

# MYD-LT527 Android Software Evaluation Guide



File Status: [ ] Craft [ √ ] Release	<b>FILE ID:</b>	MYIR-MYD-LT527-SW-EG-EN-A5.15
	<b>VERSION:</b>	V1.0[doc]
	<b>AUTHOR:</b>	Nico
	<b>CREATED:</b>	2024-01-18
	<b>UPDATED:</b>	2024-02-02



# Revision History

VERSION	AUTHOR	PARTICIPANT	DATE	DESCRIPTION
V1.0	nico		20240202	Initial Version





# CONTENTS

Revision History .....	- 2 -
CONTENTS .....	- 3 -
1. Overview .....	- 6 -
1.1. Hardware Resources .....	- 6 -
1.2. Software Resources .....	- 6 -
1.3. Documents .....	- 6 -
1.4. Preparation .....	- 7 -
1.5. Introduction to the basic interface .....	- 7 -
2. Core resources .....	- 10 -
2.1. CPU .....	- 10 -
1) View CPU Information Command .....	- 10 -
2) View CPU Utilization and Temperature .....	- 11 -
2.2. GPU .....	- 13 -
2.3. Memory .....	- 15 -
1) Viewing Memory Information .....	- 15 -
2) Getting Memory Usage .....	- 16 -
3) Memory Stress Test .....	- 16 -
2.4. eMMC .....	- 17 -
1) View eMMC Capacity .....	- 18 -
2) Viewing eMMC Partition Information .....	- 18 -
3) Performance testing of eMMC .....	- 19 -
2.5. RTC .....	- 20 -
1) Viewing System RTC Devices .....	- 20 -
2) Setting the system time .....	- 20 -
2.6. Watchdog .....	- 21 -
2.7. PMIC .....	- 22 -





3. Basic Peripheral Interface .....	24 -
3.1. LED .....	24 -
1) The directory for operating LEDs is /sys/class/leds .....	24 -
2) Test LED with green light as an example .....	24 -
3.2. Key .....	24 -
1) Device Tree Configuration Information .....	24 -
2) Key Testing .....	25 -
3.3. USB .....	26 -
1) View plug-in USB information .....	26 -
2) USB flash drive read/write test .....	27 -
3) Uninstalling a USB flash drive .....	28 -
3.4. Micro SD Card .....	28 -
1) Check TF card capacity .....	29 -
2) TF card performance test .....	29 -
3.5. ADC .....	30 -
1) Load ADC driver .....	30 -
2) Setting channels and reading values .....	30 -
3.6. Display .....	31 -
1) Display Solutions .....	32 -
2) HDIM .....	32 -
3) eDP .....	32 -
4) 7" LVDS display .....	33 -
3.7. Touch Panel .....	35 -
1) Touch Screen Connection .....	35 -
2) evtest command test .....	35 -
3.8. Ethernet .....	36 -
1) Configuring the Ethernet IP Address .....	36 -
2) Modify Mac Address .....	37 -
4. Network Applications .....	39 -
4.1. PING .....	39 -
1) Wiring and information output .....	39 -
2) Test pinging the extranet URL .....	39 -





3) View ip address .....	- 40 -
4.2. Iptables .....	- 41 -
1) Configure the development board iptables .....	- 41 -
2) ping test .....	- 42 -
4.3. iperf .....	- 43 -
1) Testing TCP Performance .....	- 43 -
2) Testing UDP Performance .....	- 45 -
4.4. WIFI .....	- 49 -
1) WI-FI connection test .....	- 49 -
5. multimedia applications .....	- 53 -
5.1. Camera .....	- 53 -
1) mipi 5640 (CSI1) Camera Test .....	- 53 -
2) mipi 5640 (CSI2) Camera Test .....	- 54 -
6. References .....	- 56 -
Appendix A .....	- 57 -
Warranty & Technical Support Services .....	- 57 -





# 1. Overview

The Android Software Evaluation Guide introduces the testing procedures and evaluation methods for core and peripheral resources under the open source Android system running on MYIR's development board. This document can be used as a pre-evaluation guide or as a test guide for general system development.

## 1.1. Hardware Resources

MYIR Electronics' MYD-LT527 board consists of a core board, MYC-LT527, and a backplane, MYB-LT527, which are soldered to the backplane using postage stamp holes. In addition, MYIR provides a wealth of software resources and documentation. Please refer to the "MYD-LT527 Product Manual" for detailed configuration parameters of the hardware part. In addition, there are a number of accessories that the user may need during the evaluation and testing process, see the list below.

Table 1-1.Optional Modules

Accessories	Interface	Description
camera	MIPI	MY-CAM003M MIPI Camera Module <a href="https://www.myirtech.com/list.asp?id=611">https://www.myirtech.com/list.asp?id=611</a>
LCD	lvds	MY-TFT070CV2 (7Inches LCD with Capacitive Touch Panel ) <a href="https://www.myirtech.com/list.asp?id=634">https://www.myirtech.com/list.asp?id=634</a>

## 1.2. Software Resources

MYD-LT527 development board BSP is based on the official open-source community version of the Linux BSP port and modification of the Bootloader, Kernel and file system parts of the software resources are all open in the form of source code, for details, please see the "MYD-LT527\_Android\_SDK Release Notes".

## 1.3. Documents

MYD-LT527 development board BSP is based on the official open-source community version of the Linux BSP port and modification of the Bootloader,



Kernel and file system parts of the software resources are all open in the form of source code, for details, please see the "MYD-LT527\_Android\_SDK Release Notes".

## 1.4. Preparation

Before you start evaluating the board software, you need to make some necessary preparations for the board and configure some basic environments, including correct hardware wiring, configuring the debugging serial ports, and setting up the startup. You can refer to the MYD-LT527 Quick Start Guide for detailed procedures. The next section focuses on how to evaluate and test the hardware resources and interfaces of the system as well as the software functions.

It is mainly tested with the help of some commonly used tools and commands under Android, as well as self-developed applications. The software evaluation guide is divided into several parts, including: core resources, peripheral resources, network applications, multimedia applications, development support applications, system tools and other categories. Subsequent chapters will provide a comprehensive explanation of each section and describe in detail the specific evaluation methods and procedures for each section.

## 1.5. Introduction to the basic interface

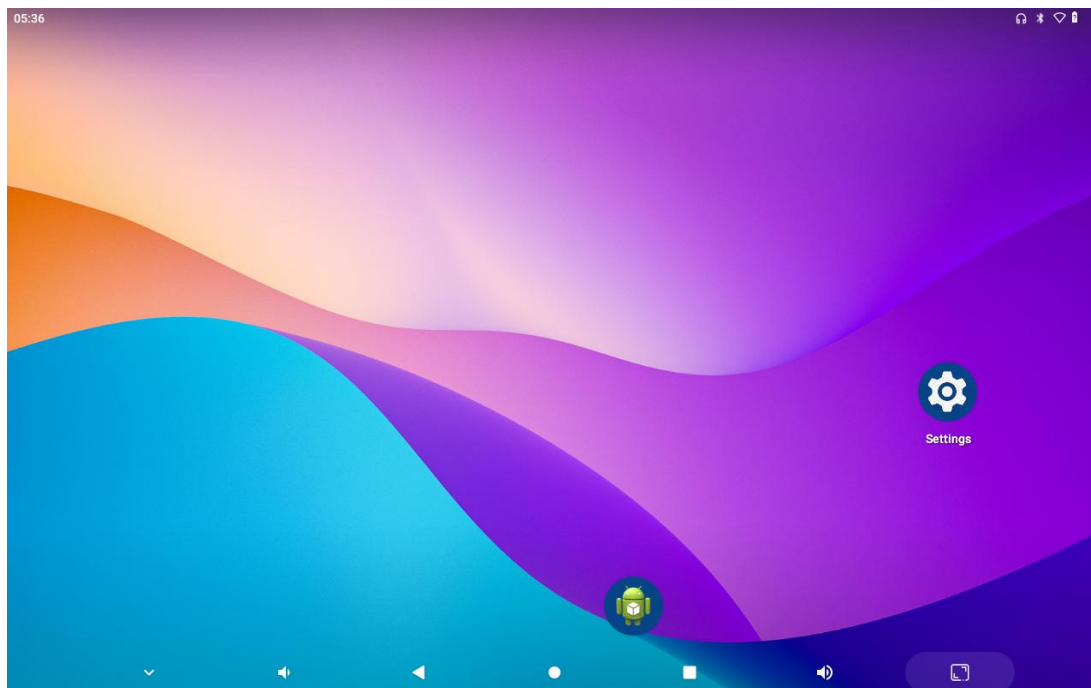


Figure 1-1 System main interface



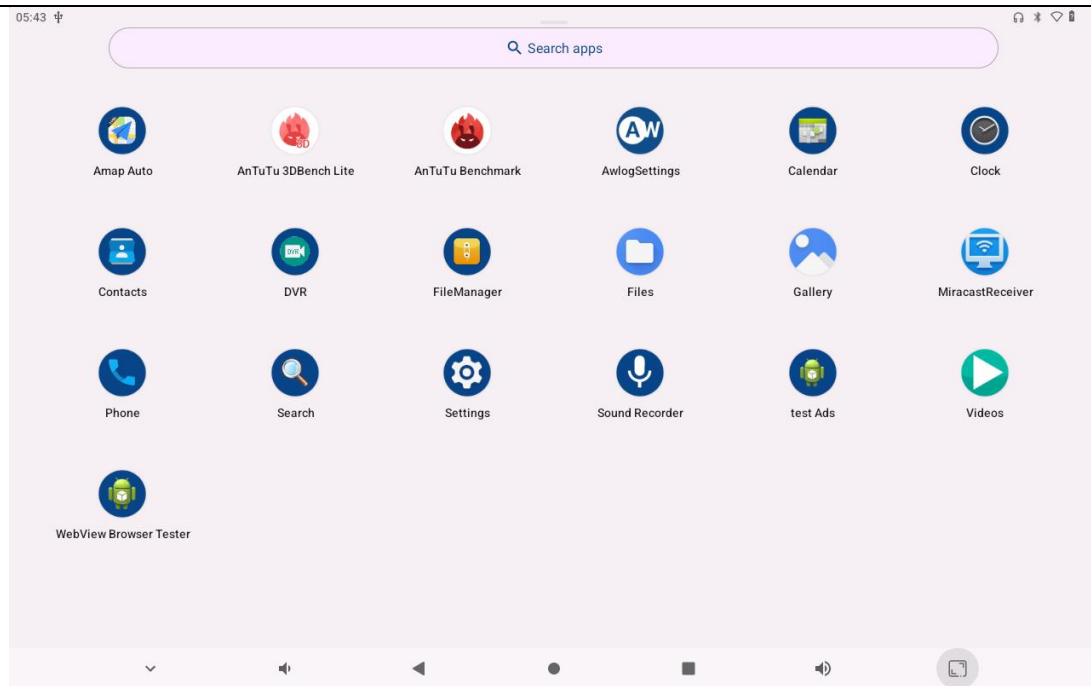


Figure 1-2 System main interface

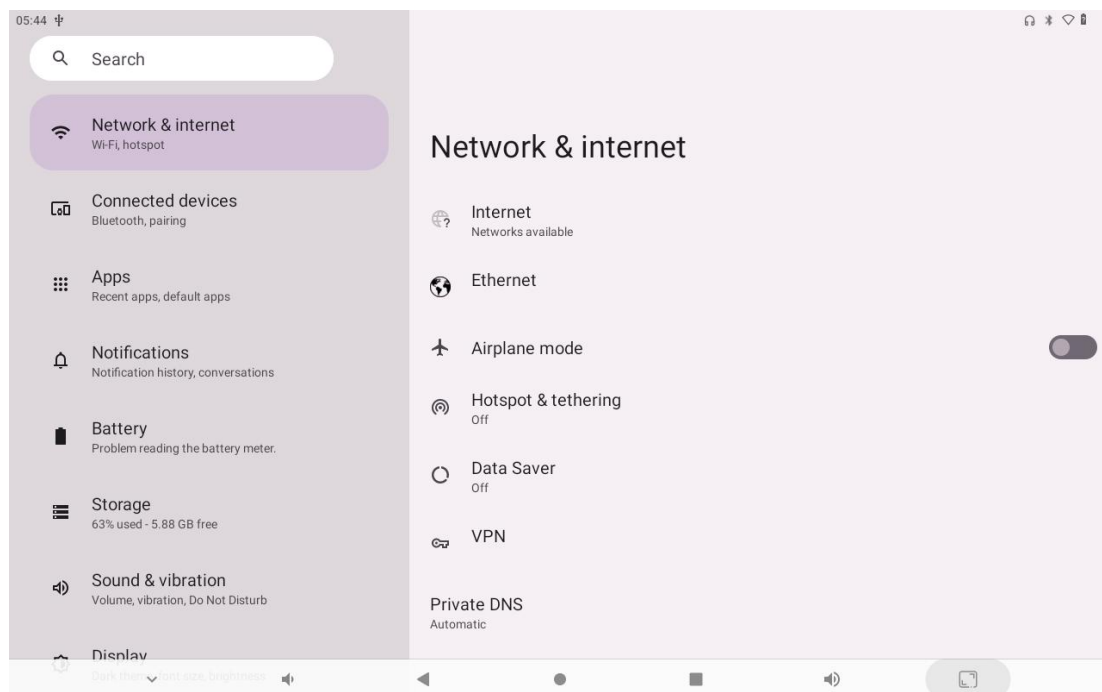


Figure 1-3 Setting interface





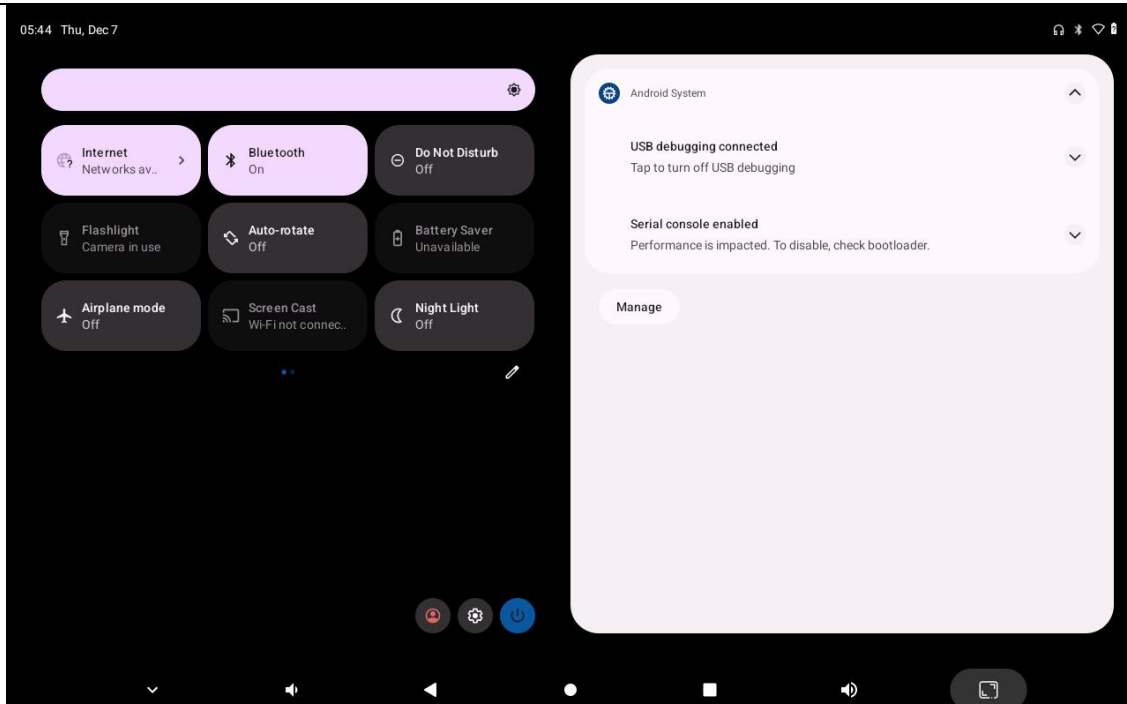


Figure1-4 status bar



## 2. Core resources

In Android system, the proc virtual file system is provided to query the parameters of each core resource and some general tools to evaluate the performance of resources. In order to make the following test more convenient and intuitive, we will show it on the Android graphical interface. Before the test, we will download the AnTuTu evaluation software, and as for how to install the software under the Android system, please refer to "MYD-LT527\_Android System Developer's Guide". The following section will read and test the parameters of the core resources, such as CPU, memory, eMMC, RTC and so on. In the following, we will read and test the parameters of CPU, memory, eMMC, RTC and other core resources.

This chapter uses AnTuTu related APP which is not provided in the SDK, if you need the APP file, please contact the sales staff to provide it

### 2.1. CPU

#### 1) View CPU Information Command

First of all, in the main interface, hold your finger on the screen and swipe up to enter the application interface, open the AnTuTu Benchmark software, click on "My Device", and then scroll down to find the CPU information.

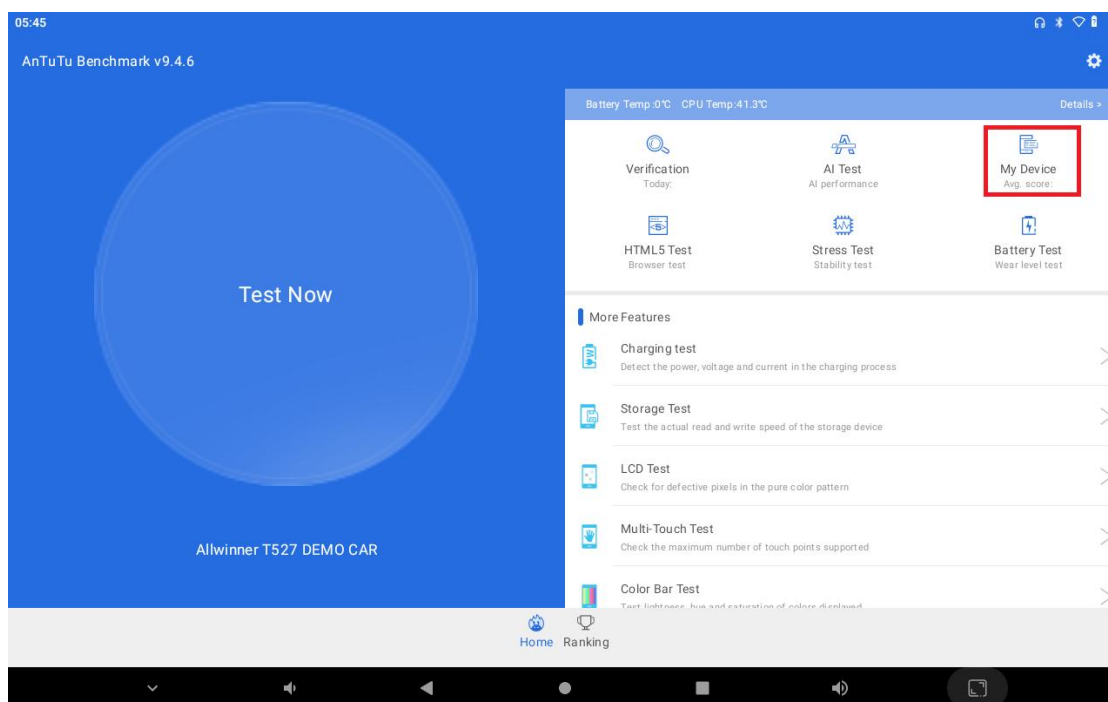


Figure 2-1 AnTuTu Features Home

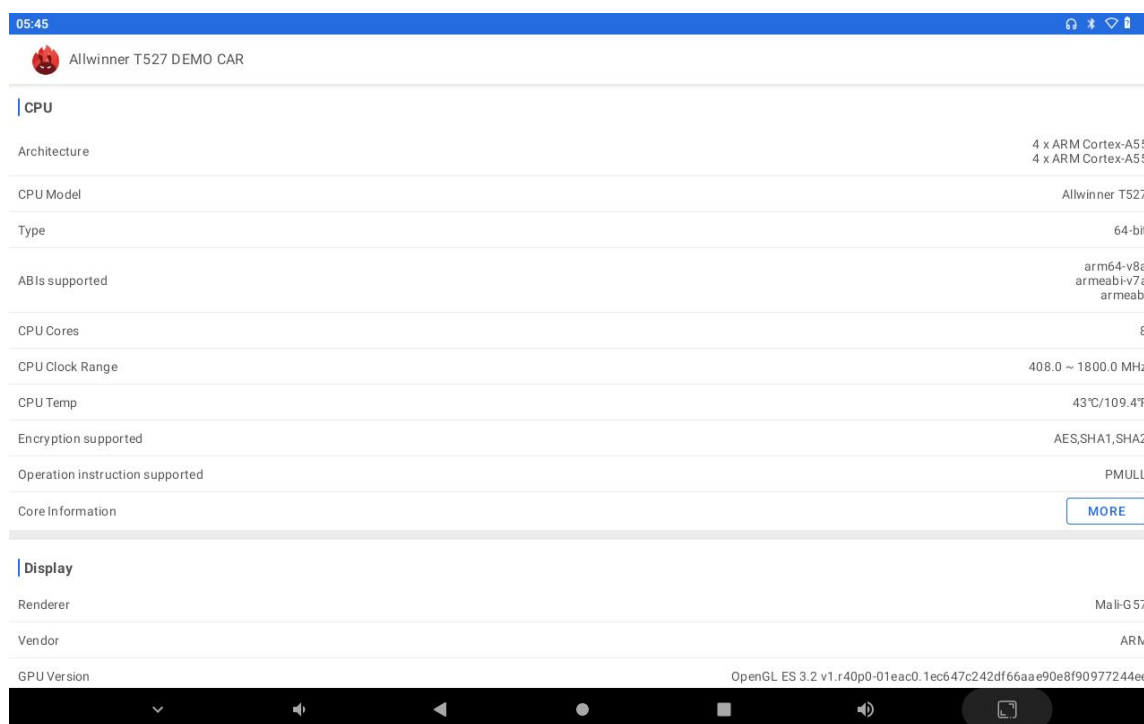


Figure 2-2 CPU information

## 2) View CPU Utilization and Temperature

Click on "Details" on the first page of AnTuTu's features, and then click on the "CPU" option to see the CPU usage and temperature.

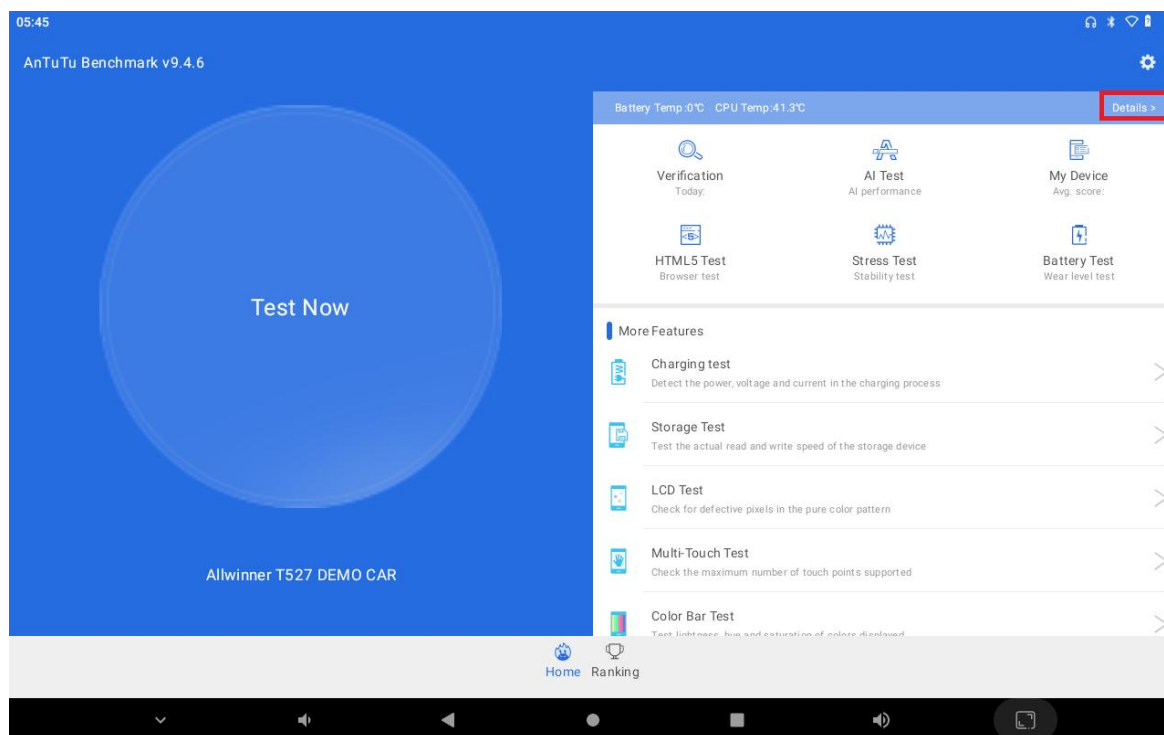


Figure 2-3 AnTuTu Features Home



Figure 2-4 CPU usage



## 2.2. GPU

Graphics Processing Unit (GPU), also known as display core, vision processor, display chip, is a microprocessor specialized in image and graphics-related operations on personal computers, workstations, game consoles, and some mobile devices (such as tablets, smartphones, etc.). As the core of the display system, the graphics processor has a powerful data computing capability and realizes the functions of two-dimension/three-dimension (2D/3D) graphics processing, image processing and display control in the form of hardware gas pedal.

The MYD-LT527 chip has an internal GPU module that supports 2D and 3D acceleration. For GPU testing, you need to download "AnTuTu 3DBrnch Lite" and open the application for GPU testing.

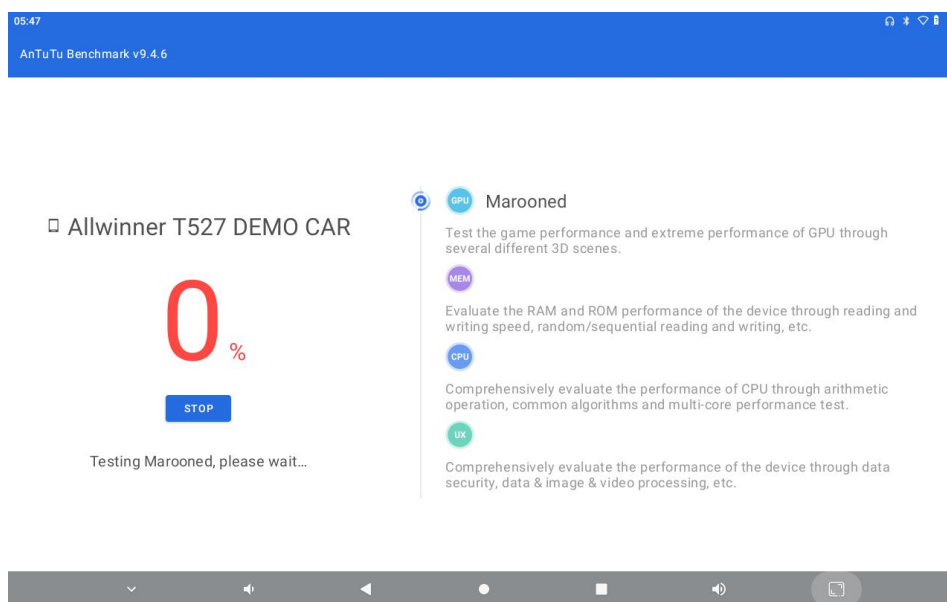


Figure 2-5 AnTuTu Review 3DLite Features Home





Figure 2-6 GPU/GPU Testing





## 2.3. Memory

There are several versions of MYD-LT527 memory, and the system divides the memory into device memory (CMA) and system memory (MEM). Device memory is a contiguous space for the driver to use, and system memory is a space allocated to the user state.

### 1) Viewing Memory Information

The `/proc/meminfo` file provides parameterized information about the memory in the system.

```
console:/ # cat /proc/meminfo
MemTotal:      1978356 kB
MemFree:       165364 kB
MemAvailable:  909776 kB
Buffers:       9820 kB
Cached:        695884 kB
SwapCached:    34968 kB
Active:        233788 kB
Inactive:      820620 kB
Active(anon):  34088 kB
Inactive(anon): 321932 kB
Active(file):  199700 kB
Inactive(file): 498688 kB
Unevictable:   4028 kB
Mlocked:      4028 kB
SwapTotal:    1483760 kB
SwapFree:     1160576 kB
Dirty:        112 kB
Writeback:     0 kB
AnonPages:    341156 kB
Mapped:       351152 kB
Shmem:        4304 kB
KReclaimable: 114020 kB
Slab:         218908 kB
```





```
SReclaimable:      88176 kB
SUnreclaim:        130732 kB
KernelStack:       22112 kB
```

- MemTotal: Size of all available RAM, physical memory minus reserved bits and kernel usage
- MemFree: LowFree+HighFree
- Buffers: Size used to cache the block device
- Cached: Buffer size of the file
- SwapCached: Memory that has been swapped out. Related to I/O
- Active: Frequently (recently) used memory
- Inactive: Recently unused memory

## 2) Getting Memory Usage

You can use the “free” command to read the memory usage, and the “-m” parameter represents the unit of MByte..

```
console:/ # free -m
```

	total	used	free	shared	buffers
Mem:	1931	1856	75	5	9
-/+ buffers/cache:		1846	85		
Swap:	1448	318	1130		

- total: Total Memory
- used: Amount of memory used
- free: Amount of memory available

## 3) Memory Stress Test

Given the size and number of memory tests, you can test the system's existing memory under pressure. You can use the system tool memtester test, such as the specified memory size of 512MB, the number of tests for 10, the test command is "memtester 512M 10".

The following is an example of a single test using 512MB of memory space:

```
console:/ # memtester 512M 10
memtester version 4.2.1 (32-bit)
```







Copyright (C) 2010 Charles Cazabon.

Licensed under the GNU General Public License version 2 (only).

pagesize is 4096

pagesizemask is 0xffff000

want 512MB (536870912 bytes)

got 512MB (536870912 bytes), trying mlock ...locked.

Loop 1:

Stuck Address : ok

Random Value : ok

Compare XOR : ok

Compare SUB : ok

Compare MUL : ok

Compare DIV : ok

Compare OR : ok

Compare AND : ok

Sequential Increment: ok

Solid Bits : ok

Block Sequential : ok

Checkerboard : ok

Bit Spread : ok

Bit Flip : ok

Walking Ones : ok

Walking Zeroes : ok

8-bit Writes : ok

16-bit Writes : ok

Done

## 2.4. eMMC

This section describes the eMMC tests for development boards configured with eMMC memory. The eMMC is a data storage device that includes a MultiMediaCard (MMC) interface and a NAND Flash component. Its cost, small



size, Flash technology independence, and high data throughput make it ideal for embedded products.

## 1) View eMMC Capacity

First of all, click "Settings" on the desktop to enter the settings interface. Then find the "Storage" option, where you can see the total capacity and used capacity of eMMC.

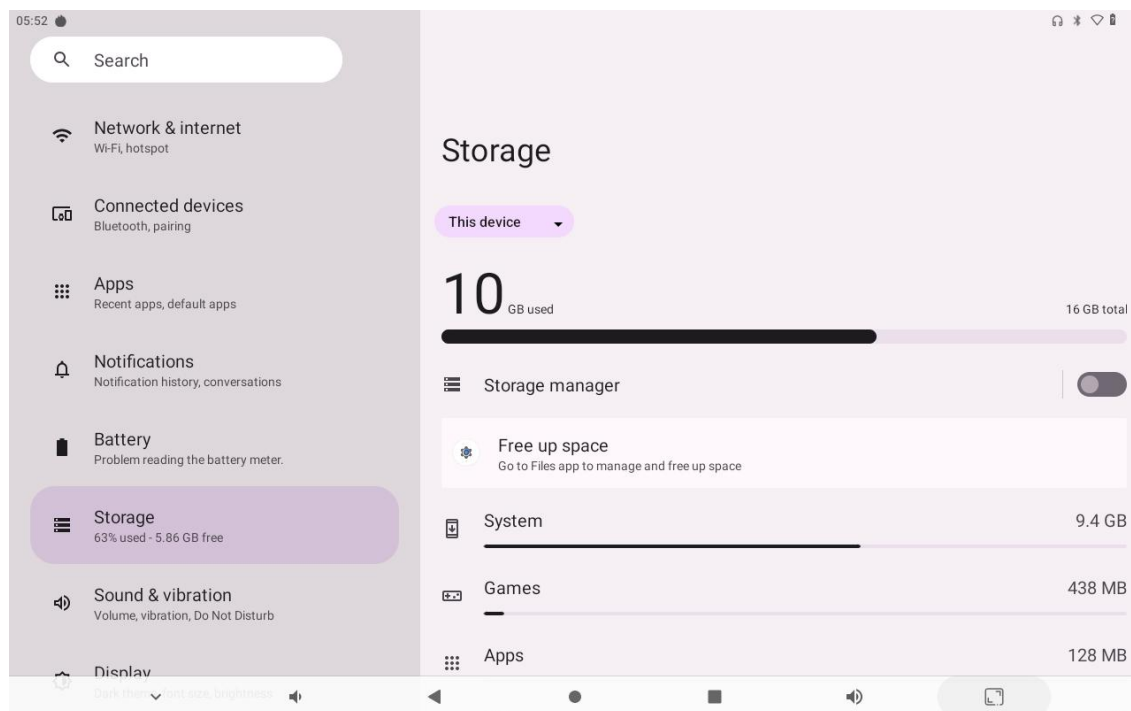


Figure 2-7 eMMC Capacity

## 2) Viewing eMMC Partition Information

With the df command, you can query the eMMC partition information, usage, mount directory and other information

```
console:/ # df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
tmpfs	966M	1.2M	965M	1%	/dev
tmpfs	966M	0	966M	0%	/mnt
/dev/block/mmcblk0p21	10M	132K	10M	2%	/metadata
/dev/block/dm-0	924M	924M	0	100%	/
/dev/block/mmcblk0p22	80M	59M	21M	74%	/treadahead
/dev/block/mmcblk0p27	64M	0	64M	0%	/Reserve0





/dev/block/dm-5	3.4G	280M	3.2G	8%	/mnt/scratch
overlay	3.4G	280M	3.2G	8%	/system
overlay	3.4G	280M	3.2G	8%	/system_dlkm
overlay	3.4G	280M	3.2G	8%	/vendor
overlay	3.4G	280M	3.2G	8%	/vendor_dlkm
overlay	3.4G	280M	3.2G	8%	/product
/dev/block/mmcblk0p26	16M	0	16M	0%	/oem
tmpfs	966M	8.0K	966M	1%	/apex
tmpfs	966M	496K	966M	1%	/linkerconfig
/dev/block/dm-6	11G	6.3G	4.1G	61%	/data
tmpfs	966M	0	966M	0%	/data_mirror
/dev/fuse	11G	6.3G	4.1G	61%	/mnt/user/0/emulated
/dev/block/vold/public:179,17	15G	256K	15G	1%	/mnt/media_rw/extsd
/dev/fuse	15G	256K	15G	1%	/mnt/user/0/extsd

- tmpfs: In-memory virtual file system, mounted to a different directory.
- devtmpfs: Used for system creation of dev

### 3) Performance testing of eMMC

The performance test is mainly to test the speed of eMMC reading and writing files under linux system, usually combined with the time and dd commands to test, due to the root directory under android has permission restrictions, so the test is conducted in the "/data" directory.

#### ● Write file test

```
console:/data # time dd if=/dev/zero of=tempfile bs=1M count=100 conv=fsy
nc

conv=1
100+0 records in
100+0 records out
104857600 bytes (100 M) copied, 0.980718 s, 102 M/s
    0m01.01s real    0m00.01s user    0m00.49s system
```

Here you can derive an eMMC write speed of 102 MB/s.





- **Read file test (clear the cache first to eliminate the effects of caching)**

```
console:/data # echo 3 > /proc/sys/vm/drop_caches
console:/data # time dd if=tempfile of=/dev/null bs=1M count=100

100+0 records in
100+0 records out
104857600 bytes (100 M) copied, 0.740895 s, 135 M/s
    0m00.81s real    0m00.02s user    0m00.30s system
```

It can be shown that the read speed is 135MB/s for direct reads from the disk

## 2.5. RTC

RTC (Real-time clock) itself is a clock, used to record the real time, when the software system shutdown Retain the system time and continue to carry on the timing, the system reopened in the time synchronization into the software system. MYD-LT527 has internal RTC, and external RTC (RX8025), if the actual product on the RTC power consumption requirements are not very high, the RTC test is usually used in the Linux system commonly used hwclock and date commands with the following test will be the system time written to the RTC, read the RTC time and set it to the system time and time power down to maintain the test. test

### 1) Viewing System RTC Devices

```
console:/ # ls /dev/rtc* -al
crw-r----- 1 system system 247,  0 1970-01-01 08:00 /dev/rtc0
crw----- 1 root  root  247,  1 1970-01-01 08:00 /dev/rtc1
```

### 2) Setting the system time

After setting the system time to Thu Dec 21 14:32:00 UTC 2023:

```
console:/ # busybox date 122114322023.00
Thu Dec 21 14:32:00 UTC 2023
```

- **Write system time to RTC**

Write the system time set by the date command in the previous step to the RTC device:



```
console:/ # busybox hwclock -w
```

- **Read RTC time and set to system time**

```
console:/ # busybox hwclock -r  
Thu Dec 21 14:32:43 2023 0.000000 seconds
```

- **Power-down hold RTC time**

Turn the development board off and disconnect the power, and after a few minutes or so, power it back on and turn it on. Check the RTC time and system time:

```
console:/ # busybox hwclock -r  
Thu Dec 21 14:35:23 2023 0.000000 seconds
```

The RTC time and system time viewed after rebooting have increased by about 20 minutes from the previous setting, indicating that the RTC is working properly. If you need to test the accuracy of the RTC in detail, you can extend the power-off time for 24 hours to test the difference between the RTC time and the standard time.

- **Synchronize system time with RTC time**

```
console:/ # busybox hwclock -s  
console:/ # date  
Thu Dec 21 22:37:39 CST 2023
```

If you add the hwclock-s command to your boot script, you can ensure that the system time and RTC time are synchronized every time you boot up.

## 2.6. Watchdog

Normally, when users want to stop the watchdog, they can stop the watchdog by writing the character "V" to the wdt node. However, MYD-LT527 watchdog driver doesn't support this way to stop the watchdog. This section demonstrates the use of watchdog, simulates a kernel crash to test the watchdog system reset function, and provides a modified example to set the watchdog timeout.

- **Simulating a kernel crash**

Simulate kernel crash, test watchdog reset function, default 32s reboot system:

```
console:/ # echo c > /proc/sysrq-trigger
```





```
[12193.548687][ T3574] sysrq: Trigger a crash
NOTICE: sunxi_usb_dev_register
NOTICE: sunxi_usb_main_loop
weak:otg_phy_config
NOTICE: usb init ok
set address 0x1f
set address 0x1f ok
err: unkown wValue(768)
err: unkown wValue(768)
```

## 2.7. PMIC

This chapter demonstrates the Suspend function of power management, which allows the board to sleep and wake up through external events. Linux kernel generally provides three kinds of Suspend: Freeze, Standby and STR (Suspend to RAM), which can be triggered by writing "freeze", and "mem" respectively to the file "/sys/power/state" in the user space. "MYD-LT527 supports freeze and mem.

### 1) View the current modes supported by the board

```
console:/ # cat /sys/power/state
freeze mem
```

**Note:** The current hardware version of this board does not support "mem" wakeup, you need to wait for the next update.

### 2) Methods for writing in userspace

```
console:/ # echo "freeze" > /sys/power/state
```

#### ● Freeze hibernation

After inputting the hibernate command, the board hibernates, at which time the user space is frozen, all I/O devices enter the low-power state, and the processor enters the idle state.

```
console:/ # echo enabled > /sys/devices/platform/soc@3000000/2500000.uart/
tty/ttyAS0/power/wakeup
console:/ # echo "freeze" > /sys/power/state
NOTICE: [SCP ERROR] :message process error
```



```
NOTICE: [SCP ERROR] :message state : 2
NOTICE: [SCP ERROR] :message attr : 2
NOTICE: [SCP ERROR] :message type : 26
NOTICE: [SCP ERROR] :message result : ff
```

At this point, the user only needs to press any key in the serial terminal to wake up the system:

```
console:/ #
```





## 3. Basic Peripheral Interface

### 3.1. LED

Linux provides a separate subsystem for operating LED devices from user space. This subsystem provides interfaces to LED devices in the form of files. These interfaces are located in the `/sys/class/leds` directory. We have listed all the LEDs on the board in the hardware resource list, and we will test the LEDs by reading and writing the sysfs commands. The following commands are generic commands and are the most common way to manipulate the LEDs.

#### 1) The directory for operating LEDs is `/sys/class/leds`

```
console:/ # cd /sys/class/leds/  
console:/sys/class/leds # ls  
green red
```

#### 2) Test LED with green light as an example

- Extinguish LED

```
console:/sys/class/leds # echo 0 > /sys/class/leds/green/brightness
```

- Light up the LEDs

```
console:/sys/class/leds # echo 1 > /sys/class/leds/green/brightness
```

### 3.2. Key

The `/dev/input/eventx` device on Linux can be used to debug input devices such as mouse, keyboard, touchpad, etc. This section focuses on testing the keys, using hexdump and dmesg commands to see if the keys respond. This section focuses on testing the keys, using the hexdump command and the dmesg command to see if the keys react.

S2 reset key

S3 USER key

S4 FEL (burn function)

S5 Wake up (No current hardware version)

#### 1) Device Tree Configuration Information







Open the companion device tree file

"*longan\device\config\chips\t527\configs\demo\_car\linux-5.15\board.dts*" to see the node for the key S3 user button:

```
gpio-keys {  
    compatible = "gpio-keys";  
    status = "okay";  
    home-key {  
        gpios = <&pio PI 11 GPIO_ACTIVE_LOW>;  
        linux,code = <172>;  
        label = "user key";  
        debounce-interval = <10>;  
        wakeup-source = <0x1>;  
        gpio-key,wakeup;  
    };  
};
```

## 2) Key Testing

- View the corresponding input getevent information

```
console:/ # getevent  
  
add device 1: /dev/input/event4  
    name:      "sunxi-keyboard"  
add device 2: /dev/input/event3  
    name:      "audiocodec Headphones"  
add device 3: /dev/input/event2  
    name:      "generic ft5x06 (79)"  
add device 4: /dev/input/event0  
    name:      "axp2202-pek"  
add device 5: /dev/input/event1  
    name:      "gpio-keys"
```

From the above, we can see that the corresponding device event of gpio-keys is event1, and users can find the corresponding event of gpio-keys according to their actual situation.





## ● getevent test key information

Execute the following command to operate key S3 and the serial terminal will print out the following message:

```
getenforce getevent
console:/ # getevent

add device 1: /dev/input/event4
  name:      "sunxi-keyboard"
add device 2: /dev/input/event3
  name:      "audiocodec Headphones"
add device 3: /dev/input/event2
  name:      "generic ft5x06 (79)"
add device 4: /dev/input/event0
  name:      "axp2202-pek"
add device 5: /dev/input/event1
  name:      "gpio-keys"
```

5

```
/dev/input/event1: 0001 00ac 00000001
/dev/input/event1: 0000 0000 00000000
/dev/input/event1: 0001 00ac 00000000
/dev/input/event1: 0000 0000 00000000
/dev/input/event1: 0001 00ac 00000001
```

Each time S3 is pressed the current terminal prints out the current event code value, i.e. the key is pressed normally.

## 3.3. USB

This section verifies the feasibility of the USB Host driver through related commands or hot-plugging, USB HUB to realize the function of reading and writing USB flash drives, and usb enumeration.

### 1) View plug-in USB information



## ● Viewing USB Device Information

After inserting the USB device, click "Setting" to find "Storage", click the red box to find the name of your corresponding USB device, and then you can see the current details.

**Note:** When the USB HOST interface used is the J5 upper port, the jumper cap at JP1 should be at 2 3 for HOST mode

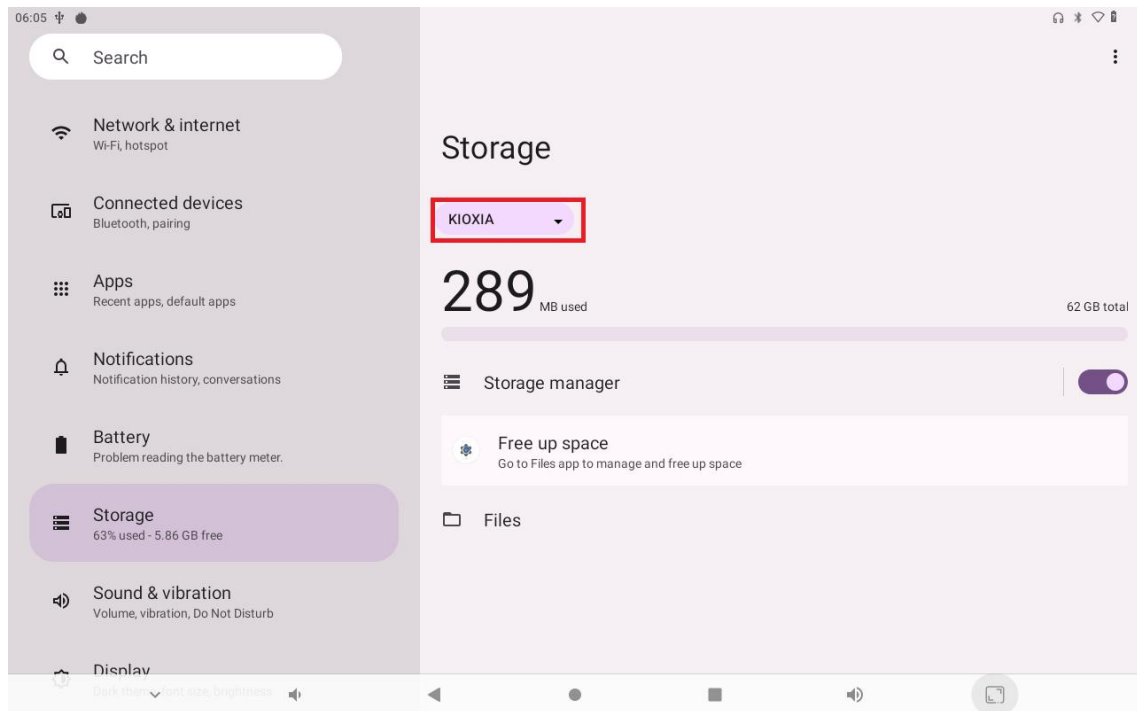


Figure 3-1 USBDevice capacity

## 2) USB flash drive read/write test

### ● Go to the mount directory

```
console:/ # cd /mnt/media_rw/usb
console:/mnt/media_rw/usb #
```

### ● read file

You need to create a test.txt file on the USB flash drive ahead of time

```
console:/mnt/media_rw/usb # ls
test.txt
console:/mnt/media_rw/usb # cat /mnt/test.txt
helloworld!
```





- **write file**

```
console:/mnt/media_rw/usb # touch test.txt
console:/mnt/media_rw/usb # echo hello> 1
console:/mnt/media_rw/usb # cat test.txt
hello
console:/mnt/media_rw/usb # sync
```

After writing the file, you need to execute the sync command to ensure that the data is completely written to the USB flash drive before uninstalling the USB flash drive device.

### 3) Uninstalling a USB flash drive

- **Uninstallation operations**

```
console:/ # umount /mnt/media_rw/usb
```

## 3.4. Micro SD Card

Micro SD Card, formerly known as Trans-flash Card (TF Card), is an extremely small flash memory card. micro SD card is the smallest of the SD card types in terms of form factor compared to the standard SD card. Although the form factor and interface shape of the Micro SD card is different from that of the original SD card, the interface specification remains the same, ensuring compatibility.

Micro SD card can be used as a standard SD card if it is inserted into a specific adapter card. SD card has become one of the most widely used memory cards in consumer digital devices, which is a multifunctional memory card with high capacity, high performance, security, etc. Micro SD card has 9 pins on the back of the card, including 4 data lines, and it supports two data transmission widths: 1bit/4bit. MYD-LT527 supports two SHMC interfaces, SDMMC1 is used on the development board to connect to Micro SD, the hardware specifications of this interface are as follows:

Supports 1bit/4bit/8bit SDMMC interface, fully compliant with SD card v3.01 interface specification Supports SDHC Class 10 MicroSD card Supports the first generation of UHS bus interface (UHS-1 speed class U3), does not support UHS-II. The maximum transfer speed of UHS-I (theoretical) is 104MB/s. The letter I indicates that the device (SD card or reader) supports UHS-I interface. The letter I



indicates that the device (SD card or card reader) supports the UHS-I interface. The letter U, including the number 3, means that the device has a read/write speed rating of U3. SDHC cards (>2GB to 32GB), SDXC cards (>32GB to 2TB) are supported. This section explains the steps and methods to view and operate TF card under Linux system.

## 1) Check TF card capacity

After inserting the USB device, click "Setting" to find "Storage", click the red box to find the corresponding SD card device name, you can see the current details.

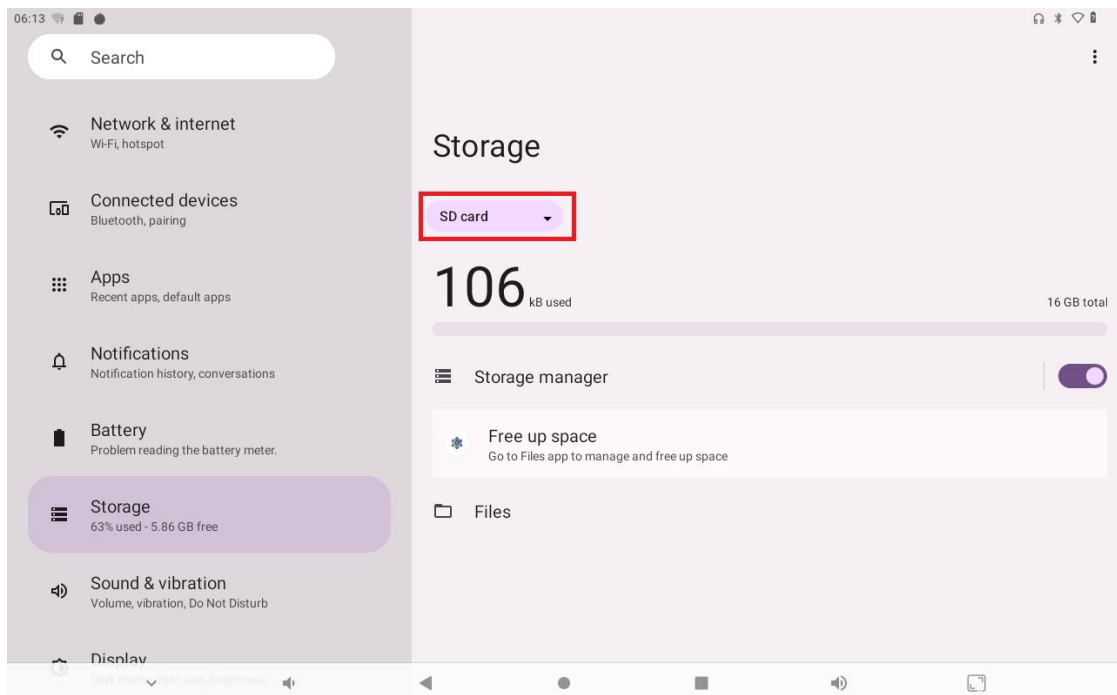


Figure 3-2 TF card capacity

## 2) TF card performance test

The performance test is mainly to test the SD card in the file system under the file reading and writing speed, usually combined with time and dd double command test

### ● Write file test

```
console:/mnt/media_rw/extsd # time dd if=/dev/zero of=test_file_w bs=1M count=500 conv=fsync
```

```
500+0 records in
```





```
500+0 records out
524288000 bytes (500 M) copied, 25.229482 s, 20 M/s
    0m25.26s real      0m00.01s user      0m02.60s system
```

Here the write disk speed is tested at 20M/s

#### ● Read file test

```
console:/mnt/media_rw/extsd # time dd if=test_file_w of=test_file_r bs=1M co
unt=500
500+0 records in
500+0 records out
524288000 bytes (500 M) copied, 21.440730 s, 23 M/s
    0m21.48s real      0m00.01s user      0m02.77s system
```

It can be seen that the speed of reading data directly from the SD card is 23m/s

## 3.5. ADC

The MYD-LT527 has two ADC controllers, each of which is an analog-to-digital converter module with 12-bit sampling accuracy, supporting two channels, with the analog input range specified by the platform. The following is an example of the ADC0 controller's 3 and 4 channel test.

### 1) Load ADC driver

```
console:/ # cd /vendor/lib/modules/
console:/vendor/lib/modules # insmod sunxi_gpadc.ko
```

### 2) Setting channels and reading values

```
console: cd /sys/devices/platform/soc@3000000/2009000.gpadc0/gpadc/gpadc_
chip0 #
console:/sys/devices/platform/soc@3000000/2009000.gpadc0/gpadc/gpadc_chip
0 #
echo 3 > data
console:/sys/devices/platform/soc@3000000/2009000.gpadc0/gpadc/gpadc_chip
0 #
cat data
voltage data is 449
```





```
console:/sys/devices/platform/soc@3000000/2009000.gpadc0/gpadc/gpadc_chip
0 #
echo 4 > data
console:/sys/devices/platform/soc@3000000/2009000.gpadc0/gpadc/gpadc_chip
0 #
cat data
voltage data is 446
console:/sys/devices/platform/soc@3000000/2009000.gpadc0/gpadc/gpadc_chip
0 #
```

### 3.6. Display

This module consists of a display engine (DE) and various types of controllers (tcon). Input layers are processed in DE and then output to display devices through one or more interfaces, so as to synthesize the layers rendered by many applications and present them to users on monitors. 2 independent units (de0 and de1 for short) of DE can accept user input layers for synthesis and output to different monitors to realize dual display. DE has 2 independent units (de0 and de1 for short), which can accept user input layers for compositing and output to different monitors to realize dual display. each independent unit of DE has 1-4 channels (typically, de0 has 4 channels and de1 has 2 channels), and each channel can process and accept 4 layers with the same format at the same time. sunxi platform has a video channel and UI channels. The video channel is powerful and can support YUV format and RGB layers. the UI channel only supports RGB layers. Briefly, the main functions of the display module are as follows:

- Support lcd(hv/lvds/cpu/dsi) output
- Support dual display output
- Support multi-layer overlay blending
- Support multiple display effect processing (alpha, colorkey, image enhancement, brightness/contrast/saturation/chroma adjustment)
- Support intelligent backlight adjustment
- Support multiple image data format input (argb, yuv)



- Support for image scaling - Support for screenshot - Support for image conversion

## 1) Display Solutions

Support HDMI, 7-inch LVD, 10-inch LVDS, eDP display

Note: HDMI and eDP can not be accessed at the same time, the default mirror can HDMI, eDP, 7-inch LVDS display alone, HDMI + 7-inch LVDS, eDP + 7-inch LVDS with the same display with the different display. If you need to switch to 10-inch LVDS display, please refer to chapter 6.2.2 of MYD-LT527\_Android System Developer's Guide to replace the display device tree.

## 2) HDIM



Figure 3-3 HDMI

## 3) eDP







Figure 3-4 eDP

#### 4) 7" LVDS display

For 7-inch LVDS screen, use 40-pin FPC cables with different sides, the blue side of the cables on both sides of the development board and the screen should be facing upwards.



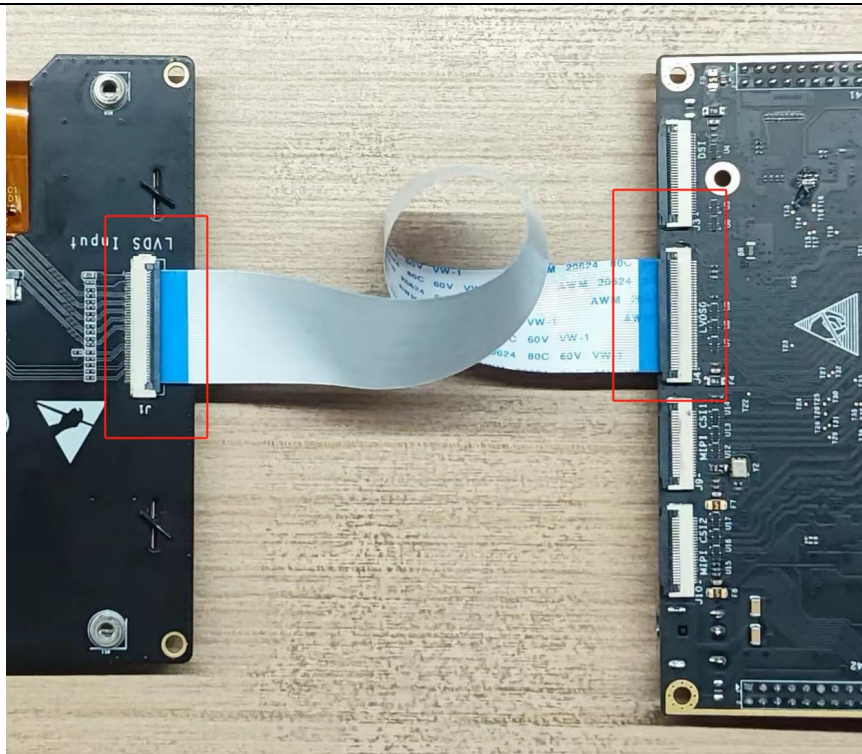


Figure 3-5 7" LVDS cable connection



Figure 3-6 LVDS





## 3.7. Touch Panel

There are capacitive touch and resistive touch, MYD-LT527 development board hardware currently does not support resistive touch, but supports capacitive touch, MYIR Technology provides LVDS display accessories, see Table 1-1. you can buy your own accessories according to the actual needs. The capacitive screen is more sensitive in use and rarely has problems. In addition, capacitive screen does not need to be more accurate. Because according to the principle of capacitive screen, capacitive screen in use is able to accurately recognize the position of the finger and screen contact, with high sensitivity. If we appear in the use of the phenomenon of clicking on the software selection, there is generally only one situation: the screen has a problem. The following is a simple test of capacitive screen touch function through the command test

### 1) Touch Screen Connection

Connect the MY-LVDS070C LVDS screen to the development board according to section 3.6.

### 2) evtest command test

Execute "getevent" in the terminal to enter the test interface. Select the test peripheral as touch screen, event3 as capacitive touch device, press enter to start the test:

```
console:/ # getevent

add device 1: /dev/input/event4
  name:      "sunxi-keyboard"
add device 2: /dev/input/event3
  name:      "audiocodec Headphones"
add device 3: /dev/input/event2
  name:      "generic ft5x06 (79)"
add device 4: /dev/input/event0
  name:      "axp2202-pek"
add device 5: /dev/input/event1
  name:      "gpio-keys"
```





3

```
/dev/input/event2: 0003 0039 00000000
/dev/input/event2: 0003 0035 000002b0
/dev/input/event2: 0003 0036 00000154
/dev/input/event2: 0001 014a 00000001
/dev/input/event2: 0003 0000 000002b0
```

## 3.8. Ethernet

There are many network configuration tools under Linux, the common ones are net-tools, iproute2, systemd-networkd, network manager and connman, etc. All of them can be selected according to the actual needs when customizing the system. MYD-LT527 has two network Gigabit ports. Here we take the eth0 Gigabit port as an example to introduce the common Ethernet configuration.

**Note:** In the case of both ports being connected, only the first device connected to the port will work.

### 1) Configuring the Ethernet IP Address

- Manually configure the network using ifconfig from the net-tools toolkit

First view the network device information through the ifconfig command as follows:

```
console:/ # ifconfig
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: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 TX bytes:0

eth0        Link encap:Ethernet  HWaddr 18:e3:02:d5:8b:55  Driver sunxi-gmac
            inet addr:192.168.40.104  Bcast:192.168.40.255  Mask:255.255.255.0
            inet6 addr: fe80::62a8:d712:cb41:2101/64 Scope: Link
```





```
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
RX packets:3416 errors:117 dropped:0 overruns:0 frame:0
TX packets:158 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:1344840 TX bytes:20223
Interrupt:169
```

The following describes how to manually configure the IP address 192.168.0.100 for eth0, with the following commands:

```
console:/ # ifconfig eth0 192.168.1.100 netmask 255.255.255.0 up
```

The above command manually configures eth0 with an IP address of 192.168.0.100, a subnet mask of 255.255.255.0, and the default configured broadcast address of 192.168.0.255, and activates it with the up parameter, as follows:

```
console:/ # ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 36:C9:E3:F1:B8:05
          inet addr:192.168.1.100  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::177a:be22:1be1:91e7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:653 errors:0 dropped:0 overruns:0 frame:0
          TX packets:72 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:49533 (48.3 KiB)  TX bytes:5639 (5.5 KiB)
          Interrupt:65
```

## 2) Modify Mac Address

To manually change the Mac address 00:0C:29:36:97:20, the command is as follows :

```
console:/ # ifconfig eth0 down
console:/ # ifconfig eth0 hw ether 00:0C:29:36:97:20
console:/ # ifconfig eth0 up
console:/ # ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0C:29:36:97:20
```





```
inet addr:192.168.1.197 Bcast:192.168.1.255 Mask:255.255.255.0
inet6 addr: fe80::dd12:1815:4621:b249/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:17112 errors:0 dropped:0 overruns:0 frame:0
TX packets:338 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:1395663 (1.3 MiB) TX bytes:27773 (27.1 KiB)
Interrupt:65
```







## 4. Network Applications

The factory-burned image of the development board contains some common network applications by default, which is convenient for users to develop or debug.

### 4.1. PING

PING is mainly used to test network connectivity, but also to test network latency and packet loss. Once you have configured your Ethernet connection, you can use PING to perform a simple test of your network connection.

#### 1) Wiring and information output

Connect the board to a switch or router via a CAT6 cable, and the console will display the connection information output by the kernel, as follows:

```
console:/ # [ 67.270410][ T69] sunxi-gmac 4500000.gmac0 eth0: Link is Up - 1Gbps/Full - flow control of
```

#### 2) Test ping the extranet URL

```
console:/ # ping www.baidu.com
```

```
PING www.a.shifen.com (153.3.238.102) 56(84) bytes of data.  
64 bytes from 153.3.238.102: icmp_seq=1 ttl=53 time=13.9 ms  
64 bytes from 153.3.238.102: icmp_seq=2 ttl=53 time=13.6 ms  
64 bytes from 153.3.238.102: icmp_seq=3 ttl=53 time=13.9 ms  
64 bytes from 153.3.238.102: icmp_seq=4 ttl=53 time=14.2 ms  
^C  
--- www.a.shifen.com ping statistics ---  
5 packets transmitted, 5 received, 0% packet loss, time 4008ms  
rtt min/avg/max/mdev = 13.633/13.903/14.203/0.245 ms
```

**Note:** To ping the public network, you need to make sure the DNS is working.

The above result shows that the IP address of www.baidu.com after domain name resolution is 153.3.238.102, icmp\_seq represents the number of icmp packets, if the number is consecutive, it means that there is no packet loss; time represents



the response delay time, of course, the shorter the better. In addition to testing Ethernet, the ping command can also be used to test Wi-Fi.

### 3) View ip address

You can check the ip address in "Setting"

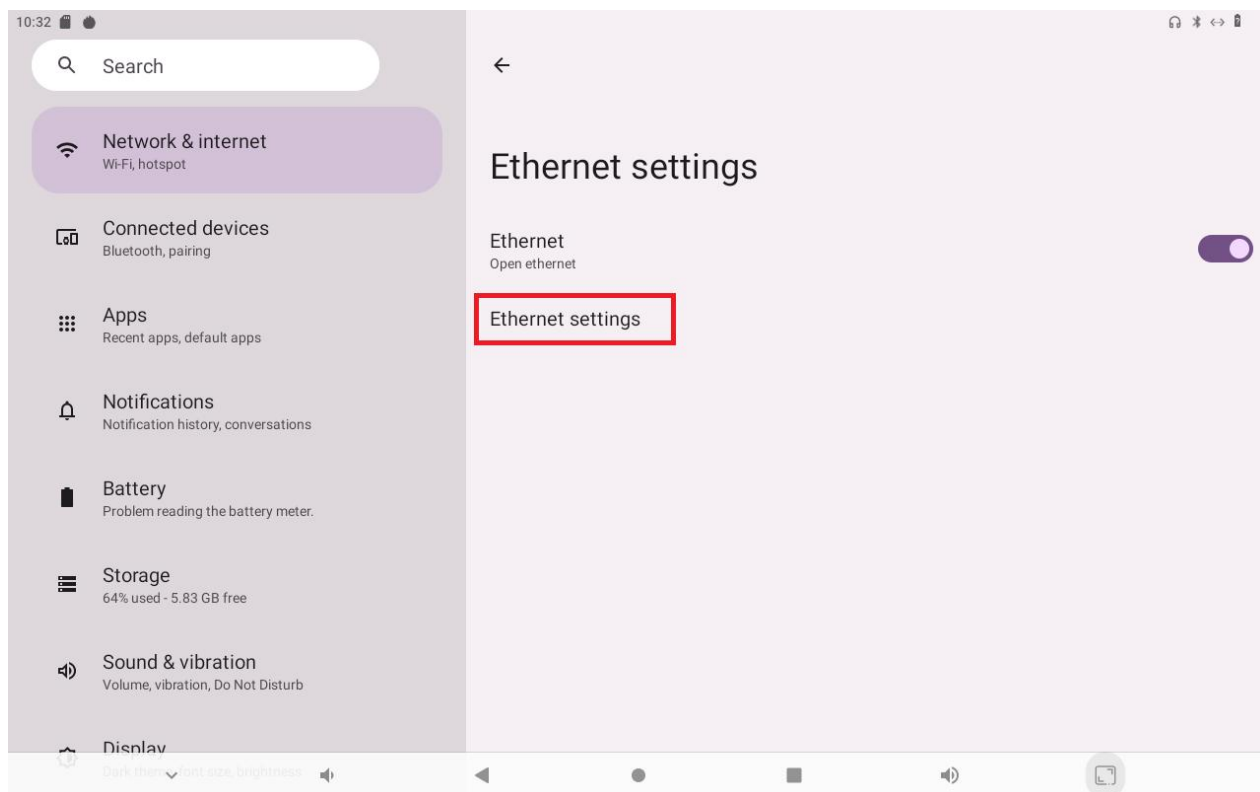


Figure 4-1 Port Information Screen





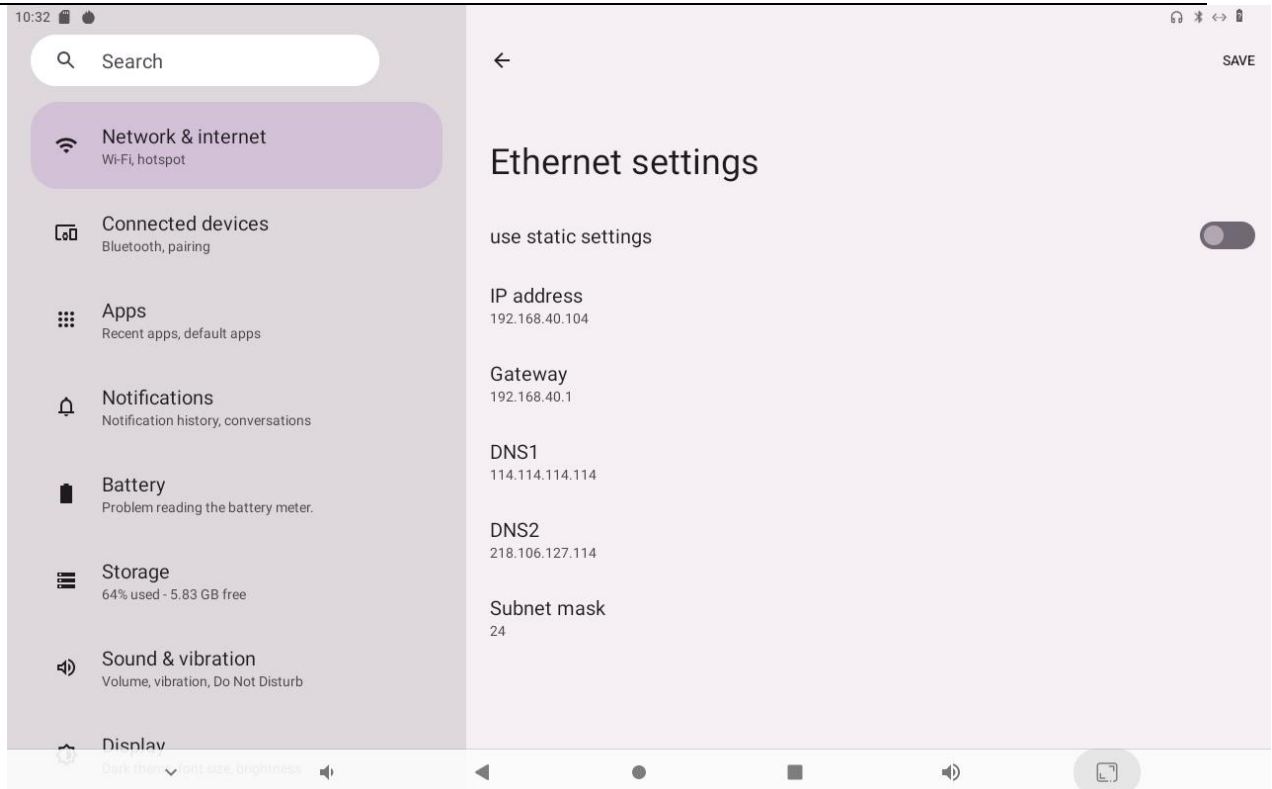


Figure 4-2 Network port IP address

## 4.2. Iptables

iptables is an administrative tool for IPv4 packet filtering and NAT. It is used to setup, maintain and check the IP packet filtering rule tables in the Linux kernel. Several different tables can be defined. Each table contains a number of built-in chains, and can also contain user-defined chains. Each chain is a list of rules that can match a set of packets. Each rule specifies what to do with the matched packets. Development boards using Linux systems often use the iptables utility to configure the firewall. iptables handles packets in a variety of ways, such as accept, reject, and drop, depending on the method defined by the packet filtering rules.

The following is a test of using iptables to intercept icmp packets and prevent other external devices on the network from pinging them. For more information on how to do this, see the: <https://linux.die.net/man/8/iptables>.

### 1) Configure the development board iptables

Use iptables on the development board to configure it to drop incoming icmp packets and not respond to ping probes from other hosts, with the command:





```
console:/ # iptables -A INPUT -p icmp --icmp-type 8 -j DROP
console:/ # iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-A INPUT -p icmp -m icmp --icmp-type 8 -j DROP
```

## 2) ping test

ping the development board on the development host and specify a deadline of 10, the result is as follows:

```
C:\Users\Lenovo>ping 192.168.1.195 -w 10

Request timed out.
Request timeout.
Request timeout.
Request timeout.

Ping statistics for 192.168.1.195.
    Packets: Sent = 10, Received = 0, Missing = 10 (100% missing)
```

The above results show that after setting the firewall, the development host cannot ping through the development board.

- **Delete the corresponding firewall rule**

```
console:/ # iptables -F
console:/ # iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
```

- **Test pinging the board again**

```
C:\Users\Lenovo>ping 192.168.1.97 -w 10
Reply from 192.168.1.97: Bytes=32 Time=1ms TTL=64
Reply from 192.168.1.97: Bytes=32 Time=1ms TTL=64
Reply from 192.168.1.97: bytes=32 time=1ms TTL=64
```





```
Reply from 192.168.1.97: bytes=32 time=1ms TTL=64
```

```
Ping statistics for 192.168.1.97.
```

```
    Packets: Sent = 10, Received = 10, Missing = 0 (0% missing).
```

```
Estimated time in milliseconds for round-trip travel.
```

```
    Min = 1ms, Max = 1ms, Avg = 1ms
```

After clearing the iptables rules, ping the board from the development host again, and it will ping through. The above example is just a simple demonstration, in fact, iptables can realize very powerful functions with various rules, so I won't go into details here.

### 4.3. iperf

iperf is a tool for actively measuring the maximum achievable bandwidth on IP networks. It supports adjusting various parameters such as test time, buffer size and protocol (TCP, UDP, SCTP for IPV4 and IPV6). iperf can be categorized into server mode or client mode according to its role. We can use it to test and view the network bandwidth, TCP window value, retransmission probability, etc. in TCP mode, and also to test the packet loss rate, latency, and jitter under the specified UDP bandwidth.

We open Windows PowerShell on the development host, the host with a Gigabit NIC serves as the server of iperf, and the development board under test serves as the client to test the TCP and UDP performance of the board's NIC. First, install iperf on the host computer, as follows:

Connect the server and the development board directly via CAT6 cable and configure their IP addresses. For example, let's set the server ip to 192.168.40.21 and the development board ip to 192.168.40.104, and use the ping command to test to make sure they are connected..

Note: Try not to connect a router or switch, so that the test results are not affected by the transmission and forwarding of the intermediate equipment.

#### 1) Testing TCP Performance





### ● server side (192.168.40.21)

On the server, iperf uses the -s parameter to indicate that it is working in server-side mode.

```
server side (192.168.40.21)
nico@system1-ubuntu1804:~$ iperf3 -s
```

```
-----
Server listening on 5201
-----
```

### ● client side (192.168.40.207)

The iperf program running on the development board works in client-side, TCP mode, with the following parameter descriptions:

- -c 192.168.40.104 : Working on the client side, connecting to the server side 192.168.40.21
- -i 2 : Test results reported at 2 second intervals
- -t 10 : Total test duration is 10 seconds

```
console:/ # iperf3 -c 192.168.40.21 -i 2 -t 10
Connecting to host 192.168.40.21, port 5201
[ 5] local 192.168.40.104 port 47184 connected to 192.168.40.21 port 5201
[ ID] Interval          Transfer      Bitrate      Retr  Cwnd
[ 5]  0.00-2.01    sec   192 MBytes   802 Mb/s     0   1.64 MBytes
[ 5]  2.01-4.00    sec   192 MBytes   807 Mb/s     0   1.81 MBytes
[ 5]  4.00-6.00    sec   192 MBytes   805 Mb/s     0   2.00 MBytes
[ 5]  6.00-8.00    sec   195 MBytes   820 Mb/s     0   2.00 MBytes
[ 5]  8.00-10.00   sec   195 MBytes   817 Mb/s     0   2.00 MBytes
-----
[ ID] Interval          Transfer      Bitrate      Retr
[ 5]  0.00-10.00   sec   966 MBytes   810 Mb/s     0          sender
```





```
[ 5] 0.00-10.05 sec 965 MBytes 806 Mbits/sec receive
r

iperf Done.
```

The client ends the test after 10 seconds and displays the above test result, which shows that the TCP bandwidth is around 810 Mbits without retransmission, and the TCP window value during the test is 50 KBytes. At the same time, the server displays the test result as follows, and then continues to listen on the port and wait for a client connection:

```
Accepted connection from 192.168.40.104, port 47174
[ 5] local 192.168.40.21 port 5201 connected to 192.168.40.104 port 47184
[ ID] Interval          Transfer      Bitrate
[ 5] 0.00-1.00 sec 88.4 MBytes 742 Mbits/sec
[ 5] 1.00-2.00 sec 97.3 MBytes 816 Mbits/sec
[ 5] 2.00-3.00 sec 95.6 MBytes 802 Mbits/sec
[ 5] 3.00-4.00 sec 96.3 MBytes 808 Mbits/sec
[ 5] 4.00-5.00 sec 96.9 MBytes 813 Mbits/sec
[ 5] 5.00-6.00 sec 95.4 MBytes 800 Mbits/sec
[ 5] 6.00-7.00 sec 98.9 MBytes 830 Mbits/sec
[ 5] 7.00-8.00 sec 96.3 MBytes 808 Mbits/sec
[ 5] 8.00-9.00 sec 98.0 MBytes 822 Mbits/sec
[ 5] 9.00-10.00 sec 97.3 MBytes 816 Mbits/sec
[ 5] 10.00-10.05 sec 4.14 MBytes 752 Mbits/sec
-----
[ ID] Interval          Transfer      Bitrate
[ 5] 0.00-10.05 sec 965 MBytes 806 Mbits/sec receive
r
-----

Server listening on 5201
-----
```

## 2) Testing UDP Performance





● **server side (192.168.40.21 )**

Continue to run iperf3 on the server using the -s parameter to indicate that it is working in server-side mode

```
nico@system1-ubuntu1804:~$ iperf3 -s
```

```
-----  
Server listening on 5201  
-----
```

● **client side (192.168.40.104)**

The iperf3 on the device works in client, UDP mode, where the parameters are described as follows:

- -u : Work in UDP mode
- -c 192.168.40.104 : Working on the client side, connecting to the server side 192.168.40.21
- -i 2 : Test results reported at 2 second intervals
- -t 10 : Total test duration is 10 seconds
- -b 100M : Set the UDP transmission bandwidth to 100 Mbps.

```
console:/ # iperf3 -c 192.168.40.21 -u -i 2 -t 10 -b 100M
```

```
Connecting to host 192.168.40.21, port 5201
```

```
[ 5] local 192.168.40.104 port 53139 connected to 192.168.40.21 port 5201
```

[ ID]	Interval	Transfer	Bitrate	Total Datagrams
-------	----------	----------	---------	-----------------

[ 5]	0.00-2.00 sec	23.8 MBytes	99.9 Mb/s	17257
------	---------------	-------------	-----------	-------

[ 5]	2.00-4.00 sec	23.9 MBytes	100 Mb/s	17272
------	---------------	-------------	----------	-------

[ 5]	4.00-6.00 sec	23.8 MBytes	100 Mb/s	17265
------	---------------	-------------	----------	-------

[ 5]	6.00-8.00 sec	23.8 MBytes	100 Mb/s	17265
------	---------------	-------------	----------	-------

[ 5]	8.00-10.00 sec	23.8 MBytes	100 Mb/s	17266
------	----------------	-------------	----------	-------

[ ID]	Interval	Transfer	Bitrate	Jitter	Lost/Total Datagrams
-------	----------	----------	---------	--------	----------------------

[ 5]	0.00-10.00 sec	119 MBytes	100 Mb/s	0.000 ms	0/86325 (0%) sender
------	----------------	------------	----------	----------	---------------------

[ 5]	0.00-10.04 sec	119 MBytes	99.6 Mb/s	0.013 ms	0/86325 (0%) receiver
------	----------------	------------	-----------	----------	-----------------------





iperf Done.

The client finishes the test after 10 seconds and displays the above test result, which shows that UDP has no packet loss at the specified bandwidth of 100 Mbps. At the same time, the server also displays the test results as follows, and then continues to listen on port 5201 waiting for the client to connect:

```
Accepted connection from 192.168.40.104, port 46356
[ 5] local 192.168.40.21 port 5201 connected to 192.168.40.104 port 53139
[ ID] Interval           Transfer    Bitrate        Jitter    Lost/TOTAL  Datagrams
ms
[ 5]  0.00-1.00   sec    11.4 MBytes  96.0 Mbits/sec  0.009 ms   0/8289 (0%)
[ 5]  1.00-2.00   sec    11.9 MBytes  100 Mbits/sec   0.005 ms   0/8634 (0%)
[ 5]  2.00-3.00   sec    11.9 MBytes  99.9 Mbits/sec  0.044 ms   0/8620 (0%)
[ 5]  3.00-4.00   sec    11.9 MBytes  100 Mbits/sec   0.007 ms   0/8644 (0%)
[ 5]  4.00-5.00   sec    11.9 MBytes  100 Mbits/sec   0.024 ms   0/8631 (0%)
[ 5]  5.00-6.00   sec    11.9 MBytes  100 Mbits/sec   0.006 ms   0/8634 (0%)
[ 5]  6.00-7.00   sec    11.9 MBytes  100 Mbits/sec   0.010 ms   0/8632 (0%)
[ 5]  7.00-8.00   sec    11.9 MBytes  100 Mbits/sec   0.016 ms   0/8633 (0%)
[ 5]  8.00-9.00   sec    11.9 MBytes  100 Mbits/sec   0.010 ms   0/8630 (0%)
[ 5]  9.00-10.00  sec    11.9 MBytes  100 Mbits/sec   0.020 ms   0/8633 (0%)
[ 5] 10.00-10.04  sec     488 KBytes  99.4 Mbits/sec   0.013 ms   0/345 (0%)
-----
```





```
[ ID] Interval          Transfer      Bitrate      Jitter      Lost/Total Datagrams
ms
[ 5] 0.00-10.04 sec    119 MBytes   99.6 Mb/s    0.013 ms    0/86325
(0%) receiver

-----

Server listening on 5201
-----
```

The client modifies the `-b` parameter to continue to increase the specified UDP bandwidth, the maximum rate that the sender can achieve is the maximum bandwidth, the packet loss rate depends on the server-side CPU performance, the size of the network card buffer, the following method to send a fixed size packet to test the packet loss rate:

```
console:/ # iperf3 -u -c 192.168.40.21 -i 2 -t 10 -b 1000M
Connecting to host 192.168.40.21, port 5201
[ 5] local 192.168.40.104 port 49294 connected to 192.168.40.21 port 5201
[ ID] Interval          Transfer      Bitrate      Total Datagrams
[ 5] 0.00-2.00 sec     91.9 MBytes   385 Mb/s     66525
[ 5] 2.00-4.00 sec     93.1 MBytes   390 Mb/s     67415
[ 5] 4.00-6.00 sec     93.9 MBytes   394 Mb/s     67974
[ 5] 6.00-8.00 sec     94.0 MBytes   394 Mb/s     68096
[ 5] 8.00-10.00 sec    94.4 MBytes   396 Mb/s     68336
-----
[ ID] Interval          Transfer      Bitrate      Jitter      Lost/Total Datagrams
ms
[ 5] 0.00-10.00 sec    467 MBytes   392 Mb/s     0.000 ms    0/338346
(0%) sender
[ 5] 0.00-10.04 sec    467 MBytes   390 Mb/s     0.012 ms    0/338346
(0%) receiver

iperf Done.
```

iperf3 also has many parameters that can be configured during the test, and users can make targeted adjustments to the test according to the actual application





needs. For example, you can increase the value of the -t parameter for a long time stress test, or specify the -P parameter for a concurrent stress test of multiple connections. For more information about iperf3 testing, please refer to:

<https://iperf.fr/iperf-doc.php#3doc>.

## 4.4. WIFI

This section mainly introduces the configuration and use of Wi-Fi under android, usually the Wi-Fi module can support two working modes, STA mode and AP mode, some devices also support STA and AP mode at the same time. STA mode allows the device to connect to an external Wi-Fi hotspot, and AP mode turns the device into a Wi-Fi hotspot for other devices to connect to. MYD- LT527 uses the AP6252 Wi-Fi and Bluetooth 2-in-1 module, currently it does not support STA and AP mode at the same time. The LT527 uses the AP6252 Wi-Fi and Bluetooth 2-in-1 module, which does not support STA and AP at the same time.

### 1) WI-FI connection test

Draw your finger down the status bar on the screen and click on "Internet"

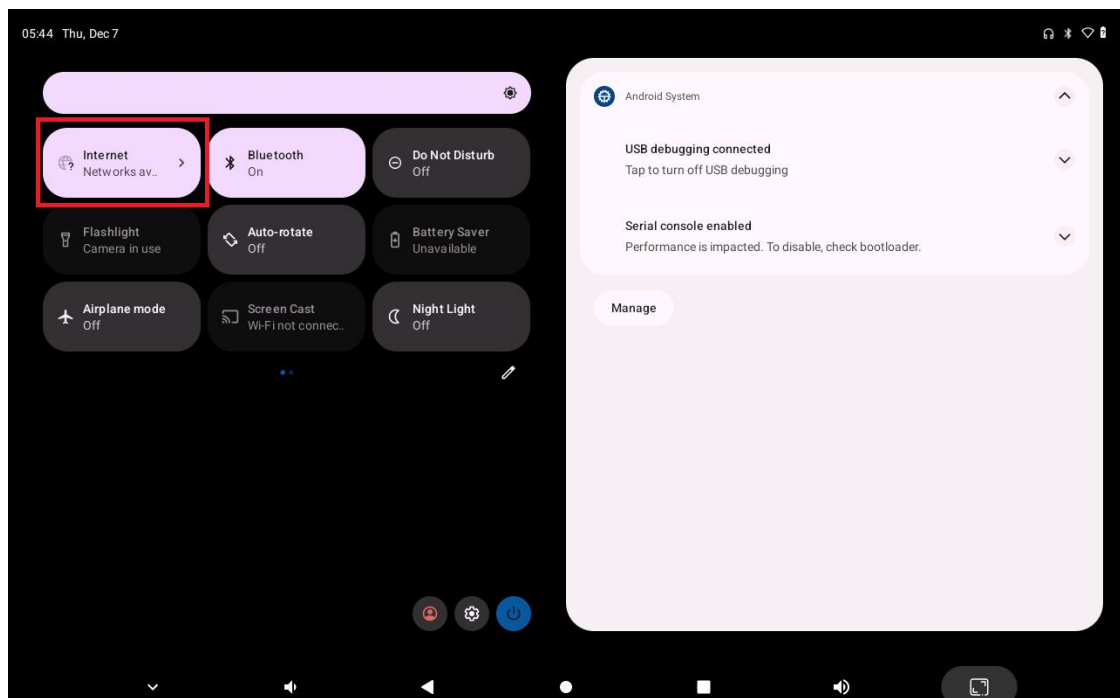


Figure 4-3 status bar

You can see the scanned wifi devices, select the wifi you need to connect to



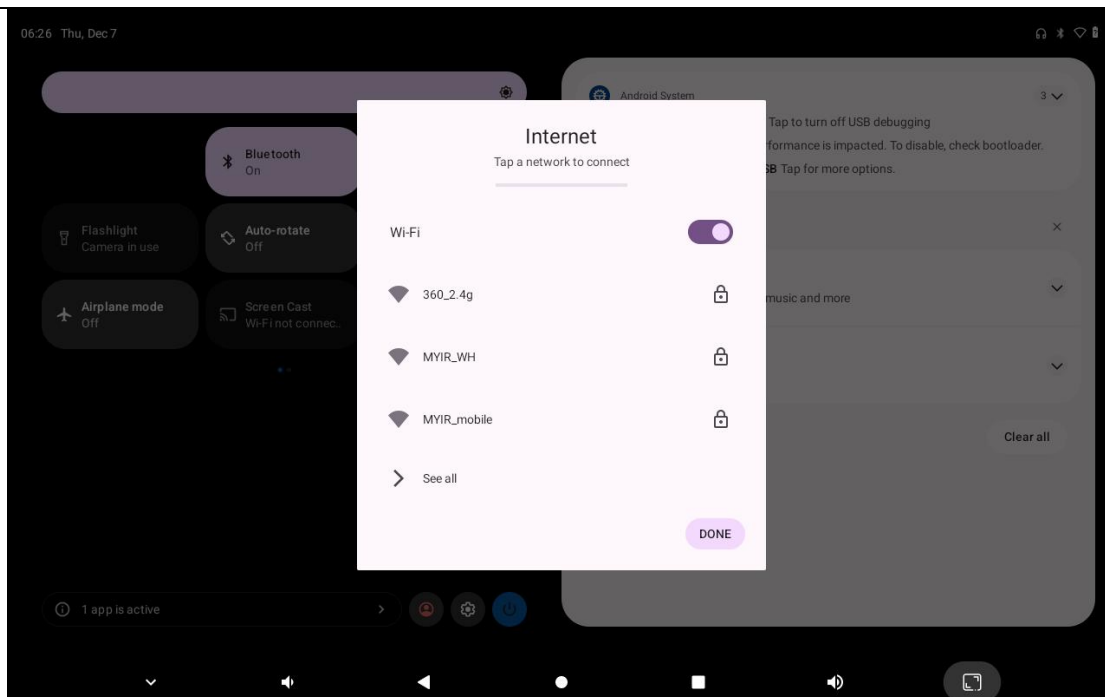


Figure 4-4 WIFI Hot List

Then enter your password.

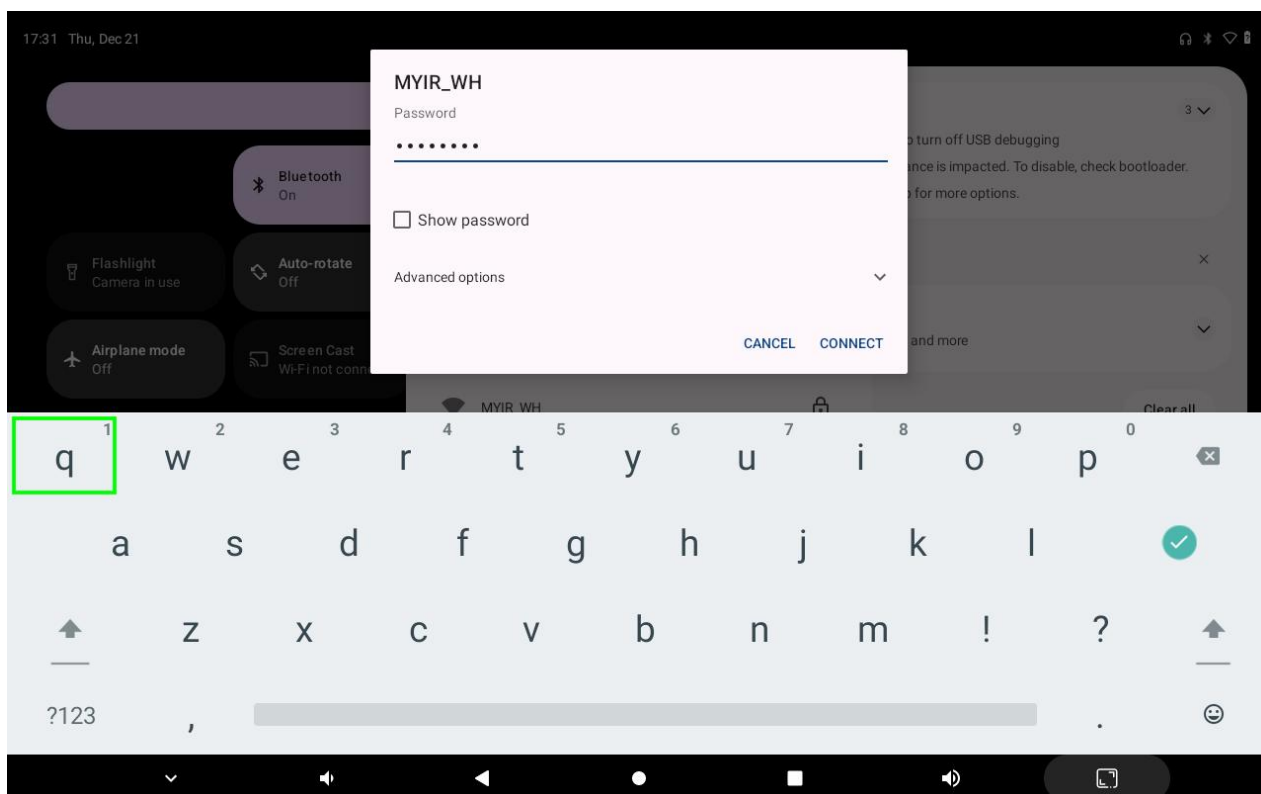


Figure 4-5 WIFI password

Waiting for connection



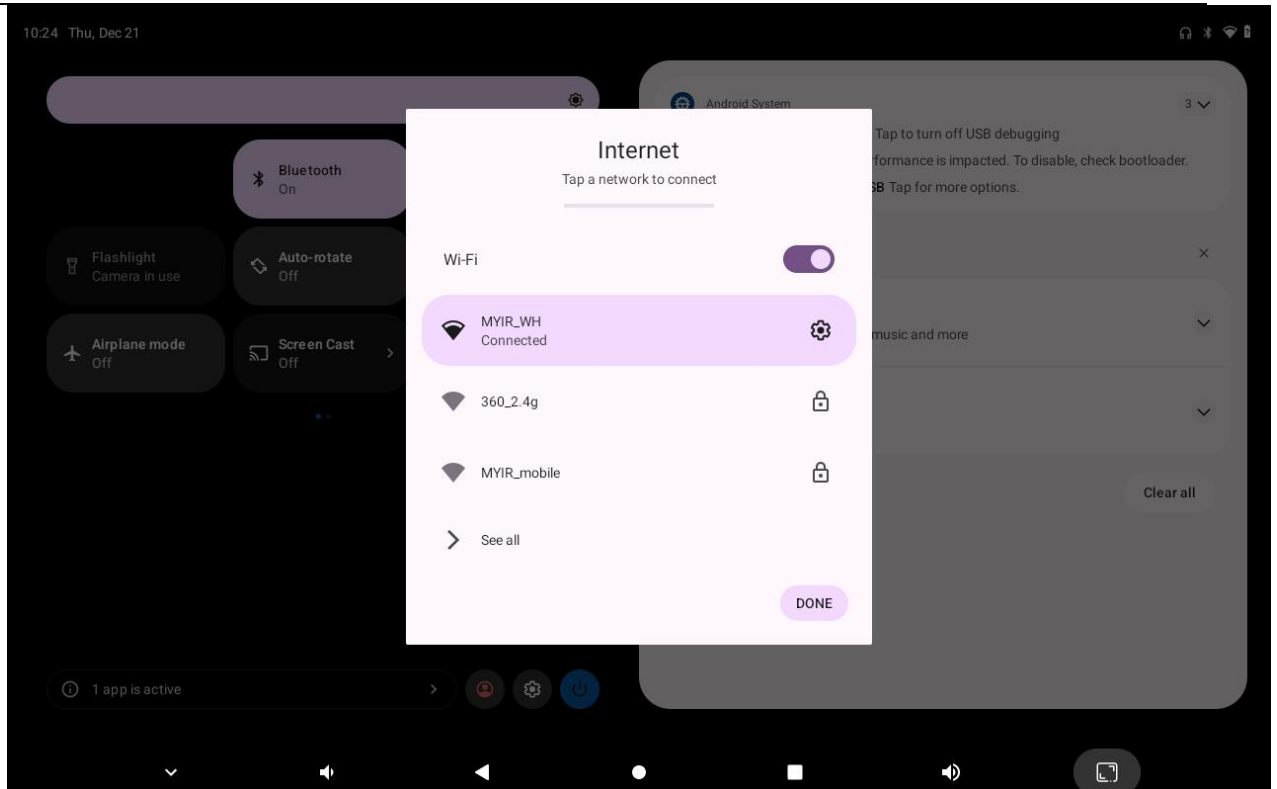


Figure 4-6 WIFI connection

Find WI-FI in "Setting" and click the icon in the red box below.

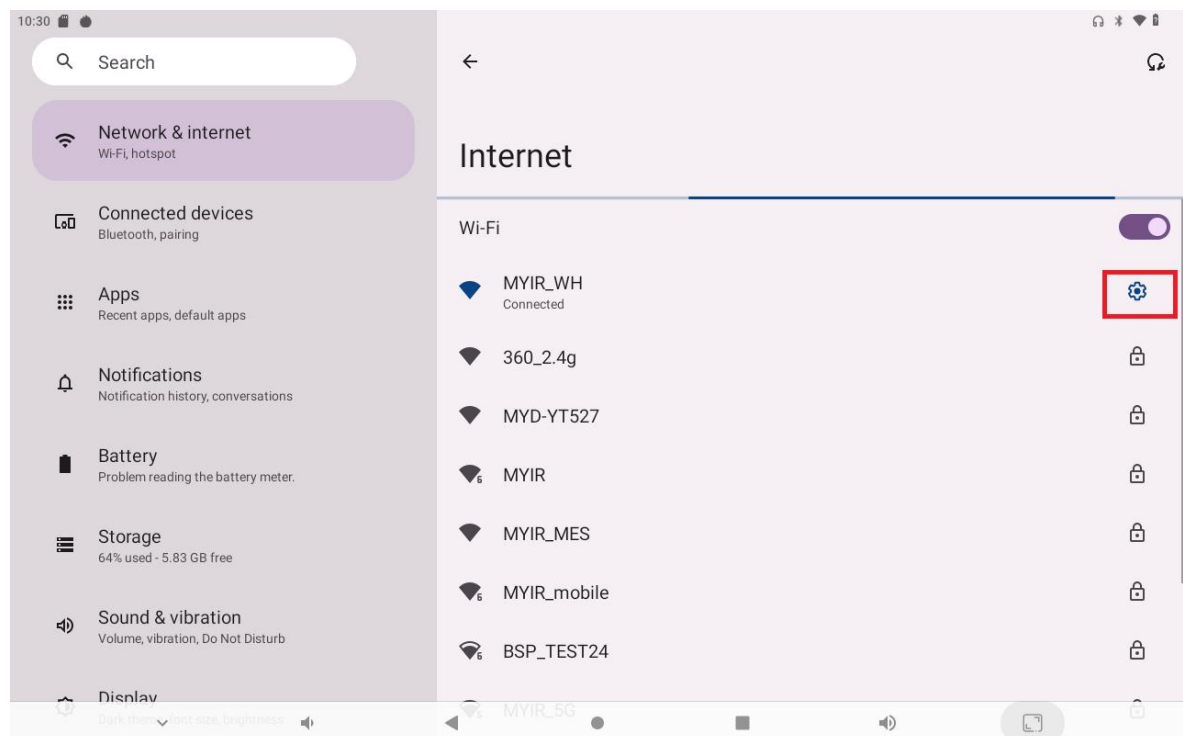


Figure 4-7 WIFI Setting interface

You can see the IP address of the WI-FI



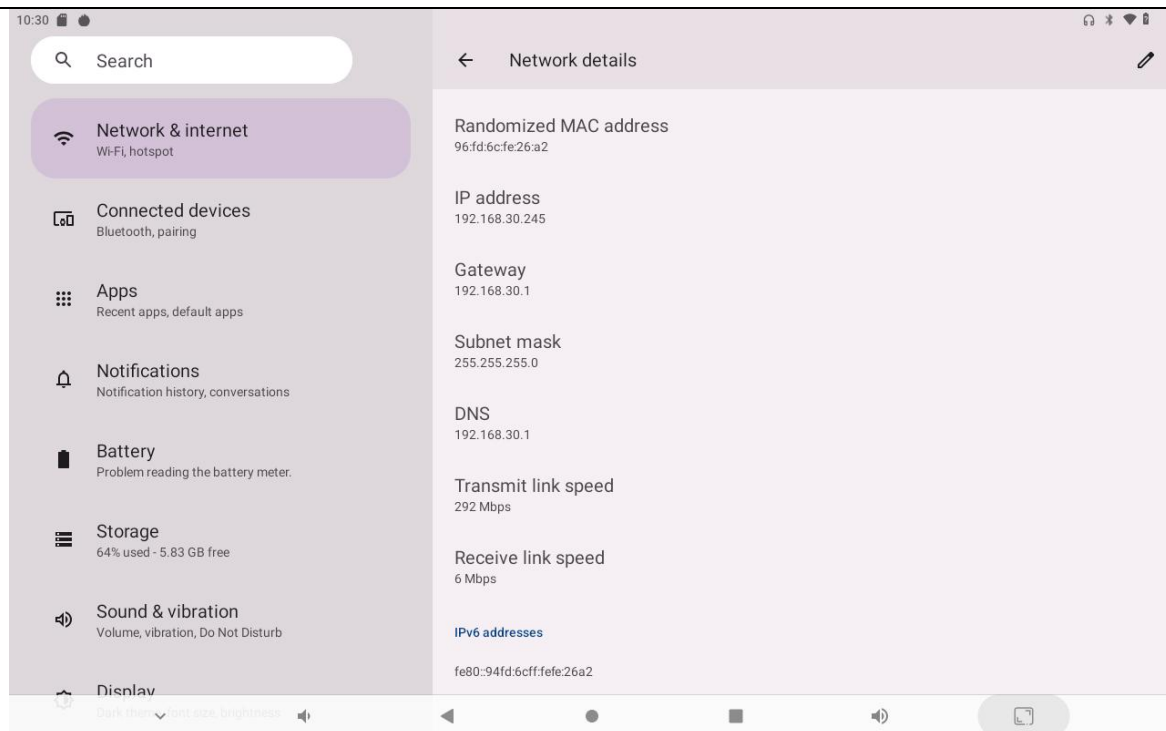


Figure 4-8 WIFI IPAddresses



## 5. multimedia applications

### 5.1. Camera

#### 1) mipi 5640 (CSI1) Camera Test

Scroll up the application interface, open the DVR application, select "Single Camera" camera, "720P@25" CSI1 corresponds to /dev/video0 node, so select "0" channel, and then confirm.

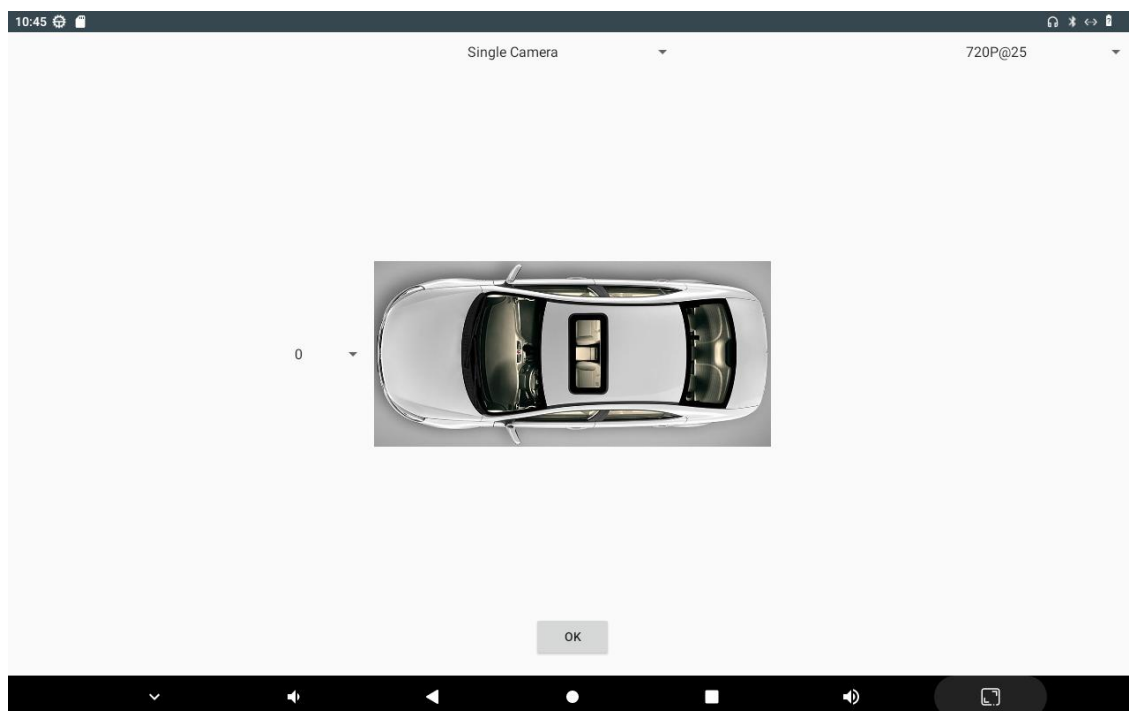


Figure 6-1 DVR application interface



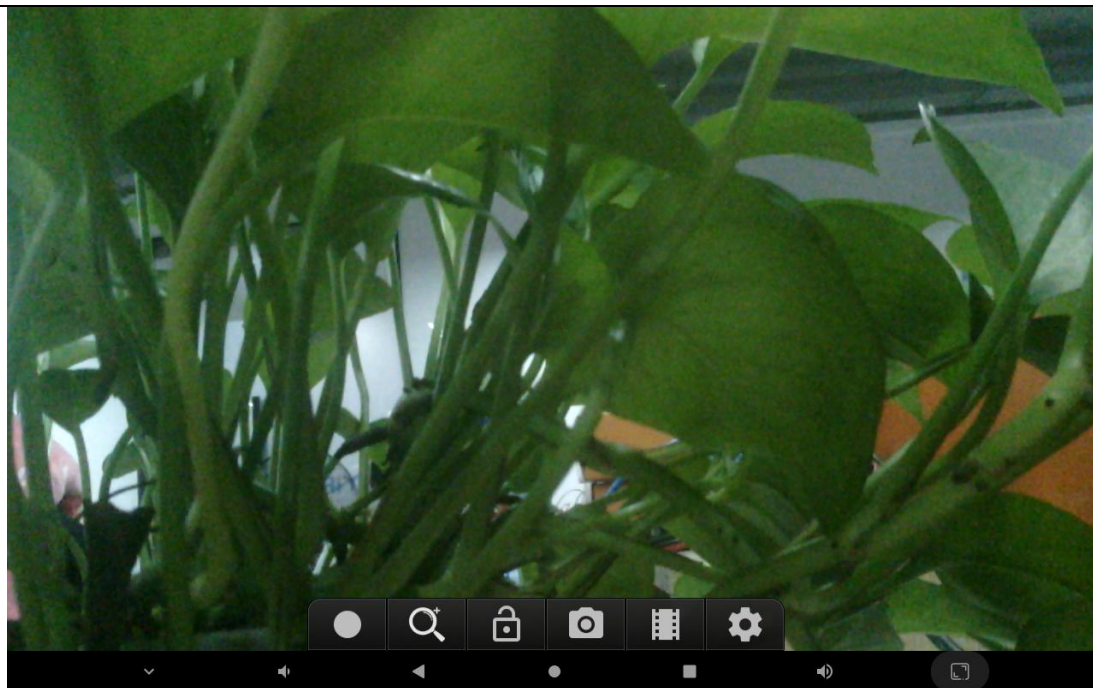


Figure 6-2 CSI1 previews

click on a button  videotape, click on a button  take a photo.

## 2) mipi 5640 (CSI2) Camera Test

Start with the same steps as above, select the single independent camera, CSI2 corresponds to /dev/video16 node, so select the "16" channel, and then confirm.

The operation of the photo and video function is the same as that of CSI0.



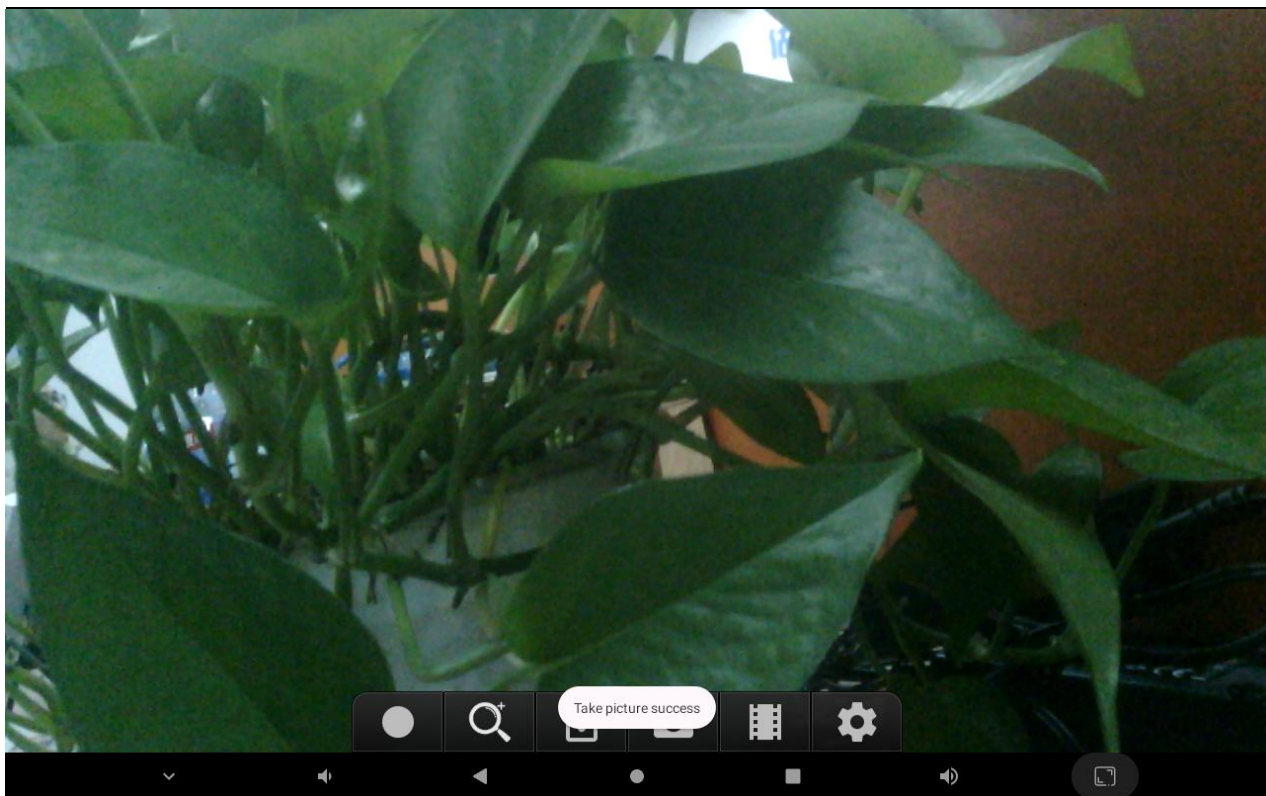


Figure 6-3 CSI2 previews







## 6. References

- Allwinner Development Community

<https://www.aw-ol.com>

- Allwinner Official Website

<https://www.allwinnertech.com/>







# Appendix A

## Warranty & Technical Support Services

**MYIR Electronics Limited** is a global provider of ARM hardware and software tools, design solutions for embedded applications. We support our customers in a wide range of services to accelerate your time to market.

MYIR is an ARM Connected Community Member and work closely with ARM and many semiconductor vendors. We sell products ranging from board level products such as development boards, single board computers and CPU modules to help with your evaluation, prototype, and system integration or creating your own applications. Our products are used widely in industrial control, medical devices, consumer electronic, telecommunication systems, Human Machine Interface (HMI) and more other embedded applications. MYIR has an experienced team and provides custom design services based on ARM processors to help customers make your idea a reality.

The contents below introduce to customers the warranty and technical support services provided by MYIR as well as the matters needing attention in using MYIR' s products.

### Service Guarantee

MYIR regards the product quality as the life of an enterprise. We strictly check and control the core board design, the procurement of components, production control, product testing, packaging, shipping and other aspects and strive to provide products with best quality to customers. We believe that only quality products and excellent services can ensure the long-term cooperation and mutual benefit.

### Price

MYIR insists on providing customers with the most valuable products. We do not pursue excess profits which we think only for short-time cooperation. Instead, we hope to establish





long-term cooperation and win-win business with customers. So we will offer reasonable prices in the hope of making the business greater with the customers together hand in hand.

### **Delivery Time**

MYIR will always keep a certain stock for its regular products. If your order quantity is less than the amount of inventory, the delivery time would be within three days; if your order quantity is greater than the number of inventory, the delivery time would be always four to six weeks. If for any urgent delivery, we can negotiate with customer and try to supply the goods in advance.

### **Technical Support**

MYIR has a professional technical support team. Customer can contact us by email (support@myirtech.com), we will try to reply you within 48 hours. For mass production and customized products, we will specify person to follow the case and ensure the smooth production.

### **After-sale Service**

MYIR offers one year free technical support and after-sales maintenance service from the purchase date. The service covers:

#### **Technical support service**

MYIR offers technical support for the hardware and software materials which have provided to customers:

- To help customers compile and run the source code we offer;
- To help customers solve problems occurred during operations if users follow the user manual documents;
- To judge whether the failure exists;
- To provide free software upgrading service.





However, the following situations are not included in the scope of our free technical support service:

- Hardware or software problems occurred during customers' own development;
- Problems occurred when customers compile or run the OS which is tailored by themselves;
- Problems occurred during customers' own applications development;
- Problems occurred during the modification of MYIR's software source code.

### **After-sales maintenance service**

The products except LCD, which are not used properly, will take the twelve months free maintenance service since the purchase date. But following situations are not included in the scope of our free maintenance service:

- The warranty period is expired;
- The customer cannot provide proof-of-purchase or the product has no serial number;
- The customer has not followed the instruction of the manual which has caused the damage the product;
- Due to the natural disasters (unexpected matters), or natural attrition of the components, or unexpected matters leads the defects of appearance/function;
- Due to the power supply, bump, leaking of the roof, pets, moist, impurities into the boards, all those reasons which have caused the damage of the products or defects of appearance;
- Due to unauthorized weld or dismantle parts or repair the products which has caused the damage of the products or defects of appearance;
- Due to unauthorized installation of the software, system or incorrect configuration or computer virus which has caused the damage of products.





## Warm tips

1. MYIR does not supply maintenance service to LCD. We suggest the customer first check the LCD when receiving the goods. In case the LCD cannot run or no display, customer should contact MYIR within 7 business days from the moment get the goods.
2. Please do not use finger nails or hard sharp object to touch the surface of the LCD.
3. MYIR suggests user purchasing a piece of special wiper to wipe the LCD after long time use, please avoid clean the surface with fingers or hands to leave fingerprint.
4. Do not clean the surface of the screen with chemicals.
5. Please read through the product user manual before you using MYIR' s products.
6. For any maintenance service, customers should communicate with MYIR to confirm the issue first. MYIR' s support team will judge the failure to see if the goods need to be returned for repair service, we will issue you RMA number for return maintenance service after confirmation.

## Maintenance period and charges

- MYIR will test the products within three days after receipt of the returned goods and inform customer the testing result. Then we will arrange shipment within one week for the repaired goods to the customer. For any special failure, we will negotiate with customers to confirm the maintenance period.
- For products within warranty period and caused by quality problem, MYIR offers free maintenance service; for products within warranty period but out of free maintenance service scope, MYIR provides maintenance service but shall charge some basic material cost; for products out of warranty period, MYIR provides maintenance service but shall charge some basic material cost and handling fee.

## Shipping cost





During the warranty period, the shipping cost which delivered to MYIR should be responsible by user; MYIR will pay for the return shipping cost to users when the product is repaired. If the warranty period is expired, all the shipping cost will be responsible by users.

### **Products Life Cycle**

MYIR will always select mainstream chips for our design, thus to ensure at least ten years continuous supply; if meeting some main chip stopping production, we will inform customers in time and assist customers with products updating and upgrading.

### **Value-added Services**

1. MYIR provides services of driver development base on MYIR' s products, like serial port, USB, Ethernet, LCD, etc.
2. MYIR provides the services of OS porting, BSP drivers' development, API software development, etc.
3. MYIR provides other products supporting services like power adapter, LCD panel, etc.
4. ODM/OEM services.

### **MYIR Electronics Limited**

Room 04, 6th Floor, Building No.2, Fada Road,  
Yunli Inteiligent Park, Bantian, Longgang District.

Support Email: support@myirtech.com

Sales Email: sales@myirtech.com

Phone: +86-755-22984836

Fax: +86-755-25532724

Website: www.myirtech.com

