also supported on Buildroot. Let's not discuss whether they are good or bad. Even Rockchip uses sysvinit to manage the system. Personally, I think sysvinit is relatively simple and easy to understand because it doesn't require creating complex link files, dependencies, etc., nor does it require learning the syntax of Systemd. These are all side topics. In fact, whatever the SDK provides, we will use it. If the user wants to change the way of managing the system, then almost everything needs to be reconfigured for the SDK!

Creating a self-starting program is quite simple. All you need to do is add your own startup program in the /etc/init.d/ directory. The author has created a template as follows.

For example, a file named S99example was created under /etc/init.d/. Note that it must be formatted as SxxXXX. The value of xx should preferably be between 50 and 99. The larger the number, the later the startup. It cannot exceed 100. XXX can be freely named and should not contain numbers or special characters.

```
vi /etc/init.d/S99example          # Edit the file
chmod 777 /etc/init.d/S99example   # Grant execution permission
sync                               # Save
```

Write the following content on S99example.

```
PATH="/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin"

start_example()
{
    # Here is where you write your auto-start program. The "&" symbol is used to run it in the background.
```

```
        # If your program is a blocking program, it must be run in the background; otherwi
se, your system will not be able to start.
        # xx represents your program path! Please write the absolute path.
        /xx/example &
    }

    stop_example()
    {
            # Terminate the program
            killall example
    }

    case "$1" in
            start)
                    echo -n "starting example... "
                    start_example
                    echo "done."
                    ;;
            stop)
                    echo -n "stoping example... "
                    stop_example
                    echo "done."
                    ;;
            restart|reload)
                    echo -n "stoping example... "
                    stop_example && sleep .3
                    echo "done."

                    echo -n "starting example... "
                    start_example
                    echo "done."
                    ;;
            *)
                    echo "Usage: $0 {start|stop|restart}"
                    exit 1
    esac

    exit 0
```

Restarting the development board allows you to see that example has started. Although the /xx/example program cannot be found, it still executed it. Here, we need to fill in our actual program.

```
starting example... done
/etc/init.d/S99example: line 9: /xx/example: No such file or directory
Starting input-event-daemon: input-event-daemon: Start parsing /etc/input-event-daemon.conf...
input-event-daemon: Adding device: /dev/input/event0...
input-event-daemon: Adding device: /dev/input/event1...
input-event-daemon: Adding device: /dev/input/event2...
input-event-daemon: Adding device: /dev/input/event3...
input-event-daemon: Adding device: /dev/input/event4...
input-event-daemon: Adding device: /dev/input/event5...
input-event-daemon: Adding device: /dev/input/event6...
input-event-daemon: Adding device: /dev/input/event7...
input-event-daemon: Adding device: /dev/input/event8...
input-event-daemon: Adding device: /dev/input/event9...
input-event-daemon: Start listening on 10 devices...
done
root@ATK-DLRK356X:/# [    7.690359] dwc3 fcc00000.dwc3: device reset
```

Chapter 5. System software related information

Here are the commonly used software versions, which are intended to help users understand the system software versions for secondary development. If you need more software, you can develop it yourself or consult for software customization and system customization.

5.1 Common system software versions

software version	Supported functions
nginx 1.12.2	High-performance web applications and services
ffmpeg 4.1.3	Multimedia processing tools
Qt5 5.15.2	Support for qml, QWidget, QWebEngine, QChart, etc. Support for sqlite, mysql drivers, etc.
gstreamer 1.20	Multimedia framework
opencv 4.4.5	Computer vision library
python 3.8.6	Advanced programming language, facilitating development
weston 10.0.0	Based on the Wayland framework, window manager and synthesizer
……	