MYD-YT507H Android System Evaluation Guide

File Status: [ ] Draft [√] Released	**FILE ID:**	MYIR-MYD-YT507H-SW-EG-EN-A4.9.170
	CURRENT VERSION:	V1. 0
	AUTHOR:	Nico
	DATE CREATED:	2022.09.24
	LAST UPDATED:	2022.09.24

Revision History

VERSION	AUTHOR	PARTICIPATOR	DATE	REMARK
V1.0	Nico		2022.09.24	Initial Version

CONTENT

MYD-YT507H Android System Evaluation Guide - 1 -

Revision History - 2 -
CONTENT - 3 -

1. overview - 5 -

1.1. Hardware resources - 5 -
1.2. Software resources - 5 -
1.3. Document resources - 6 -
1.4. Environment preparation - 6 -
1.5. Introduction to the basic interface - 6 -

2. Core resources - 9 -

2.1. CPU - 9 -
2.2. GPU - 10 -
2.3. Memory - 12 -
2.4. eMMC - 14 -
2.5. RTC - 16 -
2.6. Watchdog - 18 -
2.7. PMIC - 19 -
2.8. GPIO - 21 -

3. Basic peripheral interface - 23 -

3.1. LED lights - 23 -
3.2. Key - 24 -
3.3. USB - 26 -
3.4. Micro SD card - 27 -
3.5. ADC - 29 -
3.6. Display - 30 -
3.7. Touch Panel - 36 -
3.8. Ethernet - 38 -

3.9. RS232 and RS485 test - 42 -

4. Expand peripheral interfaces - 44 -

4.1. MY-WF005S module - 44 -
4.2. 4G module - 49 -

5. Web applications - 51 -

5.1. PING - 51 -
5.2. Iptables - 52 -
5.3. iperf - 54 -

6. Multimedia applications - 60 -

6.1. Camera - 60 -
6.2. Video test - 62 -
6.3. Audio test - 63 -
6.4. Miracast casting - 65 -

7. Resources - 67 -

Appendix A - 68 -

Warranty & Technical Support Services - 68 -

1. overview

The Android Software Evaluation Guide introduces the test steps and evaluation methods for running core resources and peripheral resources under the open source Android system on MYIR development board. This article can be used as a preliminary evaluation guide or as a test guide for general system development.

1.1. Hardware resources

MYD-YT507H board is composed of core board MYC-YT507H and bottom plate MYB-YT507H, and the core plate and bottom plate are welded with stamp holes. In addition, MYIR provides a wealth of software resources and documentation. For detailed configuration parameters on the hardware part, please refer to the MYD-YT507H Product Guide. At the same time, some accessories will be used during the evaluation test, see the list below.

Table 1-1 Optional module

Accessories	Interface mode	Description and Links
Camera	MIPI/ parallel	MIPI camera: http://www.myir-tech.com/product/my_cam003m.htm Parallel camera: http://www.myir-tech.com/product/my_cam011b.htm
LCD screen	LVDS interface	7 inch LVDS screen: http://www.myir-tech.com/product/my-lvds070c.htm
Expansion board module	Raspberry Pi interface	MY-WiredCom： http://www.myir-tech.com/product/my-wiredcom.htm
Wifi module	SDIO interface	MY-WF005S： http://www.myir-tech.com/product/MY-WF005S.htm

1.2. Software resources

The BSP of the MYD-YT507H development board is based on the Linux BSP porting and modification of the official open source community version of Allwinner. Bootloader, Kernel and file system parts of the software resources are all open in the form of source code, please see the "MYD-YT507H_Android SDK" for details Release Notes.

1.3. Document resources

According to the different stages of user use of the development board, the SDK contains different categories of documents and manuals such as documentation, release notes, evaluation guides, development guides, application notes, and common questions and answers for each stage. For a list of documents, see Table 2-4 of the MYD-YT507H_Android SDK Release Notes.

1.4. Environment preparation

Before starting to evaluate the board software, you need to make some necessary preparations and configure some basic environment for the development board, including correct hardware wiring, configuration and debugging serial port, setting up startup, and other steps. For detailed steps, please refer to the MYD-YT507 Quick Start Guide. The next sections focus on evaluating and testing the system's hardware resources and interfaces, as well as software functionality. Mainly with the help of some commonly used tools and commands under Android, as well as self-developed applications for testing. The software evaluation guide is divided into multiple parts to describe, including: core resources, peripheral resources, network applications, multimedia applications and other categories. The following chapters will provide a comprehensive explanation of each part and describe in detail the specific evaluation methods and steps of each part of the resource.

1.5. Introduction to the basic interface

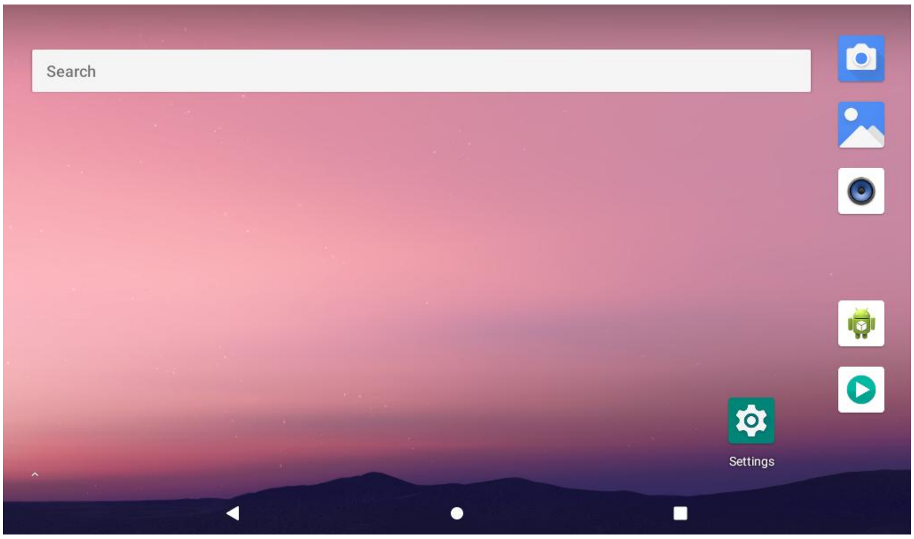


Figure 1-1 Main interface of the system

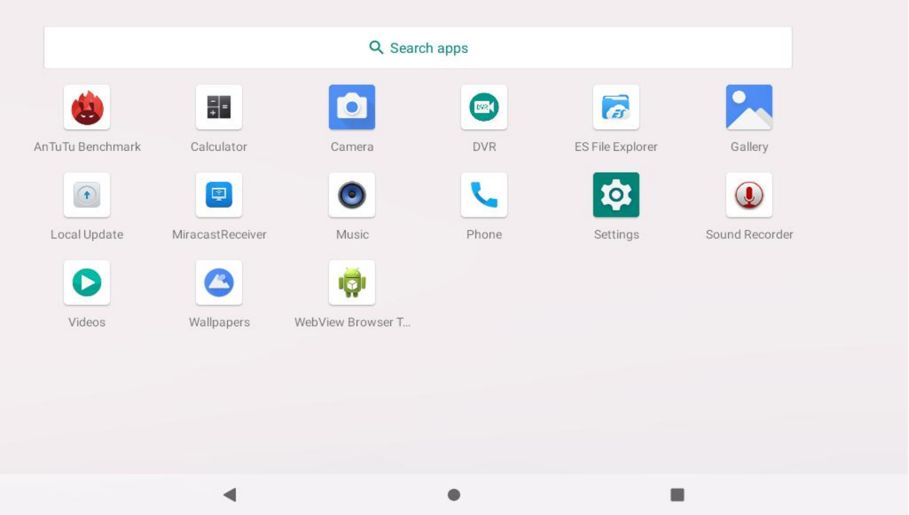


Figure 1-2 Application interface

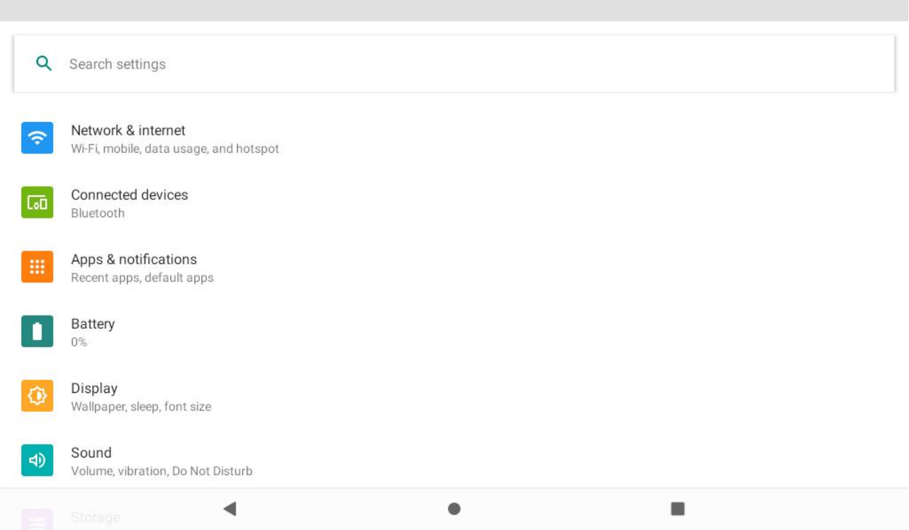

Search settings
Network & internet
Wi-Fi, mobile, data usage, and hotspot
Connected devices
Bluetooth
Apps & notifications
Recent apps, default apps
Battery
0%
Display
Wallpaper, sleep, font size
Sound
Volume, vibration, Do Not Disturb
Storage

Figure 1-3 Setup interface

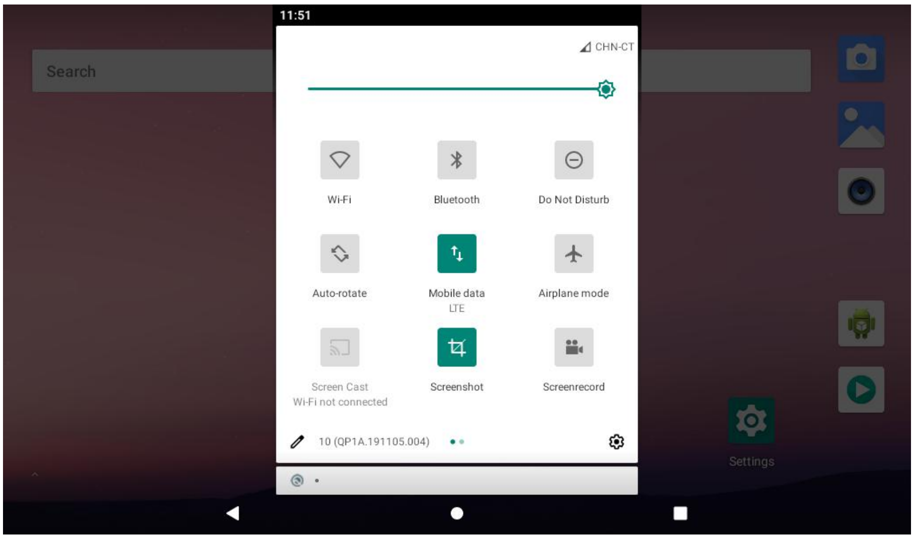
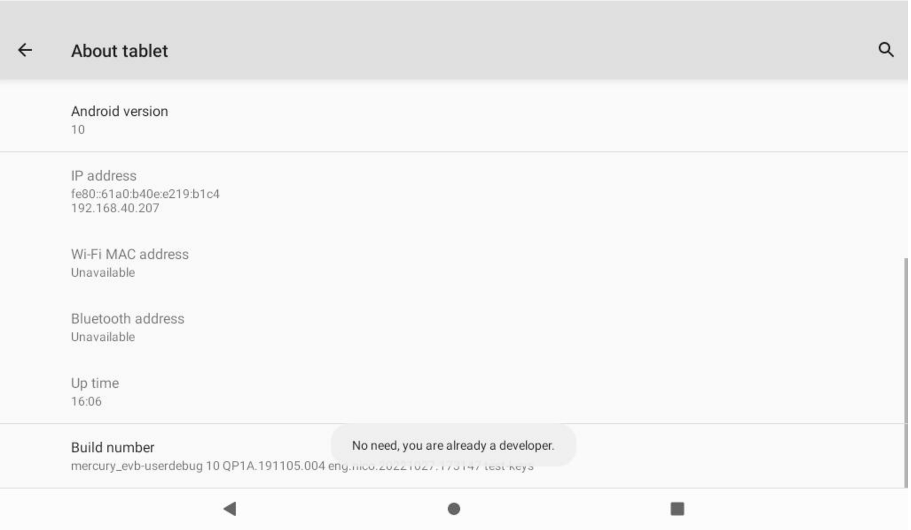


Figure 1-4 Status bar

Enter developer mode, first enter the settings interface, then enter About Tablet, find the version number, click the version number three times to enter developer mode.


About tablet

Android version
10

IP address
fe80::61a0:b40e:e219:b1c4
192.168.40.207

Wi-Fi MAC address
Unavailable

Bluetooth address
Unavailable

Up time
16:06

Build number
mercury_evb-userdebug 10 QP1A.191105.004 eng.mco.20221027.173147 test-keys

No need, you are already a developer.

Figure 1-5 Developer mode

2. Core resources

In the Android system, the proc virtual file system is provided to query the parameters of various core resources and some common tools to evaluate the performance of resources. In order to make the following more convenient and intuitive test, we will show it on the Android graphical interface, before the test we download the AnTuTu evaluation software, as for how to install the software under the Android system, please refer to MYD-YT507H_ Android System Development Guide. The following will specifically read and test the parameters of core resources such as CPU, memory, eMMC, and RTC.

2.1. CPU

1). View CPU information

First swipe up on the main interface to enter the application interface, open the AnTuTu software, click My Device, and slide down to find the CPU information.

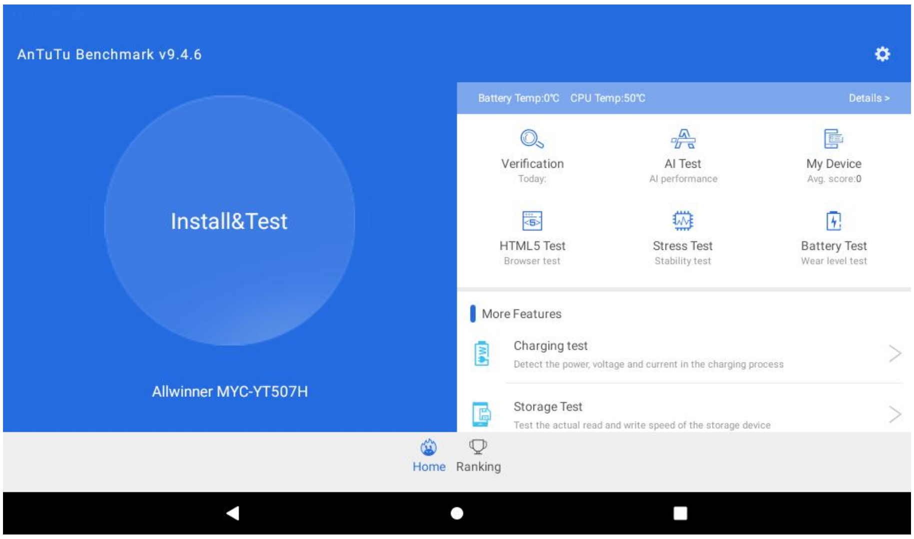
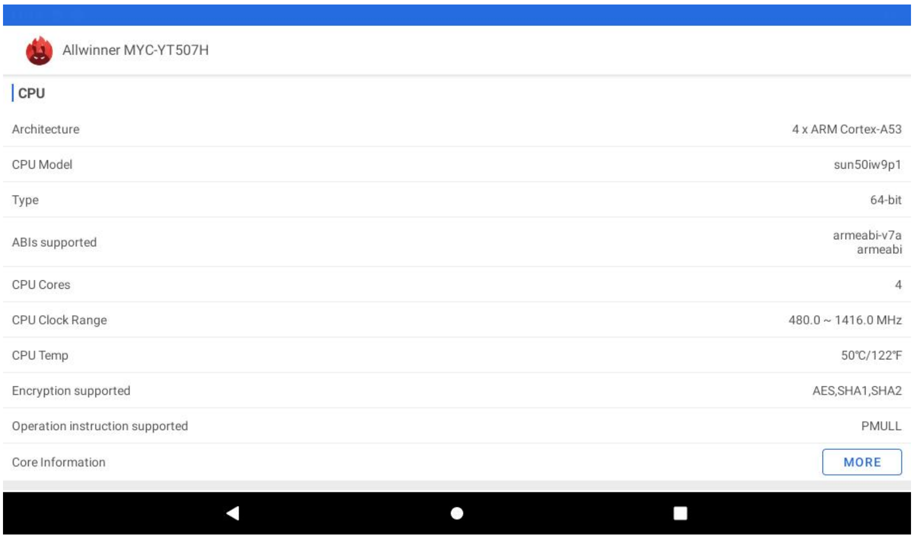


Figure 2-1 AnTuTu function homepage



Allwinner MYC-YT507H

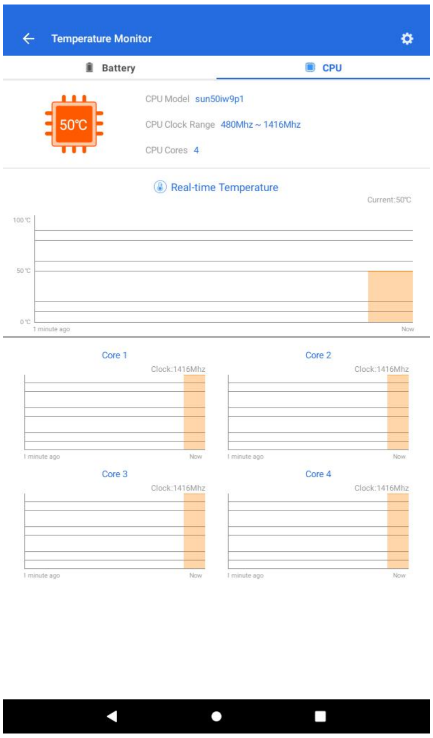
CPU

Architecture	4 x ARM Cortex-A53
CPU Model	sun50iw9p1
Type	64-bit
ABIs supported	armeabi-v7a armeabi
CPU Cores	4
CPU Clock Range	480.0 ~ 1416.0 MHz
CPU Temp	50℃/122℉
Encryption supported	AES,SHA1,SHA2
Operation instruction supported	PMULL
Core Information	MORE

Figure 2-2 CPU information

2). View CPU usage and temperature

Click View Details on the home page of AnTuTu function and select CPU options to see the CPU usage and temperature.



CPU Model sun50iw9p1

CPU Clock Range 480Mhz ~ 1416Mhz

CPU Cores 4

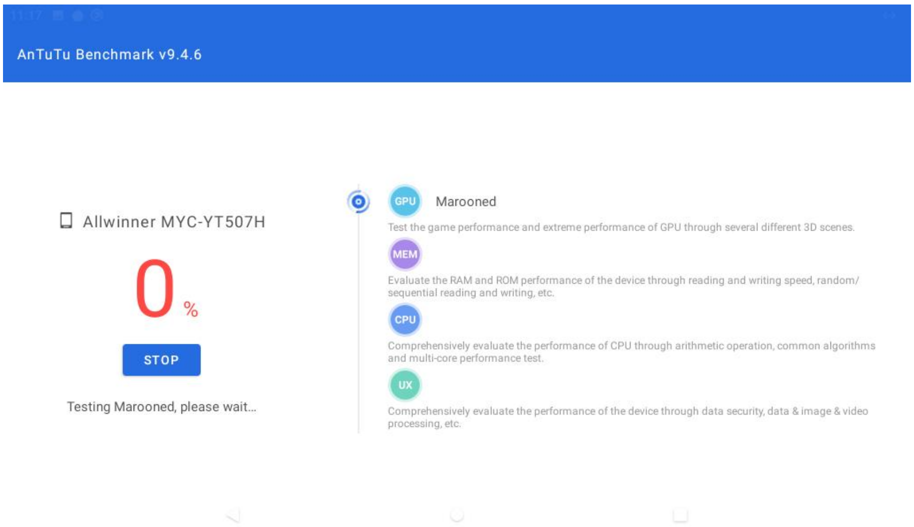
Figure 2-3 CPU and temperature

2.2. GPU

Graphics Processing Unit (GPU), also known as display core, vision processor, display chip, is a kind of specialized in personal computers, workstations, game consoles and some mobile devices (such as tablets Smartphones, etc.), microprocessors that do image and graphics-related operations. As the core of the display system, the graphics processor has powerful data computing capabilities, and realizes two-dimensional/three-dimensional (2D/3D) in the form of hardware accelerators. Functions such as graphics processing, image processing, and display control.

MYD-YT507H chip internal GPU module, support 2D, 3D acceleration, OpenGL ES1.1, 2.0, 3.0, 3.1, Open CL 1. 2, and QT graphics system.

To test the GPU, you need to download AnTuTu Review 3DLite, and then open the app to test the GPU



AnTuTu Benchmark v9.4.6

Allwinner MYC-YT507H

0 %

STOP

Testing Marooned, please wait...

GPU Marooned

Test the game performance and extreme performance of GPU through several different 3D scenes.

MEM

Evaluate the RAM and ROM performance of the device through reading and writing speed, random/sequential reading and writing, etc.

CPU

Comprehensively evaluate the performance of CPU through arithmetic operation, common algorithms and multi-core performance test.

UX

Comprehensively evaluate the performance of the device through data security, data & image & video processing, etc.

Figure 2-4 AnTuTu evaluation 3DLite function homepage

Figure 2-5 GPU test

2.3. Memory

The MYD-YT507H memory is divided into 1GByte and 2GByte versions, and the system will divide 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 allocated space for user mode.

1). View memory information

The following command allows you to get information about the memory parameters.

```
console:/ # cat /proc/meminfo
MemTotal:          2039744 kB
MemFree:           1040964 kB
MemAvailable:      1627628 kB
Buffers:              3096 kB
Cached:             623188 kB
SwapCached:              0 kB
Active:             413244 kB
Inactive:           389144 kB
Active(anon):       184752 kB
Inactive(anon):       2356 kB
Active(file):       228492 kB
Inactive(file):     386788 kB
Unevictable:          9040 kB
Mlocked:              9040 kB
SwapTotal:         1529804 kB
SwapFree:          1529804 kB
Dirty:                   0 kB
Writeback:               0 kB
AnonPages:          185256 kB
Mapped:             293116 kB
Shmem:                2532 kB
Slab:                59300 kB
SReclaimable:        20408 kB
```

```
SUnreclaim:          38892 kB
KernelStack:         10320 kB
PageTables:          10616 kB
CommitLimit:       2549676 kB
Committed_AS:      5415512 kB
VmallocTotal:    263061440 kB
CmaTotal:            12288 kB
CmaFree:              8976 kB
```

- MemTotal: All available RAM size, physical memory minus reserved bits and kernel usage
- MemFree：LowFree+HighFree
- Buffers: The size used to cache block devices
- Cached: The buffer size of the file
- SwapCached: Memory that has been swapped out. I/O related
- Active: Memory that is frequently (recently) used
- Inactive: Memory that has been used infrequently recently

2). Gets the memory usage

The following command allows you to view the memory usage.

```
console:/ #  free -m
                 total       used       free     shared    buffers
Mem:              1991        976       1015          2          3
-/+ buffers/cache:            973       1018
```

- total: The total amount of memory
- used: The amount of memory used
- free: The amount of memory that can be used

3). Memory stress test

Given the size and number of times of test memory, the system's existing memory can be stress-tested. You can use the system tool memtester for testing, such as specifying the memory size of 512MB, the number of tests is 10, and the test command is "memtester 512M 10".

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

```
console:/ #  memtester 512M
```

```
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 0xfffff000
want 512MB (536870912 bytes)
got  512MB (536870912 bytes), trying mlock ... locked.
Loop 1:
Loop 1/1:
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 focuses on eMMC testing and is suitable for development boards configured with eMMC memory. eMMC is a data storage device that includes a

MultiMediaCard (MMC) interface, 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

Slide up the application interface and open the ES File Explorer, you can see the capacity of eMMC, and you can also see the capacity usage.

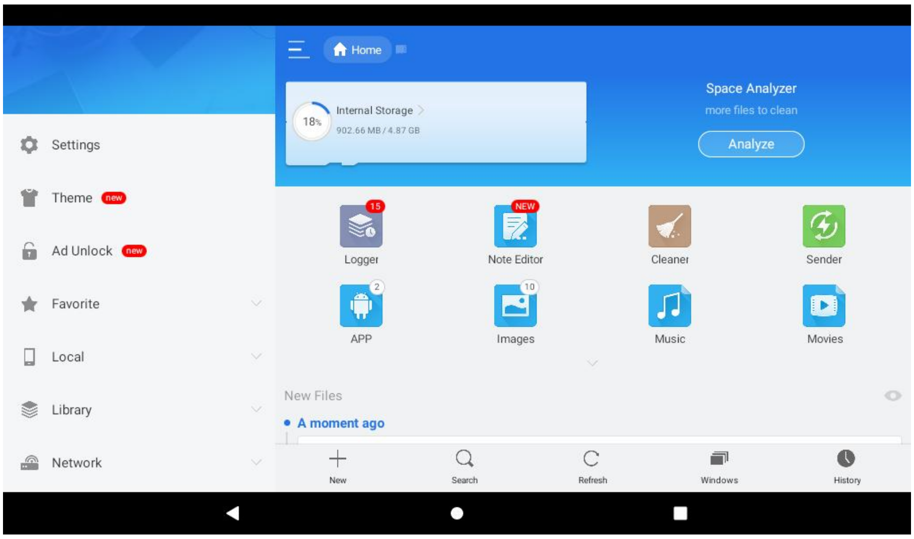


Figure 2-6 eMMC capacity

2). View the eMMC partition information

Use the following commands to query the eMMC partition information, usage, mount directory, and other information.

```
console:/ # df -h
Filesystem              Size  Used  Avail  Use%  Mounted on
tmpfs                   996M  604K  995M    1%  /dev
tmpfs                   996M     0  996M    0%  /mnt
tmpfs                   996M     0  996M    0%  /apex
/dev/block/dm-0          591M  589M  1.7M  100%  /
/dev/block/dm-3          510M  114M  396M   23%  /mnt/scratch
overlay                 510M  114M  396M   23%  /system
overlay                 510M  114M  396M   23%  /vendor
overlay                 510M  114M  396M   23%  /product
/dev/block/mmcblk0p18 4.8G  510M  4.3G   11%  /data
/dev/block/mmcblk0p7  614M  512K  613M    1%  /cache
```

```
/dev/block/mmcblk0p17   16M       0    16M    0%  /Reserve0
/data/media             4.8G   510M   4.3G   11%  /mnt/runtime/default/emulated
tmpfs                   490.1M         0      490.1M    0%  /dev/ shm
```

- /dev/mmcblk0p4 : Root file system, mounted to the root directory.
- tmpfs: Memory virtual file system, mounted to different directories.
- devtmpfs: Used for system creation dev.
- /dev/mmcblk0p8 : Available for user use partitions

3). Performance testing of eMMC

The performance test mainly tests the read and write speed of eMMC files under the Linux system, and generally combines the time and dd dual commands to test.

- **Write file test**

```
console:/data # time dd if=/dev/zero of=tempfile bs=1M count=100 conv=fsy
nc
100+0 records in
100+0 records out
104857600 bytes (100 M) copied, 3.728919 s, 27 M/s
```

Here we can get the eMMC write speed of 27 MB/s.

- **Read file test (to eliminate the impact of the cache, clear the cache first)**

```
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, 2.268299 s, 44 M/s
```

It can be seen that the speed of direct reading from disk is 44MB/s.

2.5. RTC

RTC (Real-time clock) itself is a clock, used to record the real time, when the software system shutdown retains the system time and continues to time, the system is turned on again after the time is synchronized into the software system. The MYD-YT507H has an internal RTC, as well as an external RTC (RX8025), if the actual product does not have very high RTC power requirements , RTC test is usually carried out with the common hwclock and date commands of Linux systems, the following test writes the system time to RTC, The RTC time is read and set to the system time and tested for time power-down hold.

1). View the system RTC device

```
console:/ $  ls /dev/rtc* -al
crw-r----- 1 system system 253, 0 1970-01-01 08:00 /dev/rtc0
crw------- 1 root   root   253, 1 1970-01-01 08:00 /dev/rtc1
```

2). Set the system time

Set the system time to Wed Sep 21 13:42:00 UTC 2022:

```
console:/ # busybox date 092113422022.00
Wed Sep 21 13:42:00 UTC 2022
```

- **Write the system time to the RTC**

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

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

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

```
console:/ # busybox hwclock -r
Wed Sep 21 13:42:11 2022 0.000000 seconds
```

- **Power-down hold RTC time**

Turn off the board and disconnect the power, and after a few minutes or so, power it on again. View RTC time and system time:

```
console:/ # busybox hwclock -r
Wed Sep 21 13:46:48 2022 0.000000 seconds
```

The RTC time and system time checked after rebooting are increased by about 20 minutes compared to the previous setting, indicating that the RTC is working normally. If you need to test the accuracy of the RTC in detail, you can extend the power-off time such as 24 hours, and test the difference between the RTC time and the standard time.

- **Synchronize the system time with the RTC time**

```
console:/ # busybox hwclock -s
console:/ # date
Wed Sep 21 13:48:51 CST 2022
```

2.6. Watchdog

The Linux kernel includes a Watchdog subsystem, and the hardware design process can generally use the internal watchdog timer or an external watchdog chip to implement the Watchdog function for monitoring the system operation. The system can be automatically reset when the system has an abnormal condition and cannot feed the dog. Allwinner T507 series chip has 3 watchdogs. There are two independent watchdogs (IWDG1 and IWDG2) on the MPU side. IWDG1 is located on the security bus and can only be used by security applications in a secure environment. The independent watchdogs (IWDG1 and IWDG2) are clocked by the low-speed clock (LSI) and thus remain active even if the master clock fails. This makes them best suited for applications that require the watchdog to run as a completely independent process outside of the main application, and IWDGs are best suited for recovering from unexpected software or hardware failures. interrupt controller. It allows the application to determine WWDG1, which is clocked by APB1 and provides the reset and early interrupt signals. The early interrupt provided by WWDG1 (wwdg1_it) determines which processor will handle emergency tasks when needed. There are some characteristics of the watchdog that need to be noted when using the watchdog, which are generally the following two：

NOWAYOUT feature: When the user wants to start the watchdog without being stopped, i.e., any upper-level watchdog off action is not supported, the NOWAYOUT feature can be used at this time, e.g., CONFIG_WATCHDOG_NOWAYOUT=y After enabling this feature, the configuration needs to be supported in the specific wdt driver, call watchdog _set_nowayout() to set the use of WDOG_NO_WAY_OUT. The MYD-YT507H watchdog driver supports this feature.

The magic close feature: When the user wants to stop the watchdog, write the character "V" to the wdt node to stop the watchdog. magic close is not supported by the MYD-YT507H watchdog driver. This section will demonstrate the use of watchdog, simulate a kernel crash to test the watchdog system reset function, and provide a modified example to set the watchdog timeout.

1). User-space test watchdog

- **Simulate a kernel panic**

Simulate a kernel panic, test the watchdog reset function, default 32s reboot system:

```
console:/ # echo c > /proc/sysrq-trigger
[  167.195605]  sysrq: SysRq : Trigger a crash
[  167.200299] Intern al error: Accessing user space memory outside uaccess.
h routines: 96000045 [#1] PREEMPT SMP
[  167.211444] Modules linked in: gt9xxnew_ts  bcm_btlpm mali_kbase(O)
[  167.218422] CPU: 2 PID: 8836 Comm:  sh Tainted: G           O    4.9.17
0 #1
[  167.226146] H ardware name: myir-yt507h-full (DT)
[  167.231333] task: ffffffc03dabf000  task.stack: ffffffc03adac000
[  167.237997] PC is at sysrq_handle_crash+0x20/0x30
[  167.243287] LR is at sysrq_handle_crash+0xc/0x30
[ 167.248475] pc : [<fffff80084c6730>].  LR :  [<fffffff80084c671c>] pstate: 60
400145
[  167.272462] x25: 000000000000018a x24: 0000000000000011
[  167.278440] x23: 0000000000000000 x22: 0000000000000008
[  167.284416] x21: ffffff80090f80a0 x20: 0000000000000063
[ 9548.610640] pc : [<fffffff80084e7b00>] LR :  [<fffffff80084e7aec>] pstate: 6
0400145
```

2). The user program tests the watchdog

Run the watchdog app with a timeout of 4s and feed the dog every 1s interval:

```
console:/data # ./watchdog 4 1 0
Starting wdt_driver (timeout: 4, sleep: 1, test: ioctl)
Trying to set timeout value=4 seconds
The actual timeout was set to 4 seconds
Now reading back -- The timeout is 4 seconds
```

If the above 1s is changed to greater than 4s, the required 4s feeding time for the dog is exceeded, and the board will restart.

2.7. PMIC

This section demonstrates the Suspend feature of Linux power management, which sleeps the board and wakes up via external events. The Linux kernel generally provides three types of suspend: Freeze, Standby, and STR (Suspend to RAM), The following two commands both put the development board to sleep.

1). Check the modes currently supported by the board

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

2). The method written in user space

```
console:/ # echo "mem" > /sys/power/state
console:/ # echo "freee" > /sys/power/state
```

- **MEM sleeps**

After entering the sleep command, the development board sleeps, the debug serial port can no longer be entered, at this time, the system and device state are saved to memory (in self-refresh mode, its content has been preserved), and all devices enter a low-power mode.

```
console:/ # echo "mem" > /sys/power/state
[  154.771572] PM: suspend entry 2022-09-21 05:53:26.186580415 UTC
[  154.778394] PM: Syncing filesystems ... done.
[  154.792988] PM: Preparing system for sleep (mem)
[  154.977829] Freezing user space processes ... (elapsed 0.003 seconds) done.
[  154.989059] Freezing remaining freezable tasks ... (elapsed 0.002 seconds)
done.
[  155.000156] PM: Suspending system (mem)
[  155.013130]  sunxi-mmc sdc2: sdc set ios:clk 0Hz bm PP pm OFF vdd 0 wi
dth 1 timing LEGACY(SDR12) dt B
```

- **Freeze sleeps**

After entering the sleep command, the development board sleeps, the debug serial port can no longer be entered, the user space is frozen, all I/O devices enter a low-power state, and the processor enters an idle state.

```
console:/ #  echo "freeze" > /sys/power/state
[   79.371834] PM: suspend entry 2022-09-21 05:52:03.535757563 UTC
[   79.378615] PM: Syncing filesystems ... done.
[   79.389464] PM: Preparing system for sleep (freeze)
[   79.591260] Freezing user space processes ... (elapsed 0.002 seconds) done.
[   79.602353] Freezing remaining freezable tasks ... (elapsed 0.002 seconds) done.
[   79.613384] PM: Suspending system (freeze)
[   79.626683] sunxi-mmc sdc2: sdc set ios:clk 0Hz bm PP pm OFF vdd 0 width 1 timing LEGACY(SDR12) dt B
[   79.629231] G2D: g2d_suspend.
[   79.629236] G2D: g2d_suspend succesfully.
[   79.629306] disp_suspend
[   79.651286] [ehci3-controller]: sunxi_ehci_hcd_suspend
```

At this time, press the user button and press S2 for 2~3 seconds to wake up the system:

```
[  166.807928]  sunxi-gmac gmac0 eth0: Link is Up - 1Gbps/Full - flow control rx/tx
[  172.171369]  sunxi-mmc sdc2: sdc set ios:clk 0Hz bm PP pm UP vdd 22 width 1 timing LEGACY(SDR12) dt B
[  172.197620]  sunxi-mmc sdc2: sdc set ios:clk 400000Hz bm PP pm ON vdd 22 width 1 timing LEGACY(SDR12) dt B
[  172.224272]  sunxi-mmc sdc2: sdc set ios:clk 400000Hz bm OD pm ON vdd 22 width 1 timing LEGACY(SDR12) dt B

[  172.235258]  sunxi-mmc sdc2: sdc set ios:clk 400000Hz bm OD pm ON vdd 22 width 1 timing LEGACY(SDR12) dt B
```

At this time, the debug serial port can be re-entered.

2.8. GPIO

GPIO testing is achieved through the file system sysfs interface, the following content uses PD21, PD26 as an example to illustrate the use of GPIO.

Calculate the value of the corresponding pin of GPIO = (n-1)*32+x

(Let A be 1, B be 2, and so on, H corresponds to 8).

(2 in PH2). 。

Such as:

PH2 corresponds to a value of :(8-1)x32+2=226 PH3 corresponds to a value of:

(8-1)x 32+3=227

Design the backplane's expansion interfaces PD21 and PD26

PD21=(4-1*32+21=117

PD26=(4-1)*32+26=122

```
console:/ # echo 117>/sys/class/gpio/export
console:/ # echo out>/sys/class/gpio117/direction
console:/# echo 1>xxx/value
console:/# echo 0>xxx/value
```

3. Basic peripheral interface

3.1. LED lights

The Linux system provides a separate subsystem to facilitate the operation of LED devices from user space, which provides an operating interface for LED devices in the form of a file. These interfaces are located in the /sys/class/leds directory. In the Hardware Resources list, we have listed all the LEDs on the board. The following tests the LED by reading and writing sysfs with the command. The following commands are generic and a common way to manipulate LEDs.

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

```
console:/ # cd /sys/class/leds/
console:/sys/class/leds # ls
blue
```

The duty cycle of the heartbeat light can be changed by the following command.

2). Take heartbeat light 1 as an example to test the LED

- **Turn off the heartbeat light**

```
console:/sys/class/leds # echo none > /sys/class/leds/blue/trigger
```

Turn off the heartbeat light, and then the LED can be turned off and on separately.

- **Turn off the LED**

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

- **Light up the LED**

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

- **Turn on LED trigger mode**

When the "heartbeat" mode is turned on, the LED blinks at 1Hz by default with a duty cycle of 50%:

```
console:/sys/class/leds # echo heartbeat > /sys/class/leds/blue/trigger
```

3.2. Key

The Linux /dev/input_eventx device can be used to easily debug input devices such as mouse, keyboard, and trackpad.This section focuses on testing keys.Use the hexdump command and the dmesg command to see if there is pushback.MYD-YT507H has four keys, S1 is the system reset button;S2 is the Poweroff&on button, which has been configured in the device tree.S3 is FEL burn key.

- S1 is the reset key
- S2 is the poweroff&on key (no function).
- S3 is the wakeup key
- S4 is the FEL burn button

1). Device tree configuration information

Open the companion device tree file myir-yt507.dtsi and see the node of the key S3 wakeup button:

```
powerkey0: powerkey@0{
                        status = "okay";
                        compatible = "x-powers,axp2101-pek";
                        pmu_powkey_off_time = <6000>;
                        pmu_powkey_off_func = <0>;
                        pmu_powkey_off_en = <1>;
                        pmu_powkey_long_time = <1500>;
                        pmu_powkey_on_time = <1000>;
                        wakeup_rising;
                        wakeup_falling;
  };
```

2). Key test

- **Review the corresponding input getevent information**

```
console:/sys/class/leds # getevent
add device 1: /dev/input/event0
  name:     "sunxi-keyboard"
add device 2: /dev/input/event4
```

```
  name:     "sunxi-gpadc2"
add device 3: /dev/input/event5
  name:     "sunxi-gpadc3"
add device 4: /dev/input/event2
  name:     "sunxi-gpadc0"
add device 5: /dev/input/event3
  name:     "sunxi-gpadc1"
add device 6: /dev/input/event1
  name:     "axp2101-pek"
```

From the above, we can see that the corresponding device event of gpio-keys is event1.

- **getevent tests the key information**

Execute the following command, operate the button S3, and the serial terminal will print the following information:

```
console:/sys/class/leds # getevent
add device 1: /dev/input/event0
  name:     "sunxi-keyboard"
add device 2: /dev/input/event4
  name:     "sunxi-gpadc2"
add device 3: /dev/input/event5
  name:     "sunxi-gpadc3"
add device 4: /dev/input/event2
  name:     "sunxi-gpadc0"
add device 5: /dev/input/event3
  name:     "sunxi-gpadc1"
add device 6: /dev/input/event1
  name:     "axp2101-pek"
6
/dev/input/event1: 0001 0074 00000001
/dev/input/event1: 0000 0000 00000000
/dev/input/event1: 0001 0074 00000000
/dev/input/event1: 0000 0000 00000000
```

Each time S3 is pressed, the current terminal will print out the current event code value, that is, the key is normal.

3.3. USB

This section verifies the feasibility of the USB Host driver through relevant commands or hot plugging, USB HUB, and implements the function of reading and writing USB flash drives and USB enumeration functions.

1). View the Insert USB information

- **View USB device information**

After plugging in the USB device, slide up from the bottom of the screen to the application interface, and then open the ES File Explorer, you can see the name and usage of the plugged USB device

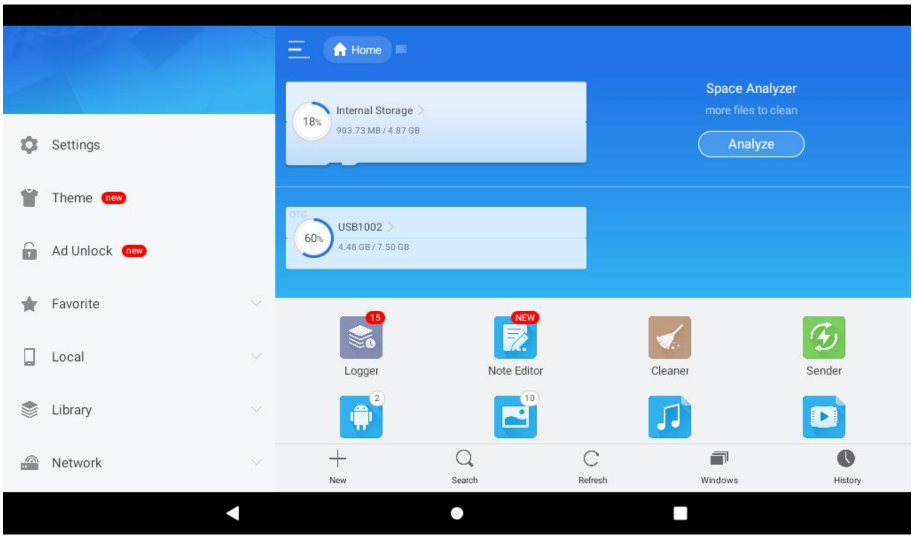


Figure 3-1 ES File Explorer home page

2). U disk read and write test

- **Go to the mount directory**

```
console:/ # cd /mnt/media_rw/B4FE-5315
console:/mnt/media_rw/B4FE-5315 #
```

- **Read the file**

You need to create a test .txt file on the USB stick in advance.

```
console:/mnt/media_rw/B4FE-5315 # ls
test.txt
console:/mnt/media_rw/B4FE-5315 # cat /mnt/test.txt
```

```
helloworld!
```

- **Write the file**

```
console:/mnt/media_rw/B4FE-5315 # touch test.txt
console:/mnt/media_rw/B4FE-5315 # echo hello> 1
console:/mnt/media_rw/B4FE-5315 # cat test.txt
hello
console:/mnt/media_rw/B4FE-5315 # sync
```

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

3). Unmount the USB flash drive

- **Uninstall operation**

```
root@myir:~#umount /mnt
```

3.4. Micro SD card

Micro SD Card, formerly known as Trans-flash Card (TF card), Micro SD card is a very small flash memory card. Micro SD cards are smaller than standard SD cards and are the smallest SD card in SD card types. Although the Micro SD card has a different form factor and interface shape than the original SD card, the interface specification remains the same to ensure compatibility. If Micro SD is inserted into a specific riser card, it can be used as a standard SD card, SD card has become the most widely used memory card in consumer digital devices, with large capacity, high performance, security and other characteristics of multi-function memory card. Micro SD cards generally have 9 pins on the back, including 4 data cables, and support two data transmission widths: 1bit/4bit. MYD-YT507H supports 3 8-bit SDMMC interfaces, and uses SDMMC1 on the development board to connect to Micro SD. The hardware specifications related to this interface are as follows:

Support 1bit/4bit bit SDMMC interface, fully compliant with SD card v3.01 interface specification support SDHC Class 10 MicroSD card support the first generation UHS bus interface (UHS-1 speed class U3), UHS-II is not supported. The UHS-I maximum transfer speed (theoretical) is 104MB/s. The letter I indicates that the device (SD card or card reader) supports the UHS-I interface. The English letter U, including the number word 3, indicates that the device has a read and write speed level of U3. Supports SDHC cards (>2GB to 32GB), SDXC cards (>32GB to 2TB). This section describes the steps and methods to view and operate TF cards under Linux.

1). Check the TF card capacity

The steps are the same as viewing USB devices, open the ES File Explorer to see the capacity and usage of the TF card.

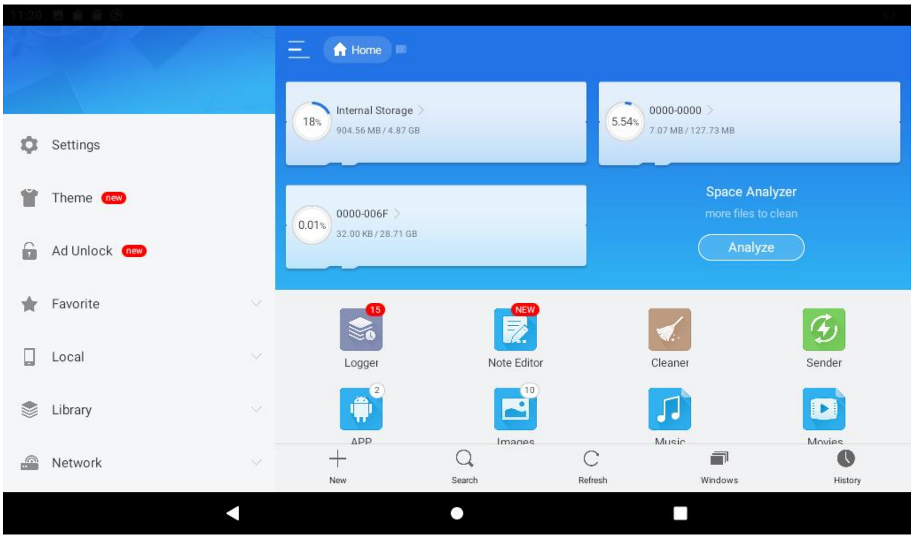


Figure 3-2 ES File Explorer home page

2). Performance test of TF card

- **Write file test**

```
console:/mnt/media_rw/CA05-7A8E # 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, 33.090494 s, 15 M/s
    0m33.14s real     0m00.04s user     0m06.30s system
```

Here the write disk speed is tested at 15M/s.

- **Read file test**

```
console:/mnt/media_rw/CA05-7A8E # time dd if=test_file_w of=test_file_r bs=1M count=500
500+0 records in
```

```
500+0 records out
524288000 bytes (500 M) copied, 16.106955 s, 31 M/s
0m16.15s real       0m00.04s user       0m06.61s system
```

It can be seen that the speed of reading directly from the SD card is 31m/s.

3.5. ADC

GPADC is a 12-bit sampling accuracy analog-to-digital conversion module, supporting four channels, and the analog input range is determined by the platform (1.8V for the T507 platform). ADC testing is achieved through the file system sysfs interface, which is illustrated below using ADC1 channel 0 as an example. GPADC0 pin of J28 interface.

1). Enable the ADC

```
console:/ # echo gpadc0,1 > /sys/class/gpadc/status
```

2). Read the sysfs interface of the ADC

Use the command to view the ADC read interface:

```
console:/ # getevent
add device 1: /dev/input/event5
  name:      "sunxi-gpadc3"
add device 2: /dev/input/event2
  name:      "sunxi-gpadc0"
add device 3: /dev/input/event3
  name:      "sunxi-gpadc1"
add device 4: /dev/input/event0
  name:      "sunxi-keyboard"
add device 5: /dev/input/event4
  name:      "sunxi-gpadc2"
add device 6: /dev/input/event1
  name:      "axp2101-pek"
```

The event2-5 interface can correspond to test ADC 0-3.

3). Read the ADC test values

```
console:/ # getevent
```

```
add device 1: /dev/input/event5
  name:       "sunxi-gpadc3"
add device 2: /dev/input/event2
  name:       "sunxi-gpadc0"
add device 3: /dev/input/event3
  name:       "sunxi-gpadc1"
add device 4: /dev/input/event0
  name:       "sunxi-keyboard"
add device 5: /dev/input/event4
  name:       "sunxi-gpadc2"
add device 6: /dev/input/event1
  name:       "axp2101-pek"
/dev/input/event2: 0004 0004 00000ab1
/dev/input/event2: 0000 0000 00000000
/dev/input/event2: 0004 0004 00000ab1
/dev/input/event2: 0000 0000 00000000
```

3.6. Display

This module consists of a display engine (DE) and various types of controllers (tcon). After the input layers are displayed in DE, they are output to the display device through one or more interfaces to achieve the effect of composing layers rendered by many applications and presenting them to the user on the display. DE has 2 independent units (can be abbreviated as de0, de1), which can accept layers of user input Compositing, output to different displays for dual display. Each independent unit of DE has 1-4 channels (typically, de0 has 4 on and de1 has 2), each channel can simultaneously process and accept 4 layers of the same format. The sunxi platform has a video channel and a UI channel. Video channels are powerful and can support both YUV formats and RGB layers. UI channels only support RGB layers. In simple terms, the main functions of the display module are as follows:

- Supports LCD(HV/LVDS/CPU/DSI) output
- Supports dual display output
- Support multi-image overlay blending processing

- Support a variety of display effect processing (alpha, colorkey, image enhancement, brightness/contrast/saturation/chroma adjustment).
- Support intelligent backlight adjustment
- Supports multiple image data format inputs (argb, yuv).
- Support image scaling processing • Support screenshot • Support image conversion

It can support HDMI display, 7 and 10.1-inch LVDS display, 21-inch dual-channel LVDS display, etc CVBS display, the image is displayed as HDMI by default, if you need to change other displays, you need to modify the device.

Modify the path of the display scheme device tree, the following path is for reference only, please modify according to your actual path.

```
dth@TH:/t507h-android-0829/longan/device/config/chips/t507/configs/evb
```

Open the board.dts file, if you need to display anything, uncomment it, comment out the previous display, and avoid conflicts.

```
/*
 * myir-yt507 support.
 */
/dts-v1/;
#include "myir-yt507.dtsi"
#include "display/myir-hdmi-1920x1080-1lvds-7-1024x600.dtsi"
#include "display/myir-lcd-1lvds-7-1024-600.dtsi"
#include "display/myir-lcd-lvds-10.1-1280-800.dtsi"
#include "display/myir-lcd-2lvds-7-1024-600.dtsi"
#include "display/myir-lcd-2lvds-21-1920-1080.dtsi"
#include "display/myir-lcd-lvds1-7-1024-600.dtsi"
#include "display/myir-hdmi.dtsi"
#include "display/myir-lvds.dtsi"
#include "display/myir-tv.dtsi"
```

1). HDMI display

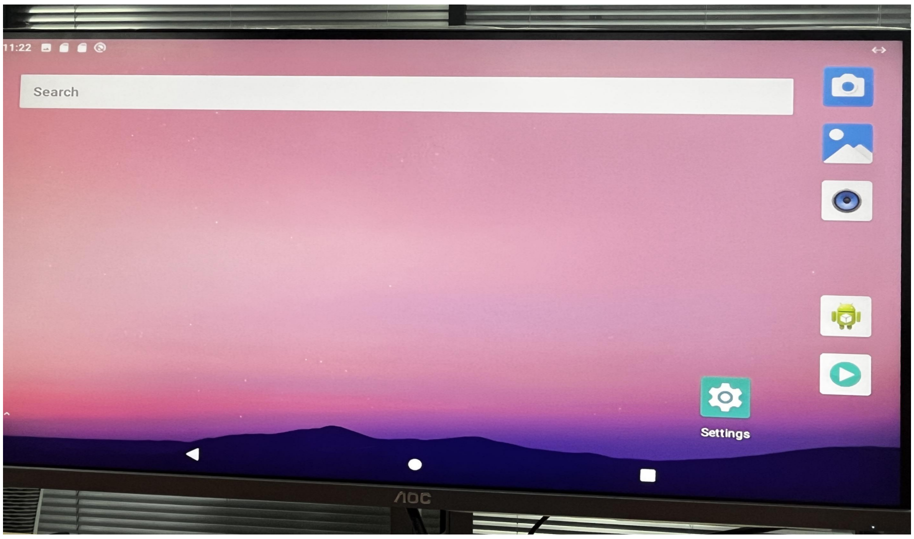


Figure 3-3 HDMI display

2). LVDS display

- **7 LVDS display**

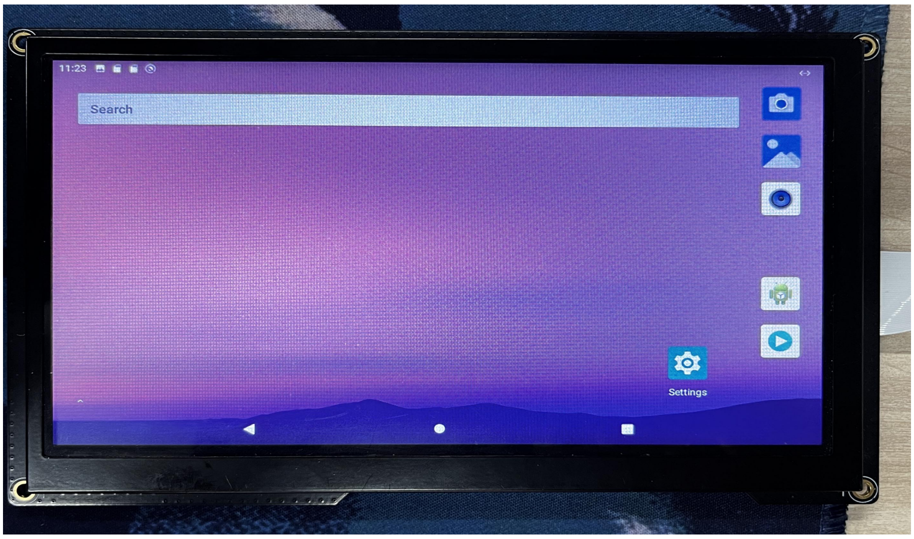


Figure 3-4 7 LVDS display

- **7 LVDS display connection method**

Find J12 at the bottom of the development board, there is a white inverted triangle on it, and also find the white inverted triangle at the bottom of the LVDS screen J1, and the two triangles should be accessed on the same side. (Use a heterogeneous cable, do not connect in the wrong direction).

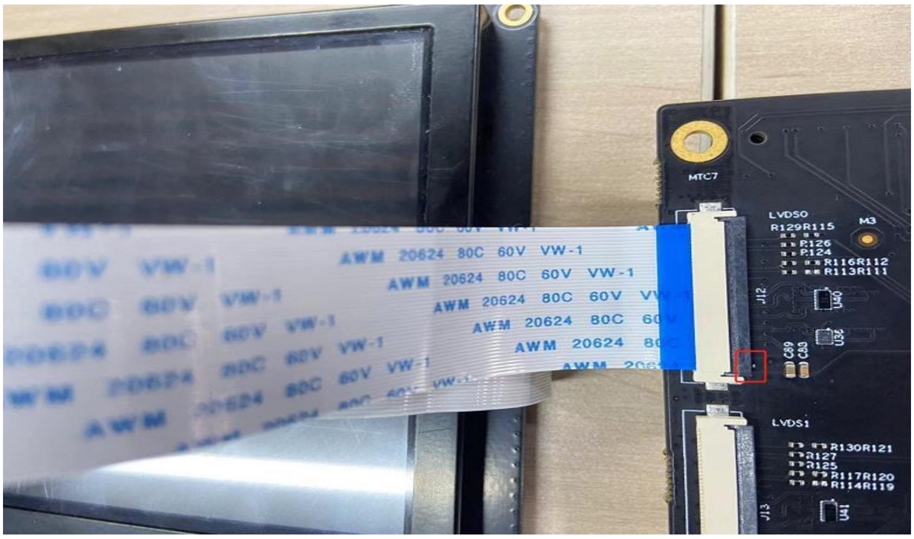


Figure 3-5 Schematic diagram of the bottom of the development board

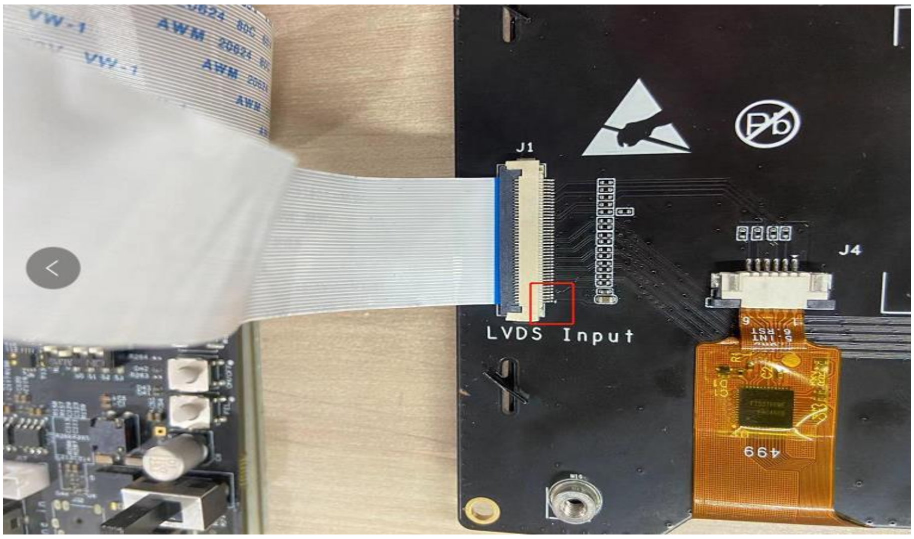


Figure 3-6 Schematic diagram of the bottom of LVDS

7 inch LVDS screen: http://www.myir-tech.com/product/my-lvds070c.htm

- **21 dual-channel LVDS display**

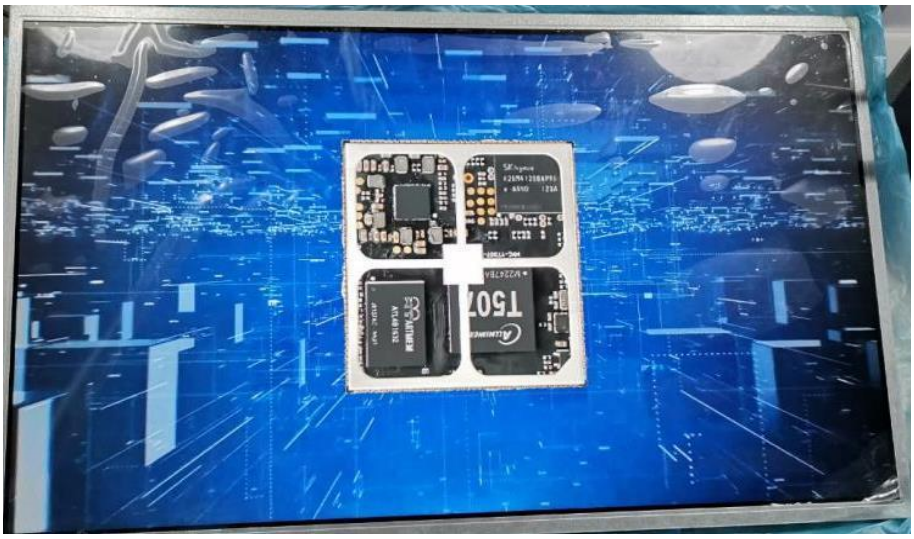


Figure 3-7 dual-channel LVDS display

3). CVBS display

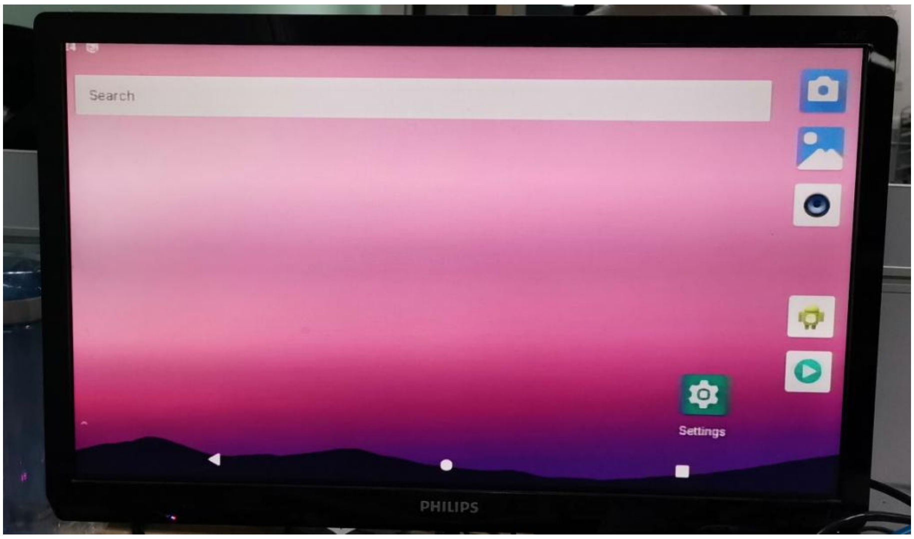


Figure 3-8 CVBS displays

4). Multi-screen display

Image the default HDMI + 7 LVDS screen dual display, one main screen and one secondary screen, in the terminal can query the information of the main screen and the secondary screen through commands. Among them, screen0 is the main screen and screen1 is the secondary screen.

```
console:/ # cat /sys/class/disp/disp/attr/sys
screen 0:
de_rate 696000000 hz, ref_fps:60
```

```
mgr0: 1280x720 fmt[yuv444] cs[0x101] range[limit] eotf[0x4] bits[8bits] err[0] f
orce_sync[0]  unblank direct_show[false]
dmabuf: cache[7] cache max[36] umap skip[1] overflow[5]
        hdmi output mode(5)     fps:60.6        1280x 720
        err:1    skip:162        irq:23012        vsync:23012      vsync_skip:0
   BUF     enable ch[1] lyr[0] z[0] prem[Y] a[globl 255] fmt [   1] fb[1024, 600;
 512, 300; 512, 300] crop[    0,    0,1024, 600] frame[    0,    0,1280, 720] addr
[fb800000,fb896000,fb8bb800] flags[0x         0] trd[0,0]
depth[ 0] transf[0]
screen 1:
de_rate 696000000 hz, ref_fps:60
mgr1: 1024x600 fmt[rgb] cs[0x204] range[full] eotf[0x4] bits[8bits] err[0] force_
sync[0] unblank direct_show[false]
dmabuf: cache[7] cache max[9] umap skip[1] overflow[1]
        lcd output       backlight(255)  fps:60.2        1024x 600
        err:0    skip:151        irq:22552        vsync:22552      vsync_skip:0
   BUF     enable ch[1] lyr[0] z[0] prem[Y] a[globl 255] fmt[   1] fb[1024, 600;
512, 300; 512, 300] crop[    0,    0,1024, 600] frame[    0,    0,1024, 600] addr[f
b400000,fb496000,fb4bb800] flags[0x         0] trd[0,0]
depth[ 0] transf[0]
disp[0]all:337, sub:337, cur:337, free:333, skip:0
disp[1]all:337, sub:337, cur:337, free:334, skip:0
```

- **Multi-screen simultaneous display**

The Android desktop system performs simultaneous display operations by default.

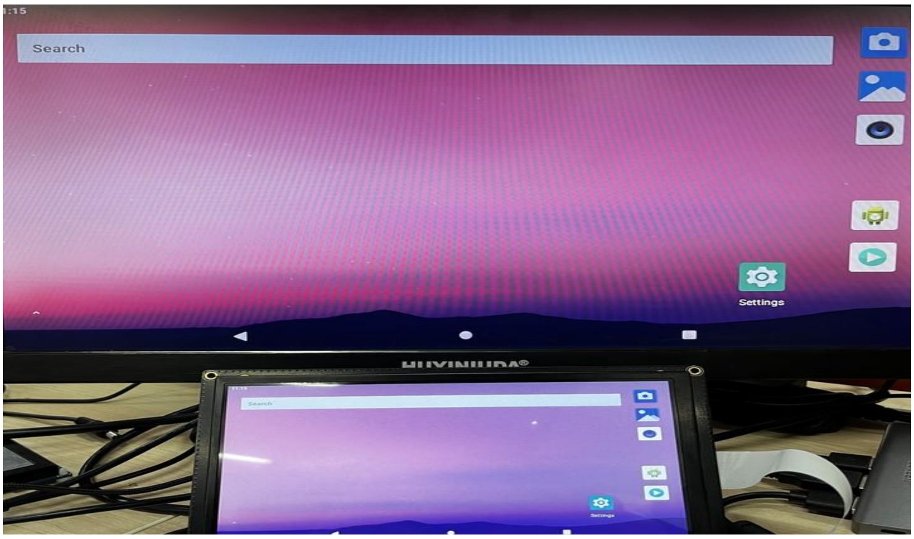


Figure 3-9 HDMI+LVDS simultaneous display

- **Multi-screen display**

The display function needs to open the video player and select the video to be played to achieve the display, under this application, the display function is turned on by default without any settings. Therefore, the abnormal display function needs to be implemented at the application layer, and other operations under Android are the same display.

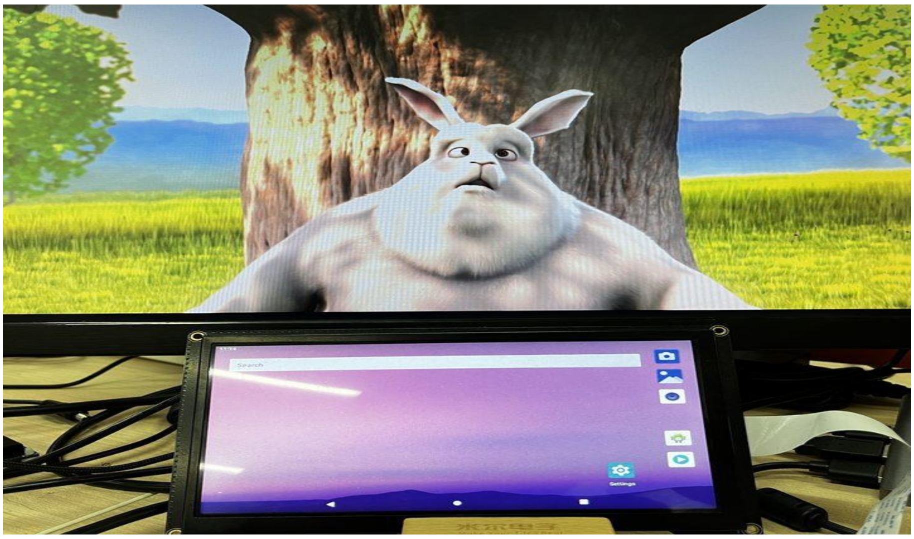


Figure 3-10 HDMI + LVDS abnormal display

3.7. Touch Panel

Touch has capacitive touch and resistive touch, MYD-YT507H development board hardware currently does not support resistive touch, but supports capacitive touch, Mir Technology provides LVDS display screen accessories, see Table 1-1, the display supports multi-touch 。 Accessories can be purchased according to actual needs. The capacitive screen is more sensitive in use and rarely has problems. In addition, the capacitive screen does not need to be accurate. Because according to the principle of capacitive screen, capacitive screen can accurately identify the position of finger contact with the screen in use, and has high sensitivity. If we click on the phenomenon that the software is not selected in use, there is generally only one situation: there is a problem with the screen. The following is a simple test to test the capacitive screen touch function with the evtest command

1). Touch screen connection

Follow Section 3.6 to connect the MY-LVDS 070C LVDS screen to the development board.

2). Multi-touch test

Use the multi-touch test function of AnTuTu software, which supports up to five points.

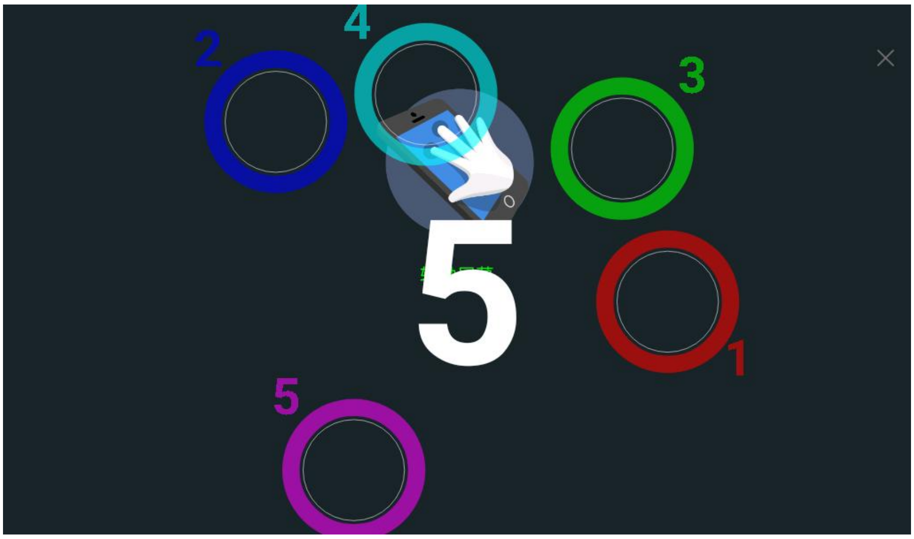


Figure 3-11 Multi-touch test

3). evtest command test

The terminal executes "getevent" to enter the testing interface. Select Test Outside to set to touch screen, here the default is input interrupt 0, test interface select '0' Press Enter to start the test:

```
console:/ # getevent
```

```
add device 1: /dev/input/event6
  name:     "WaveShare WS170120"
add device 2: /dev/input/event5
  name:     "sunxi-gpadc3"
add device 3: /dev/input/event2
  name:     "sunxi-gpadc0"
add device 4: /dev/input/event3
  name:     "sunxi-gpadc1"
add device 5: /dev/input/event0
  name:     "sunxi-keyboard"
add device 6: /dev/input/event4
  name:     "sunxi-gpadc2"
add device 7: /dev/input/event1
  name:     "axp2101-pek"
6
/dev/input/event6: 0003 0039 00000000
/dev/input/event6: 0003 0035 000000bf
/dev/input/event6: 0003 0036 000000f8
```

3.8. Ethernet

MYD-YT507H has two network ports, eth0 is the gigabit network port, eth1 is the 100 Gigabit network port. Android system currently does not support dual Ethernet work, only support eth0 gigabit network port Internet access, here is an example of eth0 gigabit network port to introduce the commonly used Ethernet configuration methods.

1). Configure the Ethernet IP address

- **Use ifconfig in the net-tools toolkit to manually configure the network**

First, use the ifconfig command to view the network device information as follows:

```
console:/ # ifconfig
eth0      Link encap:Ethernet  HWaddr 36:c9:e3:f1:b8:05  Driver sunxi-gmac
inet addr:192.168. 40.207  Bcast:192.168.40.255  Mask:255.255.255.0
          inet6 addr: fe80::d48d:4a46:6279:3735/64 Scope: Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:380 errors:0 dropped:1 overruns:0 frame:0
          TX packets:98 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:76451 TX bytes:13120
          Interrupt:65

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:1
          RX bytes:0 TX bytes:0

eth1       Link encap:Ethernet  HWaddr 9a:20:bc:c3:58:fa  Driver sunxi-gmac
          UP BROADCAST MULTICAST MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 TX bytes:598
          Interrupt:66
```

The following describes how to manually configure the IP address 192.168.1.100 for eth0 with the following command:

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

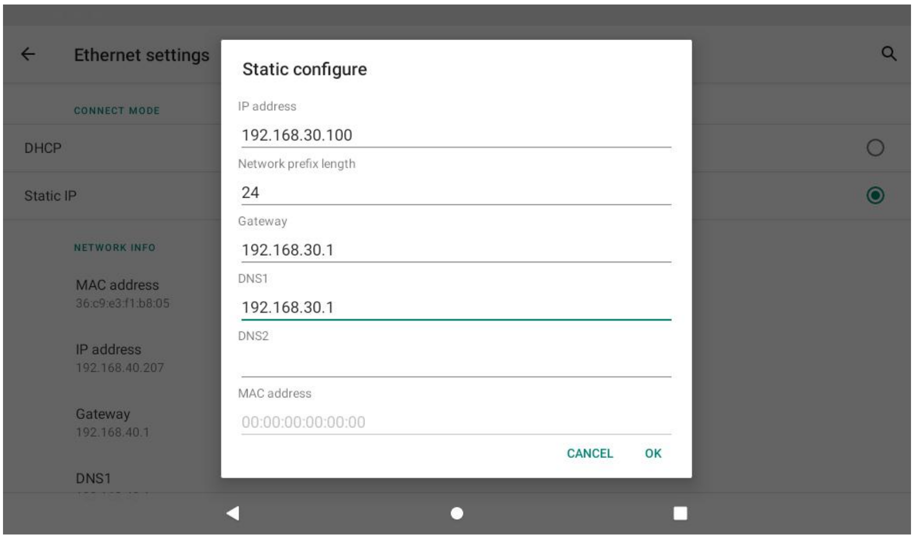
The above command manually configures eth0 with an IP address of 192.168.1.100, a subnet mask of 255.255.255.0, and a broadcast address of 192.168.1.255 configured by default and passes ,The up parameter is activated as follows:

```
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
```

- **The Ethernet IP interface is configured**

Find the setting function to enter the setting interface, enter the network and Internet, enter the Ethernet, select the static IP mode, and start configuring your own parameters, each parameter can refer to DHCP mode.



Static configure

IP address: 192.168.30.100

Network prefix length: 24

Gateway: 192.168.30.1

DNS1: 192.168.30.1

DNS2:

MAC address: 00:00:00:00:00:00

MAC address: 36:c9:e3:f1:b8:05

IP address: 192.168.40.207

Gateway: 192.168.40.1

DNS1

Figure 3-12 Modifying an IP address

After the configuration is completed, check whether the IP address has changed in the terminal to ensure that the modification is successful, and check the IP address of eth0 below, which is indeed consistent with the modification.

```
console:/ # ifconfig
eth0          Link encap:Ethernet  HWaddr 36:c9:e3:f1:b8:05  Driver sunxi-gmac
inet addr:192.168.30. 100 Bcast:192.168.30.255 Mask:255.255.255.0
          inet6 addr: fe80::4c20:8b29:44a3:161b/64 Scope: Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:1892 errors:0 dropped:1 overuns:0 frame:0
          TX packets:314 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:448293 TX bytes:26328
          Interrupt:65
```

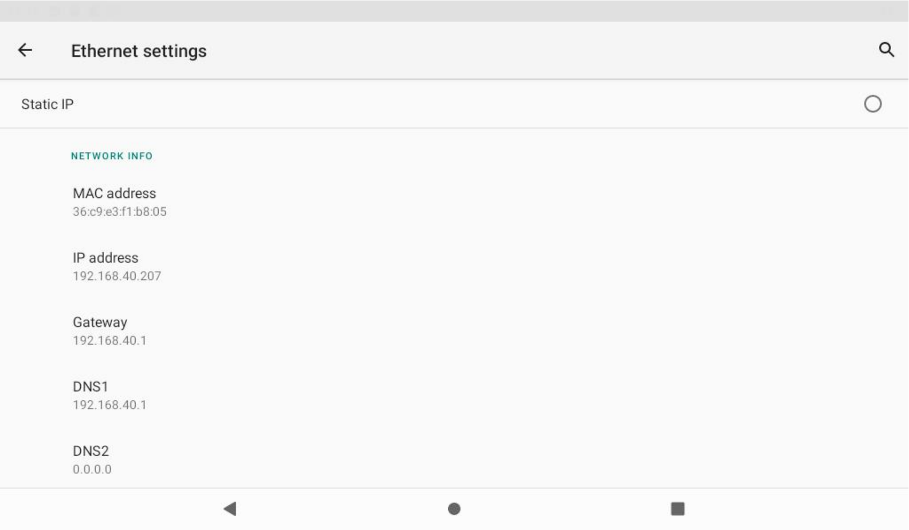
2). Modify the MAC address

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

```
console:/ # ifconfig eth0 hw ether 00:0C:29:36:97:20
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:dd:12:18:15:46:21: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
```

3). Check the IP address

Find the settings on the desktop, enter the setting interface, enter the network and the Internet, and finally enter the Ethernet to see your IP related information.



MAC address
36:c9:e3:f1:b8:05

IP address
192.168.40.207

Gateway
192.168.40.1

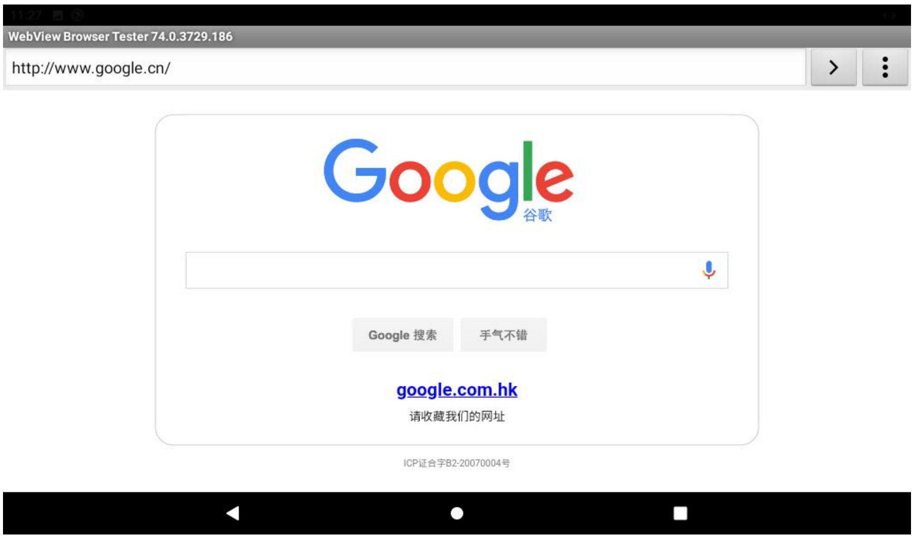
DNS1
192.168.40.1

DNS2
0.0.0.0

Figure 3-1 3 Ethernet IP address

4). Ethernet access to the Internet

Find a browser on the desktop and open the input www.baidu.com to visit the Baidu webpage.

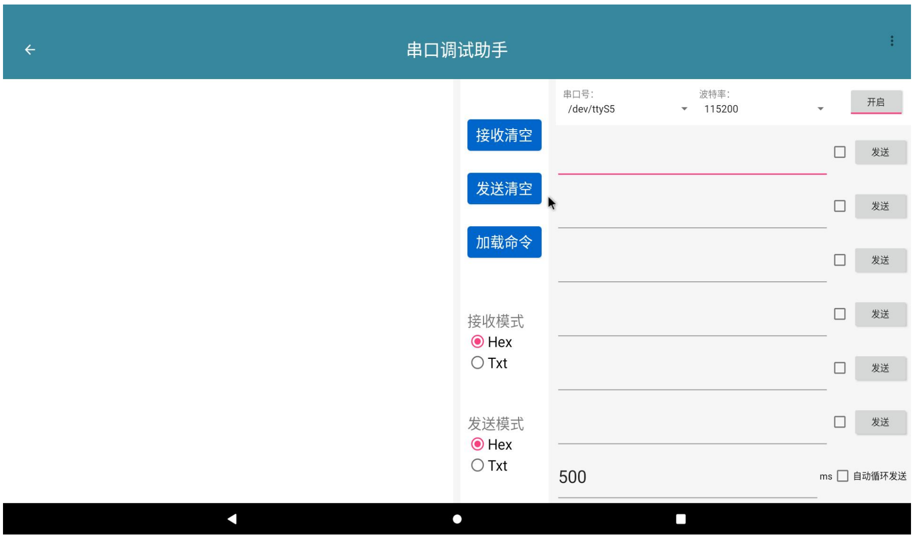


http://www.google.cn/

Figure 3-14 Ethernet access to Baidu

3.9. RS232 and RS485 test

Open the serial port debugging assistant software in the application interface to select the serial port number, baud rate, and then click to open, click the send bar, edit the sent data, and then click send, note that the serial port number corresponding to RS232 is ttyS5, and the serial port number corresponding to RS485 is ttyS2.



串口号: /dev/ttyS5

波特率: 115200

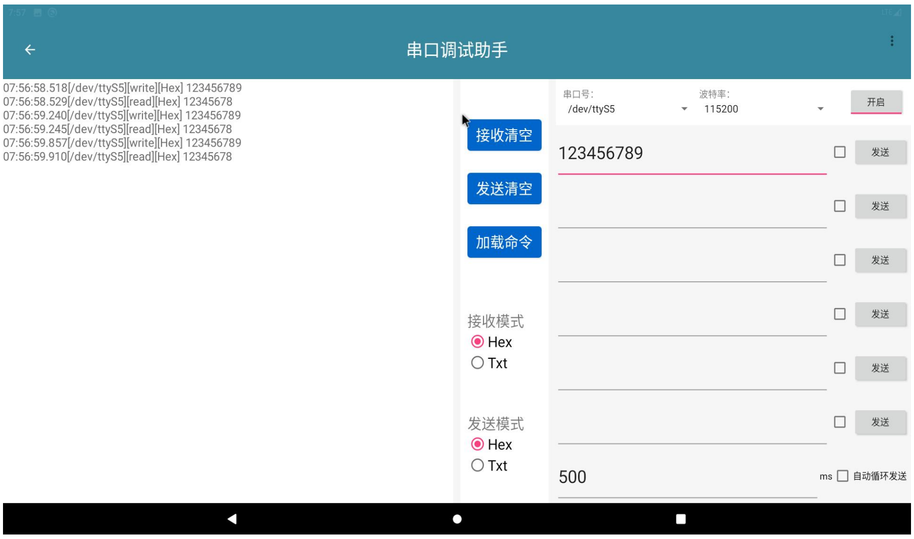
接收模式
● Hex
○ Txt

发送模式
● Hex
○ Txt

500 ms ☐ 自动循环发送

Figure 3-15 Serial port debugging assistant boundary

1). RS232 test

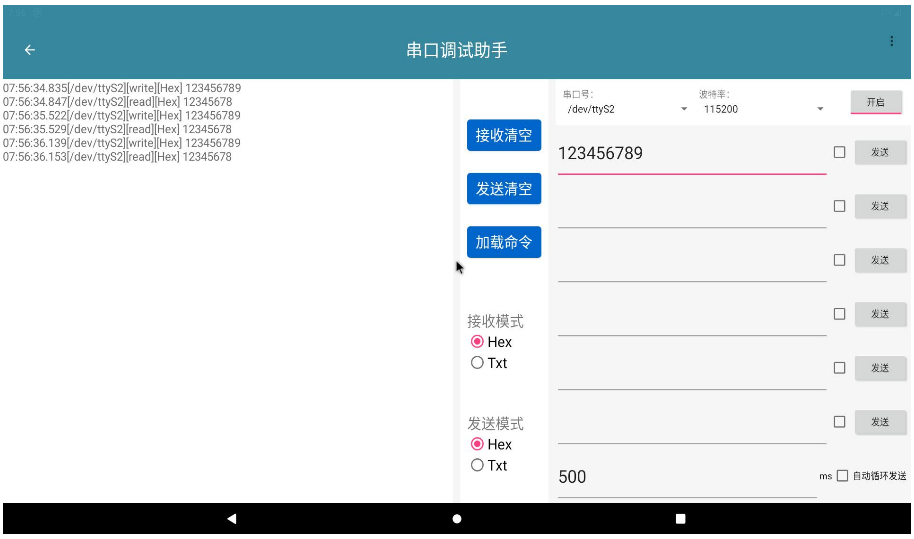


```
07:56:58.518[/dev/ttyS5][write][Hex] 123456789
07:56:58.529[/dev/ttyS5][read][Hex] 12345678
07:56:59.240[/dev/ttyS5][write][Hex] 123456789
07:56:59.245[/dev/ttyS5][read][Hex] 12345678
07:56:59.857[/dev/ttyS5][write][Hex] 123456789
07:56:59.910[/dev/ttyS5][read][Hex] 12345678
```

串口号: /dev/ttyS5
波特率: 115200
123456789
500

Figure 3-16 RS232 test

2). RS485 test



```
07:56:34.835[/dev/ttyS2][write][Hex] 123456789
07:56:34.847[/dev/ttyS2][read][Hex] 12345678
07:56:35.522[/dev/ttyS2][write][Hex] 123456789
07:56:35.529[/dev/ttyS2][read][Hex] 12345678
07:56:36.139[/dev/ttyS2][write][Hex] 123456789
07:56:36.153[/dev/ttyS2][read][Hex] 12345678
```

串口号: /dev/ttyS2
波特率: 115200
123456789
500

Figure 3-17 RS485 test

4. Expand peripheral interfaces

The MYD-YT507H development board provides a wealth of peripheral interfaces, in addition to the basic peripheral interface, it can also be connected to various expansion modules. Make the user's development more flexible and convenient. The following describes the test steps of several optional modules introduced by Mir. Users purchase by themselves according to their needs. For details of optional modules, please refer to Table 1-2 for the list of optional modules.

4.1. MY-WF005S module

WI-FI/BT module MY-WF005S is a Wi-Fi and Bluetooth two-in-one module launched by Mill, the chip program is AP6212, which needs to be purchased by users according to their needs. For module details, please refer to Table 1-1, Optional Modules. Before testing, the user needs to connect the module to the J26 interface of the development board.

1). Wi-Fi test

This section mainly introduces the configuration and use of Wi-Fi under Linux, usually Wi-Fi modules can support two working modes, namely STA mode and AP mode, and some devices also support STA Works simultaneously with AP mode. STA mode allows devices to connect to external Wi-Fi hotspots, and AP mode turns devices into Wi-Fi hotspots for other devices to connect.

MYD-YT507H can be connected to MYIR's AP6212 Wi-Fi and Bluetooth 2-in-1 modules, and currently does not support STA and AP working at the same time The corresponding driver of the AP6212 Wi-Fi module is:

```
console:/ # lsmod
Module                  Size  Used by     Tainted: G
bcmdhd                 1269760  0
mali_kbase             520192  3
```

During driver loading, the Wi-Fi firmware located in /vendor/modules will be loaded into the module. After the Wi-Fi module driver is loaded successfully, the network node wlan0 of the Wi-Fi device is generated, as shown below:

```
console:/ # ifconfig wlan0
wlan0     Link encap:Ethernet  HWaddr 5A:38:06:A6:8C:60
inet addr:192.168.30. 1 Bcast:192.168.30.255 Mask:255.255.255.0
          inet6 addr: fe80::7573:713b:e50d:48e7/64 Scope:Link
          UP BROADCAST RUNNING NOARP MULTICAST MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:61 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
RX bytes:0 (0.0 B) TX bytes:18920 (18.4 KiB)
```

2). STA mode connects to WI-FI hotspots

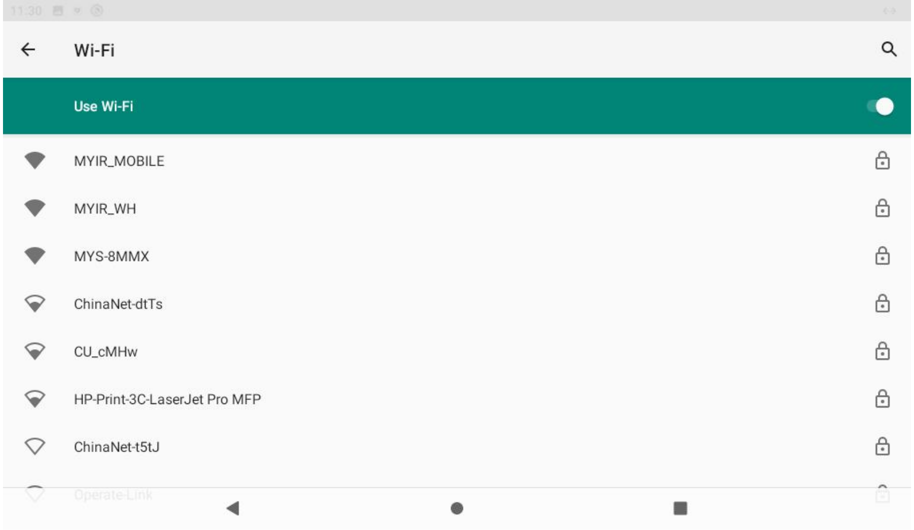
Let's try manually connecting to a nearby Wi-Fi hotspot, which is a Wi-Fi hotspot using WPA2 densification.

Ensure that the wlan0 network device is active.

```
console:/ # ifconfig wlan0 up
```

- **Scan nearby WI-FI hotspots**

You can directly view the list of WI-FI hotspots through the display interface, and enter Settings/Network and Internet/WLAN

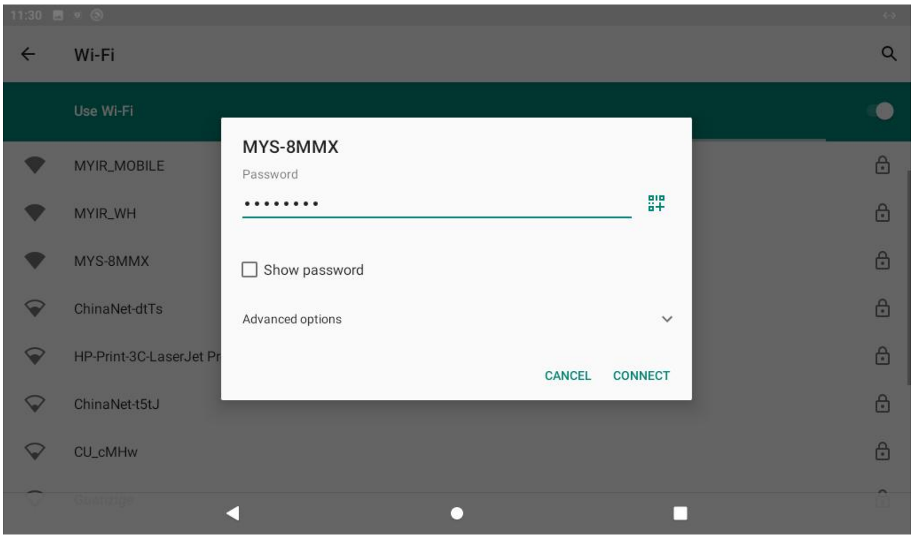


MYIR_MOBILE
MYIR_WH
MYS-8MMX
ChinaNet-dtTs
CU_cMHw
HP-Print-3C-LaserJet Pro MFP
ChinaNet-t5tJ
Operate-Link

Figure 4-1 View the list of WI-FI hotspots

- **Connect to WI-FI**

On the WLAN interface, select the WI-FI to be connected and enter the password to connect.



MYS-8MMX

Password

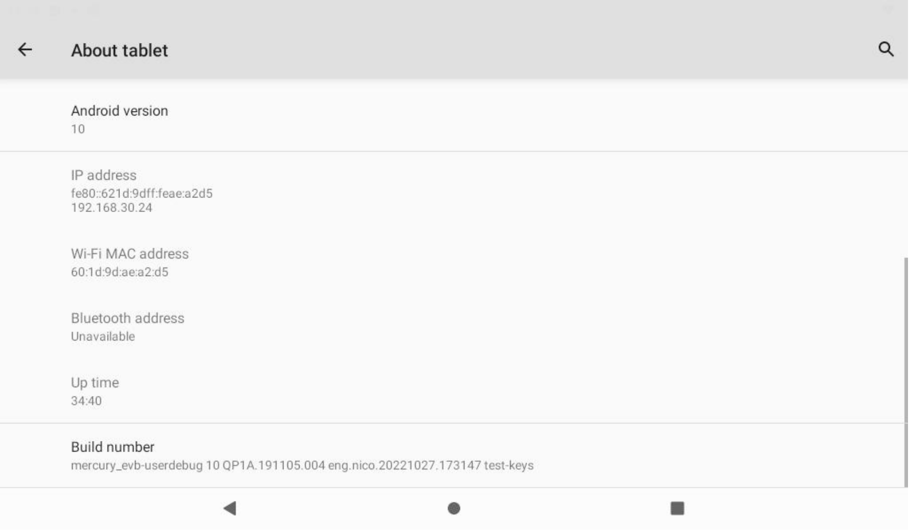
••••••••

☐ Show password

Figure 4-2 WI-FI connection

- **Get the IP address**

After connecting to WI-FI, the IP-P address is automatically obtained



Android version
10

IP address
fe80::621d:9dff:feae:a2d5
192.168.30.24

Wi-Fi MAC address
60:1d:9d:ae:a2:d5

Bluetooth address
Unavailable

Up time
34:40

Build number
mercury_evb-userdebug 10 QP1A.191105.004 eng.nico.20221027.173147 test-keys

Figure 4-3 WI-FI IP address

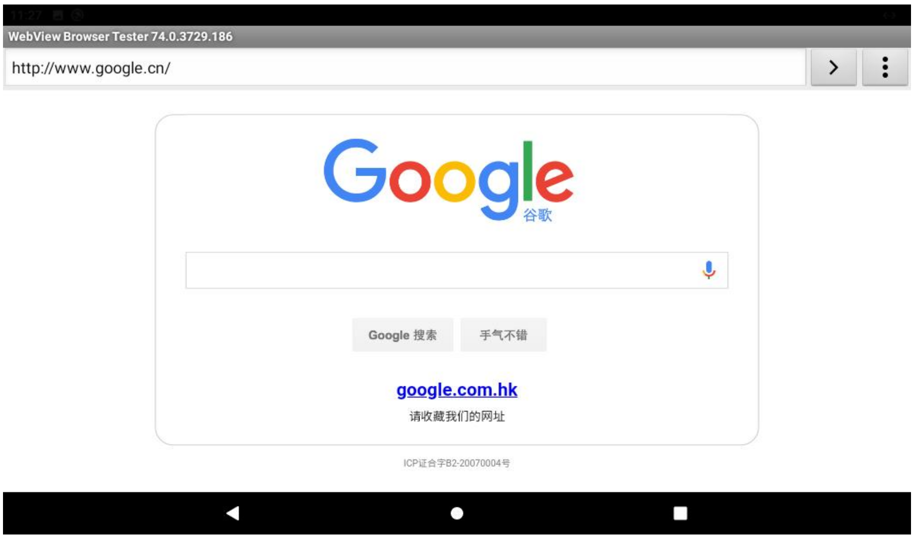
- **Ping Baidu checks whether it can be connected normally**

```
console:/ # ping www.baidu.com
PING www.a.shifen.com (14.215.177.38) 56(84) bytes of data.
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=1 ttl=55 time=10.5 ms
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=2 ttl=55 time=14.6 ms
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=3 ttl=55 time=25.4 ms
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=4 ttl=55 time=10.5 ms
^C
```

```
--- www.a.shifen.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 7ms
rtt min/avg/max/mdev = 10.469/15.227/25.357/6.085 ms
```

- **WI-FI access to Baidu**

Open a browser on the desktop and enter the address of Baidu



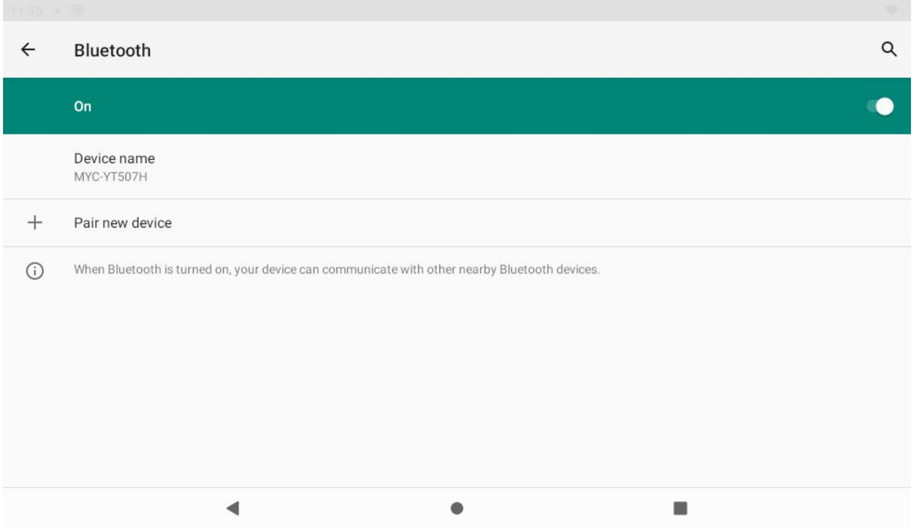
WebView Browser Tester 74.0.3729.186

http://www.google.cn/

Figure 4-4 WI-FI access to Baidu

3). Bluetooth test

Enter the settings interface, tap the Bluetooth option, tap Preferences, tap Bluetooth, and then turn on Bluetooth, you can search for nearby Bluetooth devices.



Bluetooth

On

Device name
MYC-YT507H

Pair new device

When Bluetooth is turned on, your device can communicate with other nearby Bluetooth devices.

Figure 4-5 Bluetooth is turned on

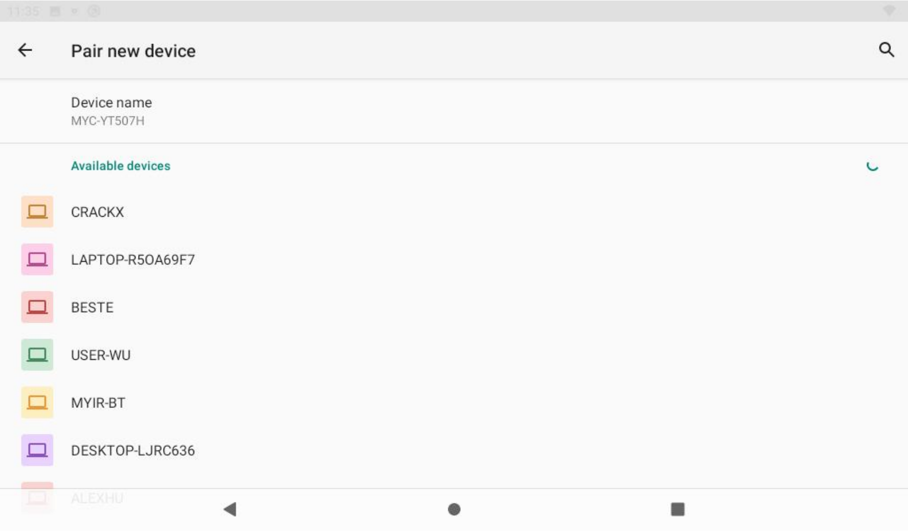
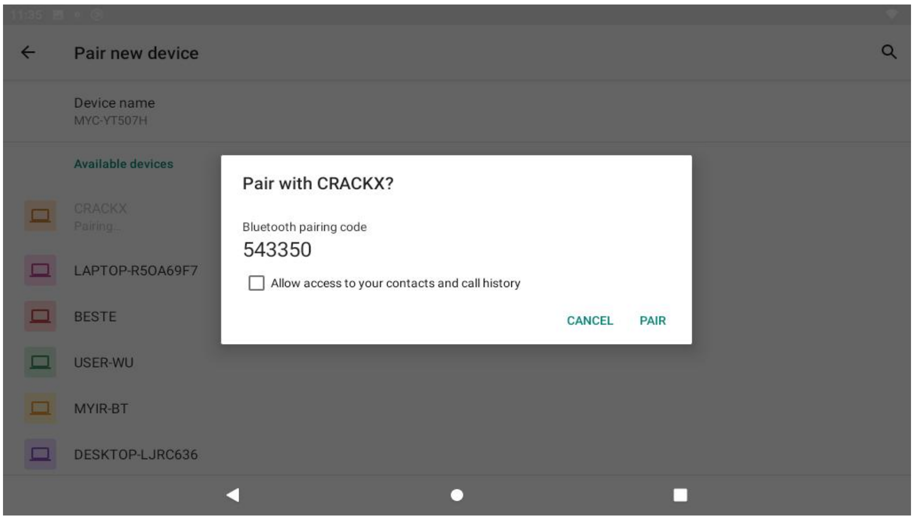


Figure 4-6 Searching for Bluetooth devices

Select the Bluetooth device you want to connect to and tap Pair

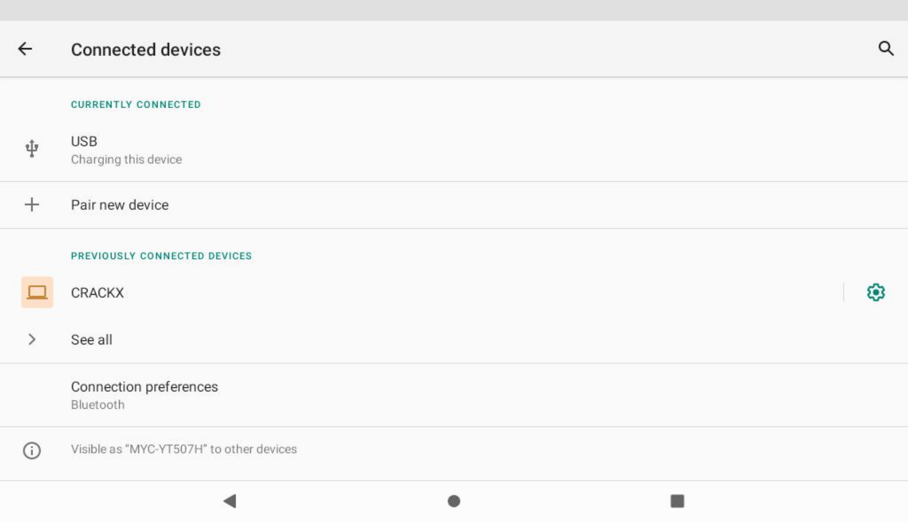


Pair with CRACKX?

Bluetooth pairing code
543350

☐ Allow access to your contacts and call history

Figure 4-7 Bluetooth pairing



Visible as "MYC-YT507H" to other devices

Figure 4-8 Bluetooth connection

4.2. 4G module

The LINUX device can also be connected to an external 4G module for dial-up Internet access, the 4G module uses QUECTEL's EC20 module, the test module is EC20 CEFDKG.

1). View VIDs and PIDs

Connect the EC20 module to the development board and use lsusb to view the EC20 module information.

```
[root@myir:/]#  lsusb
Bus 005 Device 001: ID 1d6b:0001
Bus 003 Device 001: ID 1d6b:0002
Bus 002 Device 002: ID 2c7c:0125
Bus 001 Device 001: ID 1d6b:0002
Bus 006 Device 001: ID 1d6b:0001
Bus 004 Device 001: ID 1d6b:0001
Bus 002 Device 001: ID 1d6b:0002
```

- 2c7c:0125: VID and PID information for EC20.

The array static const struct usb_device_id option_ids in ${KERNEL_DIR}/drivers/usb/serial/option.c is required The configuration is as follows:

```
#define QUALCOMM_VENDOR_ID                    0x05C6
#define QUECTEL_PRODUCT_EC25                  0x0125
static const struct usb_device_id option_ids[] = {
    --snip--
      { USB_DEVICE(QUECTEL_VENDOR_ID, QUECTEL_PRODUCT_EC25),
        . driver_info = RSVD(4) },
    --snip--
}
```

These configurations need to be turned on in the kernel:

```
+CONFIG_PPP=y
+CONFIG_PPP_BSDCOMP=y
+CONFIG_PPP_DEFLATE=y
+CONFIG_PPP_FILTER=y
+CONFIG_PPP_MPPE=y
+CONFIG_PPP_MULTILINK=y
+CONFIG_PPPOE=y
+CONFIG_PPP_ASYNC=y
```

```
+CONFIG_PPP_SYNC_TTY=y
+CONFIG_SLHC=y
```

2). View the kernel recognition module

If the kernel adds the VID and PID configuration of this module, the /dev/ttyUSB* node is generated:

```
[root@myir:/]#   ls  -l  /dev/ttyUSB*
crw-------       1  root         root          188,  0  Jan  1  00:00  /dev/ttyUSB0
crw-------       1  root         root          188,  1  Jan  1  00:00  /dev/ttyUSB1
crw-------       1  root         root          188,  2  Jan  1  00:00  /dev/ttyUSB2
crw-------       1  root         root          188,  3  Jan  1  00:00  /dev/ttyUSB3
```

3). Use a 4G signal to surf the Internet

Slide out of the status bar from the top, turn on the mobile data switch, and then open the browser to enter the Baidu page

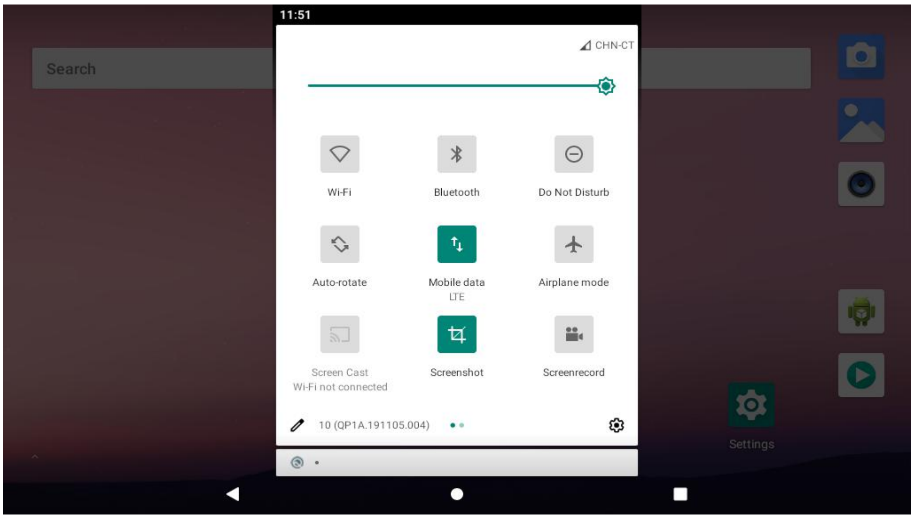
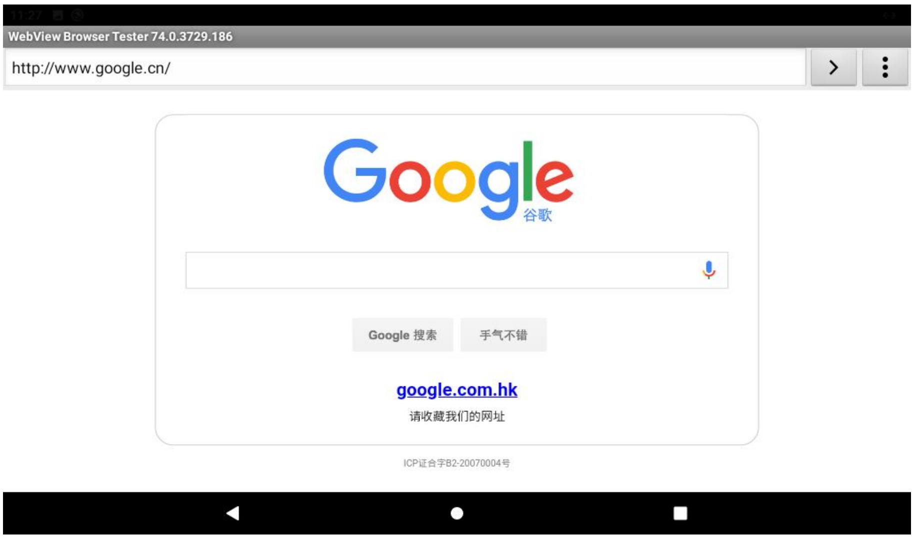


Figure 4-9 Status bar



http://www.google.cn/

google.com.hk

请收藏我们的网址

ICP证合字B2-20070004号

Figure 4-10 4G browsing Baidu

5. Web applications

5.1. PING

PING is primarily used to test network connectivity, but is also used to test network latency and packet loss. Once the Ethernet connection is configured, the network connection can be tested simply using PING.

1). Wiring and information output

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

```
console:/ # [ 9689.127024] sunxi-gmac gmac0 eth0: Link is Up - 1Gbps/Full - flow control rx/tx
```

2). Test Ping Internet URL

```
console:/ # ping www.baidu.com
PING www.a.shifen.com (14.215.177.39) 56(84) bytes of data.
64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=1 ttl=56 time=7.09 ms
64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=2 ttl=56 time=6.86 ms
64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=3 ttl=56 time=6.34 ms
64 bytes from 14.215.177.39 (14.215.177. 39): icmp_seq=4 ttl=56 time=7.06 ms
--- www.a.shifen.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 7ms
rtt min/avg/max/mdev = 6.344/6.839/7.094/0.311 ms
```

Note: Ping the Internet requires that DNS work properly.

The above result shows that the IP address of www.baidu.com after domain name resolution is 14.215.177.39, ICM p_seq represents the number of the ICMP packet, if the number is consecutive, there is no packet loss; time represents the delay time of the response, although the shorter this time the better. In addition to testing Ethernet, the ping command can also be used to test Wi-Fi.

5.2. Iptables

iptables is a management tool for IPv4 packet filtering and NAT. It is used to set up, maintain, and check the table of IP packet filtering rules in the Linux x kernel. You can define several different tables. 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 how matching packets are handled. Development boards that use Linux systems typically use the iptables tool to configure firewalls. iptables processes various packets according to the methods defined by packet filtering rules, such as accept, reject, and drop.

The following uses iptables to test intercepting ICMP packets and preventing other external devices on the network from pinging them. For specific commands, see https://linux.die.net/man/8/iptables.

1). Configure the board iptables

Use iptables configuration on the board to drop input ICMP packets and not respond to ping probes from other hosts, with the following 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 board on the development host and specify deadline as 10, the result is as follows:

```
C:\Users\Lenovo>ping 192.168.1.195 -w 10
 Pinging 192.168.1.195 with 32 bytes of data:
 Ping statistics for 192.168.1.195:
Packets: Sent = 4, Received = 0, Lost  = 4 (100% loss).
C:\Users\Lenovo>
C:\Users\Lenovo>ping 192.168.1.97 -w 10
 Pinging 192.168.1.97 with 32 bytes of data:
```

```
The request timed out.
The request timed out.
The request timed out.
The request timed out.
 Ping statistics for 192.168.1.97:
Packets: Sent = 4, Received = 0, Lost  = 4 (100% loss).
```

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

- **Delete the corresponding firewall rule**

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

- **Test the ping board again**

```
C:\Users\Lenovo>ping 192.168.1.97 -w 10
 Pinging 192.168.1.97 with 32 bytes of data:
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 = 4, Received  = 4, Lost = 0 (0% Loss).
Estimated time of round trip in milliseconds:
Minimum = 0ms, Maximum = 0ms, Average  = 0ms
```

After clearing the iptables rule, ping the board from the development host again is ready to ping. The above example is just a simple demonstration, in fact, iptables can achieve very powerful functions with various rules, which will not be detailed here.

5.3. iperf

iperf is a tool that actively measures the maximum achievable bandwidth on an IP network. IT SUPPORTS ADJUSTING VARIOUS PARAMETERS SUCH AS TEST TIME, BUFFER SIZE, AND PROTOCOL (TCP, UDP, SCTP UNDER IPV4 AND IPV6). iperf can be divided into server-side mode or client mode by role, we can use it to test and view the network bandwidth in TCP mode, TCP window value, probability of retransmission, etc. You can also test packet loss, latency, and jitter over specified UDP bandwidths.

We open the Windows PowerShell on the development host, the host with the Gigabit network card is the server of iperf, and the tested development board is used as the client to test the development board network card separately Performance of TCP and UDP. Start by installing iperf on the host as follows:

Connect the server and the board directly through a CAT6 network cable and configure their respective IP addresses. For example, we set the server IP to 192.168.40 9. Set the development board IP to 192.168.40.207, and use the ping command test to ensure that they are connected. The following test is based on the Gigabit network port eth0, and the test method of 100 Gigabit network port is consistent with that of Gigabit.

Note: Try not to connect the router or exchange machine, the network cable used is preferably a shielded network cable, so as not to be affected by the transmission and forwarding of intermediate devices.

1). Test TCP performance

- **Server（192.168.4 9）**

iperf on the server uses the -s parameter to indicate that it works in server-side mode.

```
Server (192.168.40.9).
nico@system1-ubuntu1804:~$ iperf3 -s
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

- **Client（192.168 40.207）**

The iperf program running on the board works in client-side, TCP mode, where the parameters are described as follows:

  - -c 192.168.40.207: Working on client, connecting to server 192.168.40 9
  - -i 2 : Test results are reported at intervals of 2 seconds
  - -t 10 : The total test duration is 10 seconds

```
console:/ # iperf -c 192.168.40.9
Connecting to host 192.168.40.200, port 5201
[  5] local 192.168.40.100 port 35060 connected to 192.168.40.200 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec  65.8  MBytes   552 Mbits/sec   19     168 KBytes
[  5]   1.00-2.00   sec  86.6  MBytes   726 Mbits/sec    7     164 KBytes
[  5]   2.00-3.00   sec  86.3  MBytes   724 Mbits/sec    5     167 KBytes
[  5]   3 .00-4.00   sec  87.1 MBytes   730 Mbits/sec    3     187 KBytes
[  5]   4.00-5.00   sec  87.1  MBytes   731 Mbits/sec    6     158 KBytes
[  5]   5.00-6.00   sec  86.6  MBytes   726 Mbits/sec    9     201 KBytes
[  5]   6.00-7.00   sec  86.7  MBytes   727 Mbits/sec    4     211 KBytes
[  5]   7.00-8.00   sec  86.4  MBytes   725 Mbits/sec    3     211 KBytes
[  5]   8.00-9.00   sec  86.9  MBytes   729 Mbits/sec    5     209 KBytes
[  5]   9.00-10.00  sec  86.6  MBytes   726 Mbits/sec    7     187 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec   846  MBytes   710 Mbits/sec   68             sender
[  5]   0.00-10.00  sec   845  MBytes   709 Mbits/sec                  receiver
iperf Done.
```

After 10 seconds, the client ends the test and displays the above test results, indicating that the TCP bandwidth is about 92 Mbits with no retransmission, and the TCP window value is 5 during the test MBytes. At the same time, the server also displays the test results as follows, and then continues to listen on the port and wait for the client to connect:

```
Accepted connection from 192.168.40.207, port 49436
[  5] local 192.168.40.9 port 5201  connected to 192.168.40.207 port 49438
[ ID] Interval           Transfer     Bandwidth
[  5]    0.00-1.00    sec  10.3  MBytes  586.1 Mbits/sec
[  5]    1.00-2.00    sec  11.2  MBytes  723.9 Mbits/sec
[  5]    2.00-3.00    sec   11.2 MBytes  734.0 Mbits/sec
[  5]    3.00-4.00    sec  11.2  MBytes  733.7 Mbits/sec
[  5]    4.00-5.00    sec  11.0  MBytes  722.5 Mbits/sec
[  5]    5.00-6.00    sec  9.60  MBytes  720.5 Mbits/sec
[  5]    6.00-7.00    sec  11.2  MBytes  733.8 Mbits/sec
[  5]    7.00-8.00    sec  10.3  MBytes  726.7 Mbits/sec
[  5]    8.00-9.00    sec  10.8  MBytes  730.8 Mbits/sec
[  5]    9.00-10.00   sec  11.0  MBytes  722.6 Mbits/sec
[  5]  10.00-10.03    sec   363  KBytes  796.7 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth
[  5]    0.00-10.03   sec  0.00 Bytes  0.00 bits/ sec                  sender
[  5]    0.00-10.03   sec     1.08 GBytes  730.5 Mbits/sec             receiver
-----------------------------------------------------------
```

2). Test UDP performance

- **Server (192.168 40.9)**

The server continues to run iperf3 using the -s parameter to indicate that it is working in server-side mode.

```
nico@system1-ubuntu1804:~$ iperf3 -s
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

- **Client (192.168 40. 207）**

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

- ➢ -u : Works in UDP mode
- ➢ -c 192.168.40. 207 : Working on client, connecting to server 192.168.40. 9
- ➢ -i 2 : Test results are reported at intervals of 2 seconds

- -t 10 : The total test duration is 10 seconds
- -b 100M : Set the UDP transmission bandwidth to 100Mbps

```
console:/ # iperf -c 192.168.40.9 -u -b 100M
Connecting to host 192.168.40.9, port 5201
[  5] local 192.168.40.196 port 34954 connected to 192.16 8.40.9 port 5201
[ ID] Interval           Transfer     Bitrate         Total Datagrams
[  5]   0.00-2.00   sec  22.9 MBytes  96.0 Mbits/sec  16572
[  5]   2.00-4.00   sec  22.8 MBytes  95.7 Mbits/sec  16524
[  5]   4.00-6.00   sec  22.8 MBytes  95.7 Mbits/sec  16523
[  5]   6.00-8.00   sec  22.8 MBytes  95.6 Mbits/sec  16510
[  5]   8.00-10.00  sec  22.8 MBytes  95.7 Mbits/sec  16523
[  5]  10.00-12.00  sec  22.8 MBytes  95.7 Mbits/sec  16524
[  5]  12.00-14.00  sec  22.8 MBytes  95.7 Mbits/sec  16523
[  5]  14.00-16.00  sec  22.8 MBytes  95.7 Mbits/sec  16524
[  5]  16.00-18.00  sec  22.8 MBytes  95.7 Mbits/sec  16523
[  5]  18.00-20.00  sec  22.8 MBytes  95.7 Mbits/sec  16523
[  5]  20.00-22.00  sec  22.8 MBytes  95.7 Mbits/sec  16524
[  5]  22.00-24.00  sec  22.8 MBytes  95.6 Mbits/sec  16508
[  5]  24.00-26.00  sec  22.8 MBytes  95.7 Mbits/sec  16524
[  5]  26.00-28.00  sec  22.8 MBytes  95.7 Mbits/sec  16523
[  5]  2 8.00-30.00  sec  22.8 MBytes  95.7 Mbits/sec  16523
[  5]  30.00-32.00  sec  22.8 MBytes  95.7 Mbits/sec  16524
^C[  5]  32.00-33.51  sec  17.2 MBytes  95.7 Mbits/sec  12440
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagra
ms
[  5]   0.00-33.51  sec   382 MBytes  95.7 Mbits/sec  0.000 ms  0/276835
(0%)  sender
[  5]   0.00-33.51  sec  0.00 Bytes  0.00 bits/sec  0.000  ms  0/0 (0%)  recei
ver
```

After 10 seconds, the test end of the client and display the above test results, indicating that UDP has no packet loss when the specified bandwidth is 100Mbps.

At the same time, the server also displays the test results as follows, and then continues to listen on port 5201 and wait for the client to connect:

```
Accepted connection from 192.168.40.207, port 49440
[  5] local 192.168.40.9 port 5201 connected to 192.168.40.207 port 33925
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total Datag
rams
[  5]   0.00-1.00   sec  3.76  MBytes  96.5 Mbits/sec  0.097 ms  102/2822 (3.
6%)
[  5]   1.00-2.00   sec  4.51  MBytes  95.8 Mbits/sec  0.390 ms  6/3272 (0.1
8%)
[  5]   2.00-3.00   sec  3.42  MBytes  95.7 Mbits/sec  0.079 ms  14/2493 (0.
56%)
[  5]   3.00-4.00   sec  3.35  MBytes  95.1 Mbits/sec  0.098 ms  5/2432 (0.2
1%)
[  5]   4.00-5.00   sec  6.02  MBytes  95.5 Mbits/sec  0.237 ms  1747/6104
(29%)
[  5]   5.00-6.00   sec  4.13  MBytes  95.6 Mbits/sec  0.101 ms  42/3033 (1.
4%)
[  5]   6.00 -7.00   sec  3.76 MBytes  95.5 Mbits/sec  0.089 ms  98/2819 (3.
5%)
[  5]   7.00-8.00   sec  4.14  MBytes  95.7 Mbits/sec  0.097 ms  65/3062 (2.
1%)
[  5]   8.00-9.00   sec  3.59  MBytes  95.1 Mbits/sec  0.108 ms  126/2724 (4.
6%)
[  5]   9.00-10.00  sec  4.14  MBytes  95.7 Mbits/sec  0.088 ms  119/3117
(3.8%)
[  5]  10.00-10.02  sec  32.5  KBytes  95.8 Mbits/sec  0.902 ms  0/23 (0%)
- - - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total Datag
rams
[  5]   0.00-10.02  sec  0.00 Bytes  0.00 bits/sec  0.902  ms  2324/31901 (7.
3%)
---------------------------------------------------------
```

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

iperf3 has many parameters that can be configured during the testing process, and users can adjust the test according to the actual application needs. For example, you can increase the value of the -t parameter for long-term stress testing, or specify the -P parameter for stress testing with multiple connections. For more information about iperf3 testing, please refer to: https://iperf.fr/iperf-doc.php#3doc.

6. Multimedia applications

6.1. Camera

1). MIPI 5640 camera test

Slide up the app interface, open the DVR app, select a single independent camera, select 0 channels, and confirm.

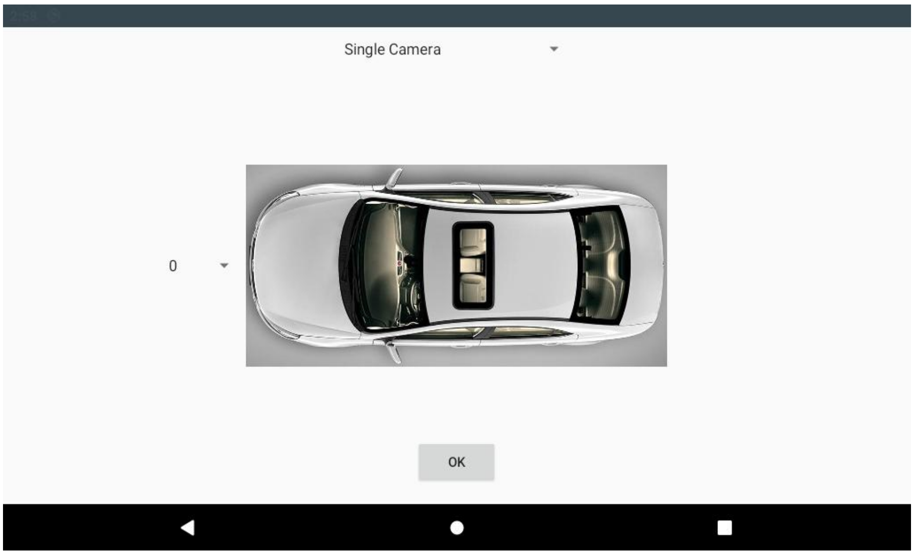


Figure 6-1 DVR application interface

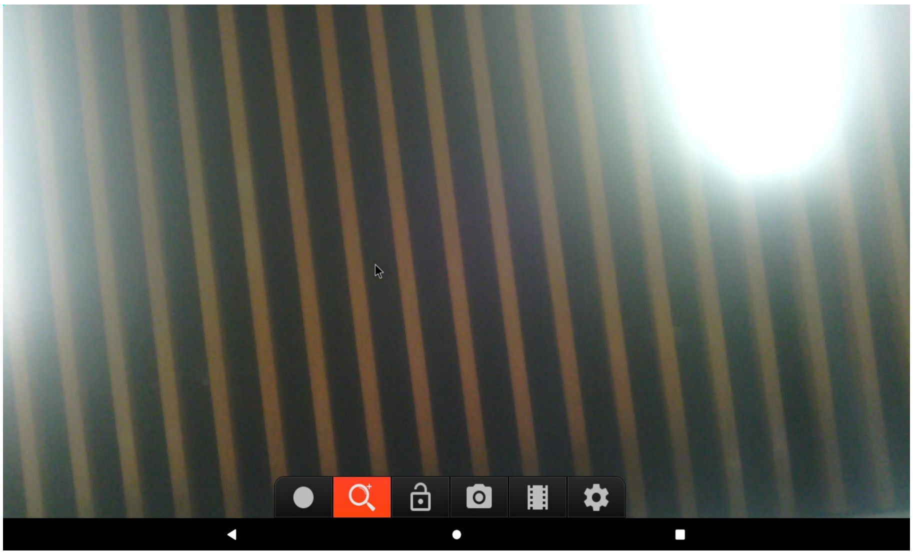


Figure 6-2 MIPI camera preview interface

Tap the button to ● record and the button to 📷 take a photo.

2). DVP 5640 camera test

The starting steps are the same as the MIPI above, except that the channel selection is 2, and the photo and video functions are also the same as MIPI.

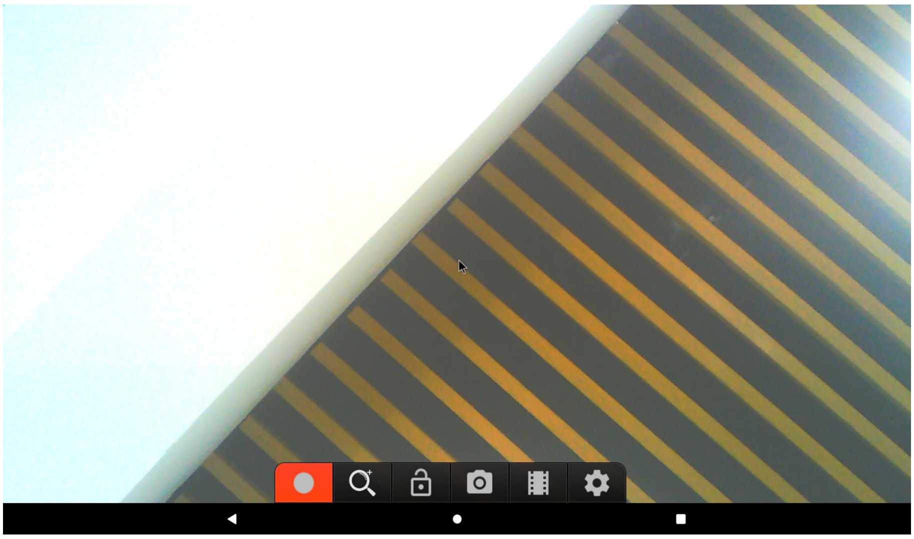
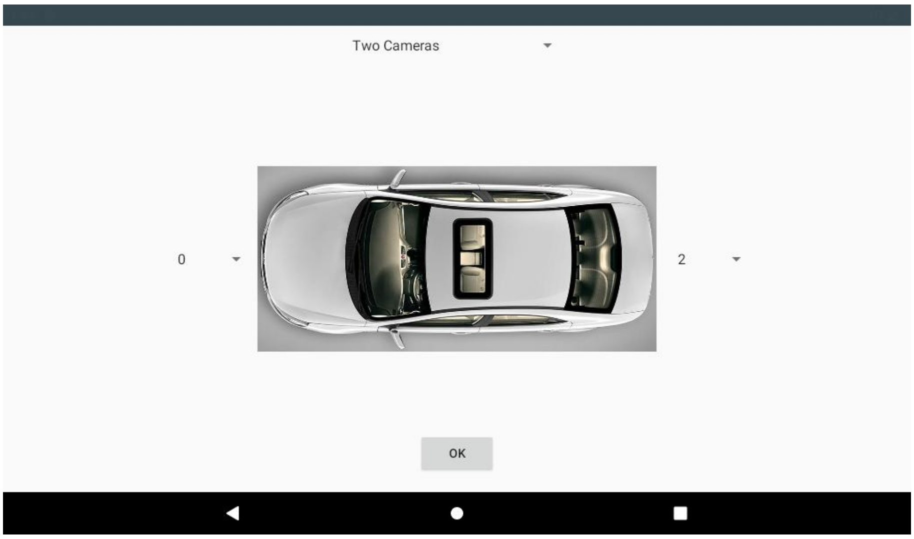


Figure 6-3 DVP camera preview interface

3). MIPI 5640 and DVP 5640 dual camera testing

First open the DVR app, select two independent cameras, select 0 for one channel, 2 for one channel, and then confirm. The photo and video functions are used in the same way as above.



Two Cameras

0

2

Figure 6-4 Dual camera selection interface

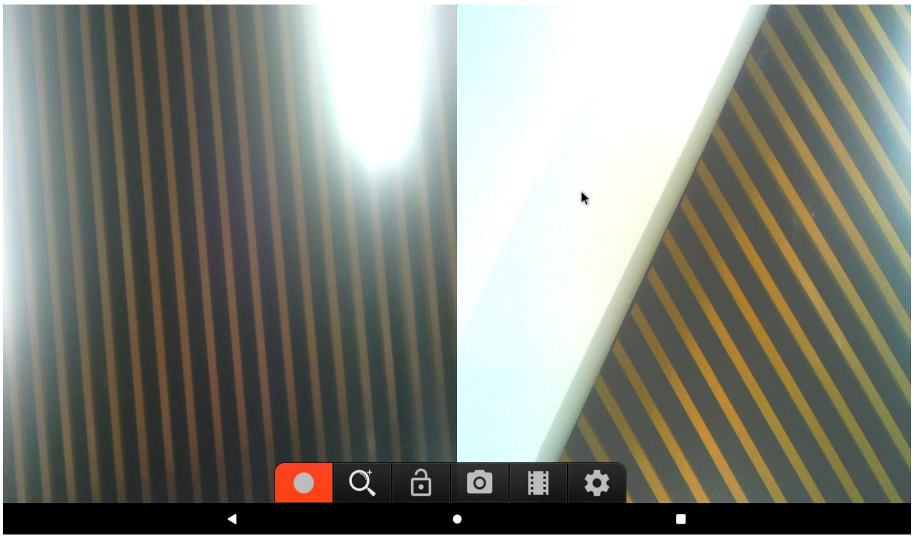


Figure 6-5 Dual-camera preview interface

6.2. Video test

Open the video player on your desktop and select the video you want to play. Support video playback in multiple formats, 1080P60 frame video, 4K30 frame video, V9 video, H264/H265 video. (Put the downloaded video in advance into the /storage/emulated/0/Movies path).

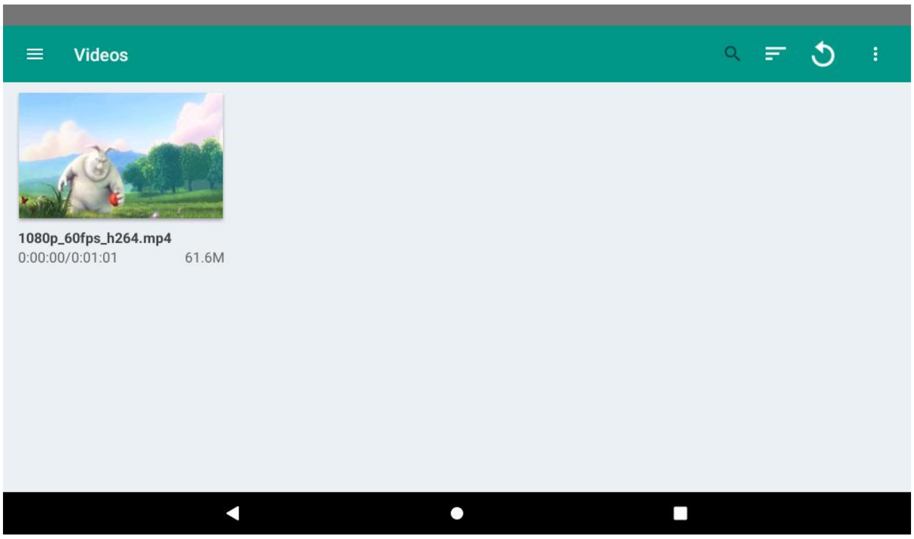


Figure 6-6 Video player interface

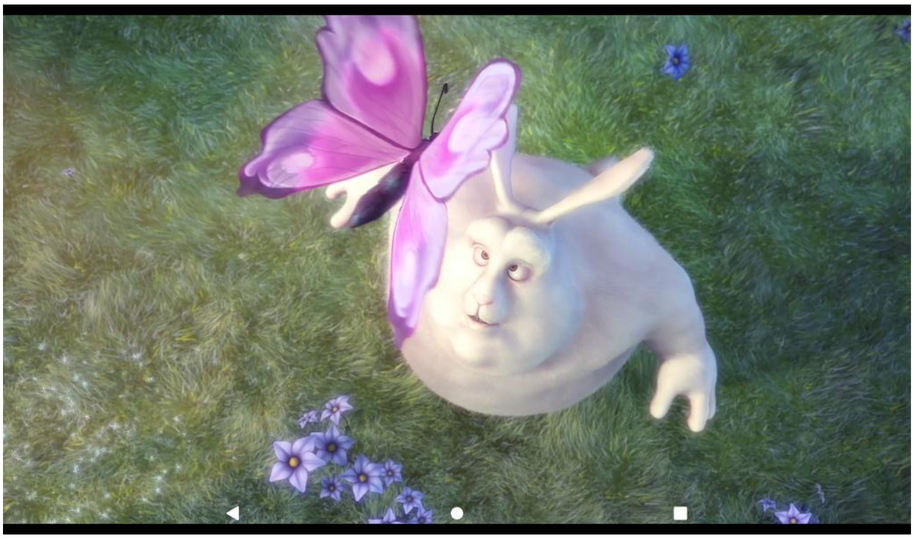


Figure 6-7 Video playback

6.3. Audio test

This section is to test playing audio. There are three ways, namely Audio HeadPhone, Audio LineinOut and Audio SPDIF, first place the audio file in the /storage/emulated/0/Music directory.

1). Use the tool to play

- **HeadPhone**

Use the tinyplay playback test tool. Audio playback device node for operating HDMI (I2S1), I2S2, I2S3.

```
Usage: tinyplay file.wav [-aD ahub card] [-ad ahub device] [-D card] [-d devic
e] [-p period_size] [-n n_periods]
```

In the MYD-YT507H development board, we plug the playback device into the J16 dock and execute the following command to play the audio.

```
console:/storage/emulated/0/Music # tinyplay 1.wav -aD 3 -ad 0
Playing sample: 2 ch, 44100 hz, 16 bit 35028336 bytes
```

- **LineinOut**

Use the tinyplay playback test tool. Audio playback device node for operating audiocedec, SPDIF, USB audio.

```
Usage: tinyplay file.wav [-D card] [-d device] [-p period_size] [-n n_periods]
```

In the MYD-YT507H development board, we plug the playback device into the J17 base and execute the following command to play audio.

```
console:/storage/emulated/0/Music #tinyplay 1.wav -D 0 -d 0
Playing sample: 2 ch, 44100 hz, 16 bit
```

- **SPDIF**

When playing SPDIF audio, the tinyplay playback test tool is also used. We plug the playback device into the J11 dock and execute the following command to play the audio.

```
console:/storage/emulated/0/Music # tinyplay 1.wav -D 1 -d 0
Playing sample: 2 ch, 44100 hz, 16 bit
```

2). Play with ES Media Player

Open ES File Explorer, find music in the home directory to open, select the music you want to play, select ES media player to open.

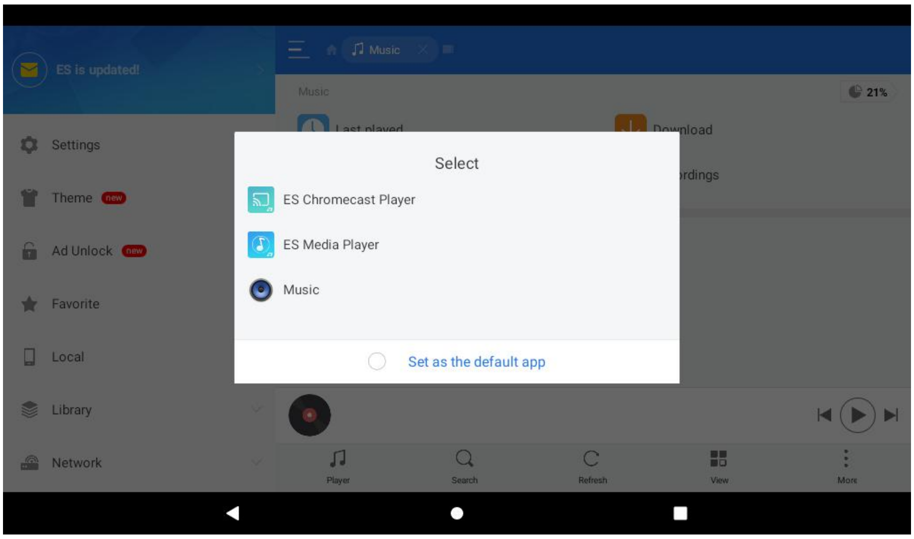


Figure 6-8 Player selection

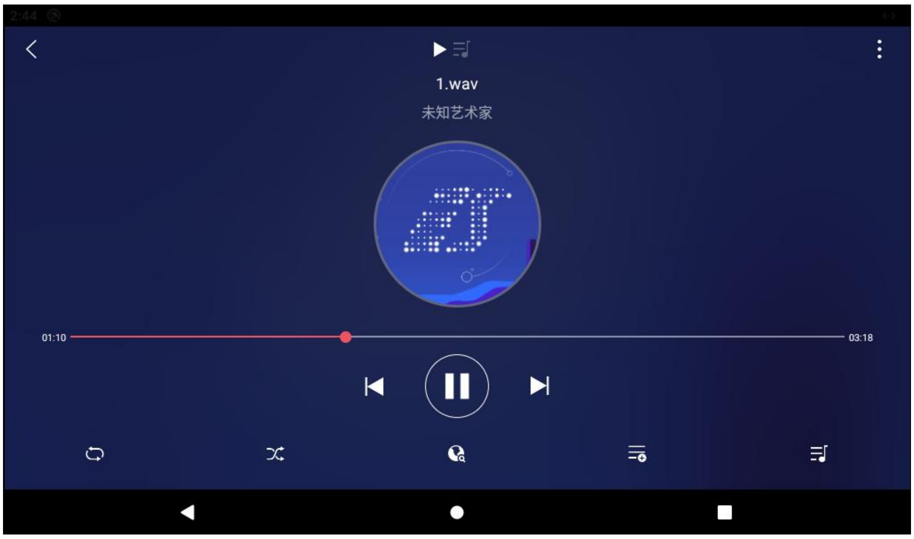
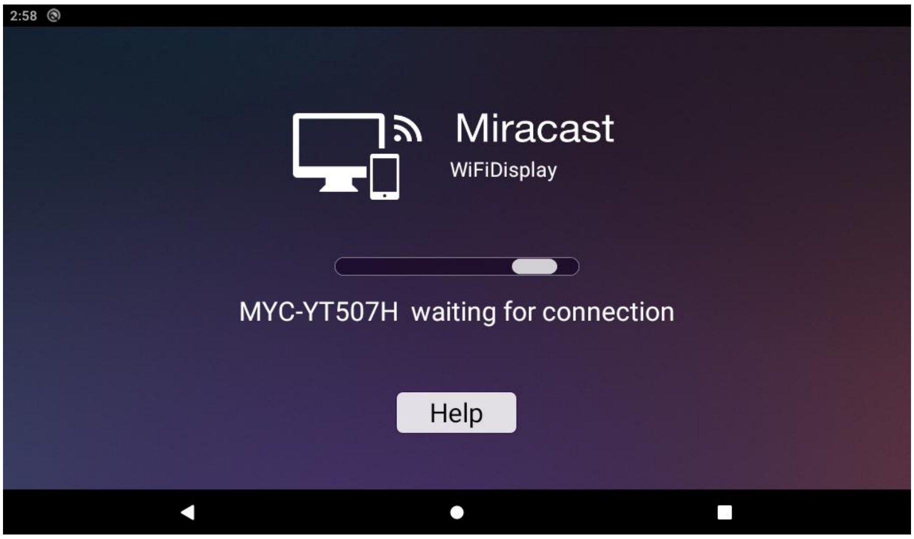


Figure 6-9 Music playback

6.4. Miracast casting

To use this function, the development board needs to turn on the WI-FI function, and you also need a mobile phone that supports wireless screen mirroring, and this test uses a Xiaomi mobile phone test. Before using this function, the phone and the development board need to be connected to the same WI-FI, then slide out of the application interface, open the Miracast app, and wait for the connection.



MYC-YT507H waiting for connection

Figure 6-10 Miracast connection interface

Turn on the mobile phone screen mirroring function and then need to wait for a while, let the mobile phone search for our development board device, search for the device and click connect. After the connection is successful, the phone's screen will be displayed on the development board in real time

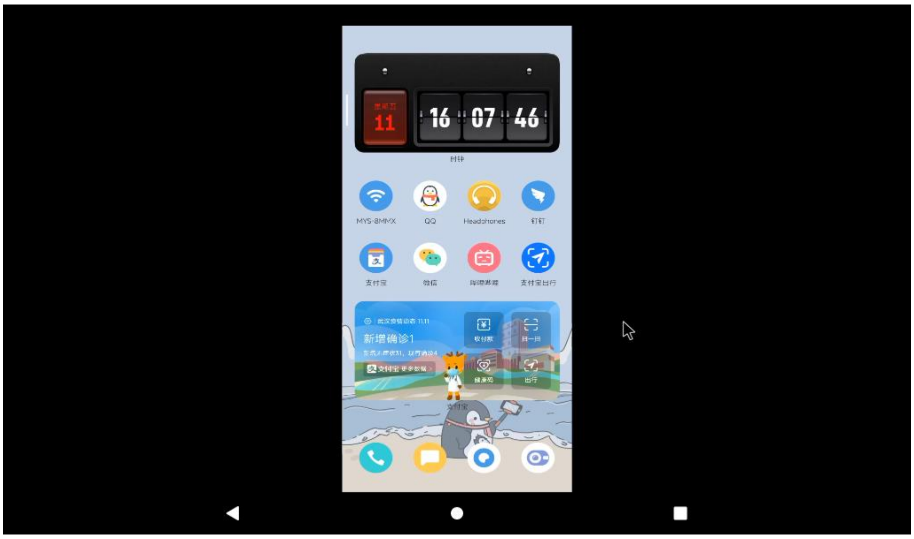


Figure 6-12 Mobile phone screen mirroring

7. Resources

- **All-in-one community development**

  https://www.aw-ol.com
- **Systemd network configuration**

  https://www.freedesktop.org/software/systemd/man/systemd.network.html
- **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