MYD-YT113X Linux

Software Evaluation Guide

File status: [ ] Draft [√]Release	**FILE ID:**	MYIR-MYD-YT113X-SW-EG-EN-L5.4.61
	VERSION:	V2.0[Doc]
	AUTHOR:	MSW0307
	RELEASE:	2024-08-09
	UPDATED:	2024-07-28

Revision History

VERSION	AUTHOR	PARTICIPANT	DATE	DESCRIPTION
V1.0[Doc]	MSW0202	MSW0019	2023-05-01	Initial version Applicable to the Longan version
V1.1[Doc]	MSW0202	MSW0019	2023-08-30	Add applicable model MYD-YT113-I, remove RS232 testing, add RS485 testing Applicable to the Longan version
V2.0[Doc]	MSW0307	MSW0041	2024-08-09	Suitable for Tina 5.0

CONTENT

1. Overview

The Linux Software Evaluation Guide is designed to introduce the testing procedures and evaluation methods for core resources and peripheral resources running on the open-source Linux system on the MYiR development board. This document serves as an initial assessment guide and can also be used as a general testing manual for system development.

1.1. Hardware Resources

The MYD-YT113X board from MYiR Electronics consists of a core board MYC-YT113X and a base board MYB-YT113X. The core board and the base board are soldered with stamp holes. In addition MYIR provides a wealth of software resources as well as documentation. The following table summarizes the MYD-YT113X chip models and their main features:

Table 1-1. Board Model Introduction

Core Chip Model	Processor Core	Core Board	Base Board
YT113i	Dual-core ARM Cortex-A7	MYC-YT113i-4E256D MYC-YT113i-4E512D MYC-YT113i-8E512D MYC-YT113i-8E1D	MYD-YT113i-4E256D MYD-YT113i-4E512D MYD-YT113i-8E512D MYD-YT113i-8E1D

For detailed hardware configuration parameters, please refer to the "*MYD-YT113X Product Manual*" document. During the evaluation and testing process, users may utilize some optional modules, listed below in Table 1-2.

Table 1-2. Optional Modules

Accessories	Interface Type	Explanation and Link
LCD Screen	Lvds interface	7-inch LVDS screen： https://www.myirtech.com/list.asp?id=634
Expansion Board Module	Raspberry Pi Interface	MY-WiredCom Module： https://www.myirtech.com/list.asp?id=665
4G module	PCIE interface	EM05-CE
5G module	PCIE interface	RM500Q

1.2. Software Resources

The BSP of MYD-YT113X development board is based on the official open-source community version Linux BSP port and modification of Allwinner, and the system image is built using the buildroot project. Bootloader, Kernel, and various software resources of the file system are all open in the form of source code. For specific content, please refer to the "*MYD-YT113X SDK Release Note*".

The development board has already burned an image based on the core board model at the factory, and you only need to power it on to use it..

1.3. Documentation Resources

According to the different stages of users using the development board, the SDK contains documentation for each stage, release notes, evaluation guides, development guides, application notes, frequently asked questions and answers, and other different categories of documents and manuals. For a list of specific documents, see the descriptions in Table 2-4 of "*MYD-YT113X SDK Release Note*".

1.4. Environment Preparation

Before you start evaluating the development board software, you need to make some necessary preparations for the board and configure some basic environments, including proper hardware wiring, configuring debugging serial ports, setting up startup, and other steps. 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 Linux, as well as self-developed applications. The software evaluation guide is divided into several sections to describe, 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 steps for each part of the resources.

1.5. Resource List

Table 1-3. Evaluation List

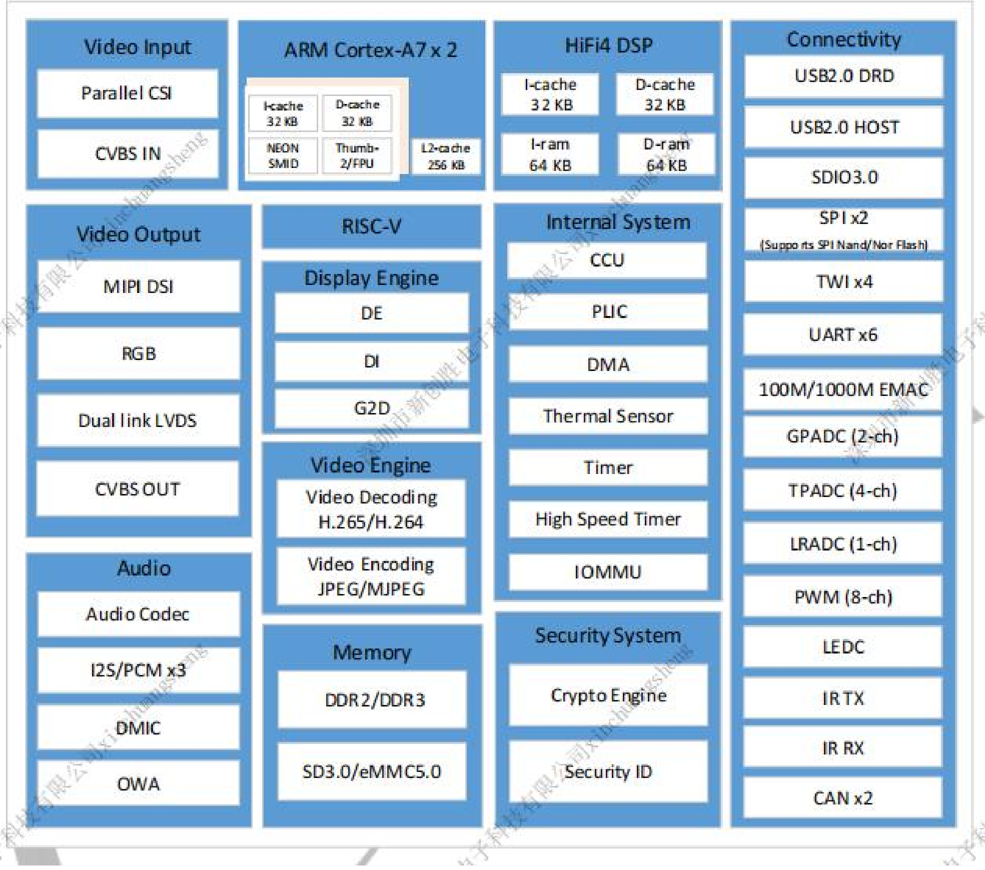
Sequence	Functions	Notes
1	CPU	
2	eMMC	
3	RTC	
4	Watchdog	Internal and External Watchdog Test
5	GPIO	
6	LED	
7	KEY	
8	USB	
9	OTG	
10	SD	
11	GPADC、TPADC	
12	LVDS	
13	Touch Panel	
14	Audio	
15	RS485	
16	CAN	
17	Ethernet	
18	WIFI	
19	4G/5G	

2. Core Resources

In Linux 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. The parameters of core resources such as CPU, memory, eMMC, etc. will be read and tested in detail below.

2.1. CPU

MYD-YT113X series development board equipped with Arm ® Cortex ® - A7 * 2 processor, with a maximum clock speed of up to 1.2GHz.



Figur 2-1. CPU Resource Diagram

The status or functions of the CPU will be tested by the following commands.

2.1.1. Command to view CPU information:

You can get the vendor and parameter information of the CPU in the system by reading the "*/proc/cpuinfo*" file.

```
root@myd-yt113-i:~# cat /proc/cpuinfo
processor        : 0
model name       : ARMv7 Processor rev 5 (v7l)
BogoMIPS         : 48.00
Features         : half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idiv
t vfpd32 lpae
CPU implementer : 0x41
CPU architecture: 7
CPU variant      : 0x0
CPU part         : 0xc07
CPU revision     : 5

processor        : 1
model name       : ARMv7 Processor rev 5 (v7l)
BogoMIPS         : 48.00
Features         : half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idiv
t vfpd32 lpae
CPU implementer : 0x41
CPU architecture: 7
CPU variant      : 0x0
CPU part         : 0xc07
CPU revision     : 5

Hardware         : Generic DT based system
Revision         : 0000
Serial           : 0000000000000000
```

- Processor: The number of logical processing cores in the system can be the physical cores or the logical cores virtualized by using the Hyper-Threading technology for multi-core processors.
- Model name: The name and number to which the CPU belongs

- BogoMIPS: Rough estimate of CPU's million instructions per second (Million Instructions Per Second) calculated when the system kernel is started.

2.1.2. View CPU utilization

Perform the following operations on the T113 series chip to view CPU usage:

```
root@myd-yt113-i:~# top
Mem: 129480K used, 100680K free, 1188K shrd, 5432K buff, 76596K cached
CPU:  52% usr   0% sys   0% nic  47% idle   0% io   0% irq   0% sirq
Load average: 1.02 1.01 0.95 2/73 1739
  PID  PPID USER     STAT   VSZ %VSZ %CPU COMMAND
 1644     1 root     S     121m  54%  53% mxapp2
 1569     1 root     S    28176  12%   0% adbd
 1589     1 root     S     4912   2%   0% /usr/sbin/sshd
 1655     1 nobody   S     4528   2%   0% /usr/sbin/dnsmasq
 1670     1 root     S     2936   1%   0% -sh
 1418     1 root     S     2440   1%   0% /sbin/udevd -d
 1499     1 root     S     2392   1%   0% dbus-daemon --system
 1415     1 root     S     2204   1%   0% /sbin/klogd -n
 1411     1 root     S     2204   1%   0% /sbin/syslogd -n
    1     0 root     S     2204   1%   0% init
 1739  1670 root     R     2204   1%   0% top
 1594     1 root     S     2204   1%   0% /usr/sbin/telnetd -F
 1531     1 root     S     2092   1%   0% /sbin/dhcpcd -f /etc/dhcpcd.conf
 1657     1 root     S     1352   1%   0% /sbin/swupdate-progress -w
   10     2 root     IW       0   0%   0% [rcu_preempt]
 1587     2 root     SW       0   0%   0% [irq/171-usb_id]
    7     2 root     IW       0   0%   0% [kworker/u4:0-ev]
    9     2 root     SW       0   0%   0% [ksoftirqd/0]
 1682     2 root     SW       0   0%   0% [usb-storage]
 1736     2 root     IW       0   0%   0% [kworker/1:2-eve]
 1478     2 root     IW<      0   0%   0% [kworker/0:2H-mm]
 1365     2 root     IW<      0   0%   0% [kworker/1:1H-kb]
 1737        2  root       IW          0    0%    0%  [kworker/1:1-eve]
 1216        2  root       SW          0    0%    0%  [irq/39-mmc2]
   15        2  root       SW          0    0%    0%  [ksoftirqd/1]
   34        2  root       IW          0    0%    0%  [kworker/0:1-eve]
```

- %usr: Represents the CPU usage of user space programs (not scheduled through nice)
- %sys: Represents the CPU usage of the system space, mainly the kernel program
- %nic: nic: Represents the CPU usage of user space programs that have been scheduled through nice
- %idle: Idle CPU
- %irq: Number of hard interrupts handled by the CPU
- %sirq: Number of soft interrupts handled by the CPU

2.1.3. Acquire CPU Temperature Information

The CPU has an integrated temperature sensor for CPU temperature collection, which can conveniently capture the internal temperature of the CPU.

```
root@myd-yt113-i:~# cat /sys/class/thermal/thermal_zone0/temp
44622
```

2.1.4. CPU Stress Test

```
root@myd-yt113-i:~# echo "scale=5000;4*a(1)" | bc -l -q &
[1] 1687
3.141592653589793238462643383279502884197169399375105820974944592307\
81640628620899862803482534211706798214808651328230664709384460955058\
22317253594081284811174502841027019385211055596446229489549303819644\
28810975665933446128475648233786783165271201909145648566923460348610\
45432664821339360726024914127372458700660631558817488152092096282925\
40917153643678925903600113305305488204665213841469519415116094330572\
.........
[1]+   Done                          echo "scale=5000;4*a(1)" | bc -l -q
```

The above command generates Pi in the background, accurate to 5000 decimal spots. The calculation process can take some time. In the meantime, we can check the change in CPU utilization with the top command, as shown below:

```
root@myd-yt113-i:~# top
Mem: 128616K used, 101544K free, 816K shrd, 4988K buff, 76000K cached
CPU:  95% usr   4% sys   0% nic   0% idle   0% io   0% irq   0% sirq
Load average: 1.16 0.80 0.39 3/71 1688
  PID  PPID USER      STAT    VSZ %VSZ %CPU COMMAND
 1687  1680 root      R      2260   1%  48% bc -l -q
```

After about 3 minutes, the result of PI is calculated. The CPU usage on this device is high, with no exceptions occurring, indicating that the CPU stress test can pass. By increasing the precision requirements, the test pressure can be further increased.

2.2. Memory

This time, we use the version of MYD-YT113X with 256MB of memory as an example. The system will divide the memory into device memory (CMA) and system memory (MEM). Device memory is a section of contiguous space for the driver to use, and system memory is allocated space for the user state.

2.2.1. To check memory information

Retrieve parameter details from the system,you can access the "*/proc/meminfo*" *file*.

```
rroot@myd-yt113-i:~# cat /proc/meminfo
MemTotal:           230160 kB
MemFree:            101744 kB
MemAvailable:       179740 kB
Buffers:            4988 kB
Cached:             76064 kB
SwapCached:              0 kB
Active:            30144 kB
Inactive:          72196 kB
Active(anon):      21376 kB
Inactive(anon):      776 kB
Active(file):      8768 kB
Inactive(file):   71420 kB
Unevictable:           0 kB
Mlocked:                0 kB
HighTotal:             0 kB
HighFree:              0 kB
LowTotal:          230160 kB
LowFree:           101744 kB
SwapTotal:             0 kB
SwapFree:              0 kB
Dirty:                0 kB
```

```
Writeback:              0  kB
AnonPages:          21292  kB
Mapped:             36768  kB
Shmem:                880  kB
KReclaimable:        5936  kB
Slab:               17428  kB
SReclaimable:        5936  kB
SUnreclaim:         11492  kBB
......
```

- MemTotal:The total system memory amount, measured in kilobytes (KB). Since Nuvoton officially reserves memory for linuxfb display and vc8k, and both OP-TEE and the kernel also occupy DDR,only 144MB of memory is displayed at this time.
- MemFree: Amount of free memory available, indicating currently unused memory.
- MemAvailable: Estimated amount of available memory, taking into account the kernel's memory management strategy.
- Buffers: Memory used for file system buffers, i.e., buffers used for file read/write operations. In this example, buffers occupy 9,500 KB.
- Cached: Memory used for file system cache, including cached files and directory data.
- SwapCached: Amount of memory that has been cached to swap space. Typically 0 indicates that swap space is not being used.
- Active: Active memory, the amount currently in use. This includes active anonymous memory (Active(anon)) and active file memory (Active(file)).
- Inactive: Inactive memory, previously used but not currently in use. This includes inactive anonymous memory (Inactive(anon)) and inactive file memory (Inactive(file)).

2.2.2. Obtain memory usage information

You can use the free command with the -m parameter to display memory usage in megabytes.

```
root@myd-yt113-i:~# free -m
              total        used        free      shared  buff/cache   available
Mem:            224          40          99           0          84         175
Swap:             0           0           0
```

- ➢ Total:Total memory
- ➢ used: Used memory
- ➢ free: Free memory

2.2.3. Memory stress testing

Given the size and number of memory tests, you can stress test the system's existing memory. You can use the system tool memtester test, such as specifying the memory size of 60MB, the number of tests for 10, the test command is "*memtester 60M 10*".

The following is an example of a single test using 60MB of memory space.

```
root@myd-yt113-i:~# memtester  60M 1
memtester version 4.3.0 (32-bit)
Copyright (C) 2001-2012 Charles Cazabon.
Licensed under the GNU General Public License version 2 (only).

pagesize is 4096
pagesizemask is 0xfffff000
want 60MB (62914560 bytes)
got  60MB (62914560 bytes), trying mlock ...locked.
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
```

2.3. eMMC

This section focuses on eMMC testing for development boards configured with eMMC memory. eMMC is a data storage device that includes a MultiMediaCard (MMC) interface. Its cost, small size, Flash technology independence and high data throughput make it ideal for embedded products.

2.3.1. EMMC test

1). Viewing eMMC Capacity

You can use the fdisk -l command to query eMMC partition information and capacity under Linux.

```
root@myd-yt113-i:~# fdisk -l
Found valid GPT with protective MBR; using GPT
Disk /dev/mmcblk0: 7471104 sectors, 3648M
Logical sector size: 512
Disk identifier (GUID): ab6f3888-569a-4926-9668-80941dcb40bc
Partition table holds up to 8 entries
First usable sector is 73728, last usable sector is 7471070

Number  Start (sector)    End (sector)  Size Name
     1           73728          108165 16.8M boot-resource
     2          108166          110213 1024K env
     3          110214          112261 1024K env-redund
     4          112262          147461 17.1M boot
     5          147462         2244613 1024M rootfs
     6         2244614         2246661 1024K dsp0
     7         2246662         2279429 16.0M private
     8         2279430         7471070 2534M UDISK
```

2). Viewing eMMC Partition Information

You can use the df command to query eMMC partition information,usage,mount points, and other details under Linux. Here's an example output:

```
root@myd-yt113-i:~# df -h
Filesystem                 Size      Used  Available  Use%  Mounted on
/dev/root               991.9M    392.5M     583.4M   40%  /
devtmpfs                102.7M         0     102.7M    0%  /dev
tmpfs                   112.4M         0     112.4M    0%  /dev/shm
tmpfs                   112.4M    268.0K     112.1M    0%  /tmp
tmpfs                   112.4M    172.0K     112.2M    0%  /run
/dev/mmcblk0p8             2.4G      7.4M       2.2G    0%  /media
```

- /dev/root : Root filesystem mounted at the root directory.
- tmpfs : Virtual memory filesystems mounted at various directories.
- Devtmpfs : Used by the system to create devices (/dev).

3). eMMC Performance Testing

Performance testing primarily measures the read and write speeds of eMMC for files under the Linux system. The following will introduce the use of the timedd command for conducting read and write performance tests.

- **Write Testing**

```
root@myd-yt113-i:~# time dd if=/dev/zero of=write_file bs=100M count=5  c
onv=fsync
5+0 records in
5+0 records out
524288000 bytes (524 MB, 500 MiB) copied, 13.8718 s, 37.8 MB/s

real    0m13.925s
user    0m0.001s
sys     0m6.088s
```

- **Read Testing**

In embedded systems, it is often necessary to test the read and write performance of system files. Ignore the impact of cache when reading files. At this point, you can specify the parameter iflag=direct,nonblock.

```
root@myd-yt113-i:~# time dd if=write_file of=/dev/null bs=100M count=5 ifla
g=direct,nonblock
```

```
5+0 records in
5+0 records out
524288000 bytes (524 MB, 500 MiB) copied, 11.33 s, 46.3 MB/s

real    0m11.394s
user    0m0.000s
sys     0m0.430s
```

2.4. PMIC

This chapter demonstrates the Suspend function of Linux power management, which puts the development board to sleep and wakes it up through external events. The Linux kernel generally provides three types of Suspend: Freeze, Standby, and STR (Suspend to RAM), which can be triggered by writing "*Freeze*" and "*Mem*" respectively to the "*/sys/power/state*" file in user space. MYD-YT113X supports two methods: freeze and mem.

1). View the supported modes of the current development board

```
root@myd-yt113-i:~# cat /sys/power/state
freeze mem
```

2). Methods for writing in userspace

```
root@myd-yt113-i:~# echo "freeze" > /sys/power/state
root@myd-yt113-i:~# echo "mem" > /sys/power/state
```

- **Mem sleep mode**

After entering the hibernate command, the board hibernates, the debugging serial port can no longer be input, at this time, the system and device status is saved to memory (in the self-refreshing mode, has retained its content), all devices enter the low-power mode.

```
root@myd-yt113-i:~# echo "mem" > /sys/power/state
```

- **Freeze sleep mode**

After inputting the hibernate command, the development board hibernates and the debugging serial port can no longer be input, at this time, the user space is frozen, all I/O devices enter the low-power state, and the processor enters the idle state.

```
root@myd-yt113-i:~# echo "freeze" > /sys/power/state
```

At this point, press the user button S1 to wake up the system and debug the serial port to re-enter.

3. Basic peripheral interface

3.1. GPIO

The test of GPIO is realized through the file system sysfs interface, the following contents take PD21 as an example to illustrate the process of using GPIO.

Calculate the value of the pin corresponding to gpio as (n-1) * 32+x (assuming A is 1, B is 2, and so on, D corresponds to 4) (x in PD21 represents 21), as follows:

Design the expansion interface of the baseboard PD21

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

```
echo 117 > /sys/class/gpio/export

echo out > /sys/class/gpio117/direction

echo 1 > xxx/value (high)

echo 0 > xxx/value (low)
```

Parameters:

- function: multiplexing
- data: level data (1 for high level; 0 for low level)
- plevel: drive capability
- pull: pull-up/down condition

1). Check the current GPIO usage

You can check the current GPIO usage with the following command.

```
root@myd-yt113-i:~# mount -t debugfs debugfs /sys/kernel/debug
root@myd-yt113-i:~# cat /sys/kernel/debug/gpio
gpiochip0: GPIOs 0-223, parent: platform/pio, pio:
 gpio-117 (                    |sysfs               ) in  lo
 gpio-129 (                    |user key            ) in  hi IRQ ACTIVE LOW
 gpio-130 (                    |led-blue            ) out lo ACTIVE LOW
 gpio-131 (                    |connect            ) in   hi IRQ
 gpio-139 (                    |phy-rst            ) out hi
 gpio-140 (                    |otg_id             ) in   hi IRQ
 gpio-166 (                    |cd                 ) in   hi ACTIVE LOW

gpiochip1: GPIOs 224-239, parent: i2c/3-0020, can sleep:
 gpio-230 (                    |led-green          ) out lo ACTIVE LOW
```

2). Export GPIO

```
root@myd-yt113-i:~# echo 117 > /sys/class/gpio/export
```

After successful export,a directory named gpio117 will be created under "*/sys/class/gpio/*".

3). Set/Check GPIO Direction

- **Set as input:**

```
root@myd-yt113-i:~# echo "in" > /sys/class/gpio/gpio117/direction
```

- **Set as output:**

```
root@myd-yt113-i:~# echo "out" > /sys/class/gpio/gpio117/direction
```

- **Check GPIO direction:**

```
root@myd-yt113-i:~# cat /sys/class/gpio/gpio117/direction
out
```

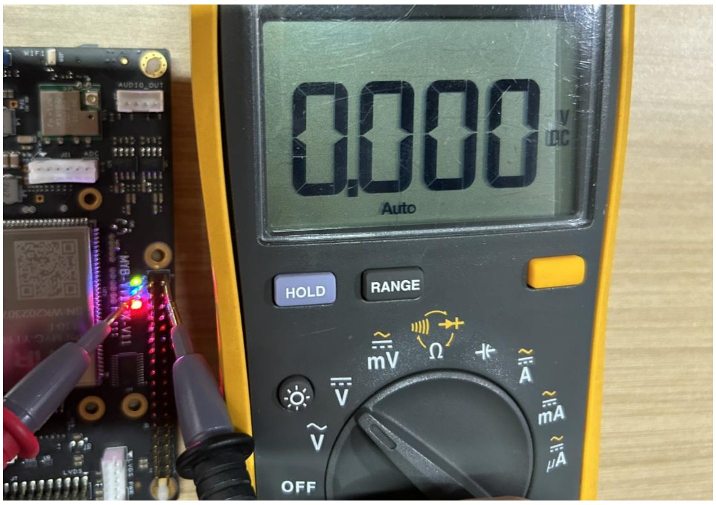
If it returns "*in*",it indicates the GPIO is set as input;if "*out*,"it indicates output.

4). Set/Check GPIO Value

- **Set output low**

Shown in Figure 3-1:

```
root@myd-yt113-i:~# echo "0" > /sys/class/gpio/gpio117/value
```

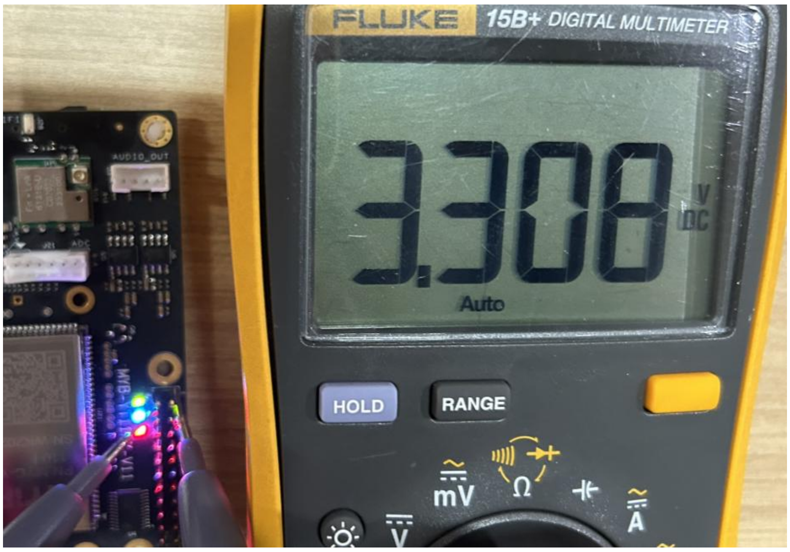


Figur 3-1. Set low output

- **Set high output**

As shown in Figure 3-2

```
root@myd-yt113-i:~# echo "1" > /sys/class/gpio/gpio117/value
```



Figur 3-2. Set high output

- **Check the value of gpio**

```
root@myd-yt113-i:~# cat /sys/class/gpio/gpio117/value
1
```

You can see that PD21 outputs a high level. You can use a multimeter to measure the PD21 pin of J2 extended IO, and you can see that the voltage is around 3.3V.

3.2. LED

Linux system provides an independent subsystem for easy user-space operations of LED devices, accessible through file interfaces in the "*/sys/class/leds*" directory. The following commands demonstrate how to test LEDs using sysfs read and write operations. These commands are universal and apply to general LED manipulation methods.

1). The directory for operating the LED is/sys/class/les

```
root@myd-yt113-i:/sys/class/leds# ls
led-blue    led-green
```

You can change the brightness of the heartbeat light by writing different values to "*/sys/class/leds/led-blue/brightness*".

2). Take heartbeat light led-blue as an example to test LEDs

```
root@myd-yt113-i:/sys/class/leds# echo none > /sys/class/leds/led-blue/trigger
```

Turn off the heartbeat light, and then you can perform separate LED off and on operations.

- **Turning off the LED**

```
root@myd-yt113-i:/sys/class/leds# echo 0 > /sys/class/leds/led-blue/brightness
```

- **Turning on the LED**

```
root@myd-yt113-i:/sys/class/leds# echo 1 > /sys/class/leds/led-blue/brightness
```

- **Enabling LED trigger mode:**

After writing 'heartbeat' to the trigger file, the 'run' LED will resume blinking at a 1Hz cycle.

```
root@myd-yt113-i:/sys/class/leds# echo heartbeat > /sys/class/leds/led-blue/tri
gger
```

3.3. Key

Linux's "*/dev/input/eventx*" devices can be used conveniently to debug input devices such as mice, keyboards, and touchpads. This section primarily focuses on testing keys. Use the "*hexdump*" command and "*dmesg*" command to check if the keys are responsive. The MYD-YT113X board has two keys.

- S1 : User Key
- S2 : reset Key

- **Device tree configuration information**

Open the accompanying device tree file "*device/config/chips/t113_i/config/myrir-image-yt113i-xxx/board.dts*", and you can see the node where the S1 User Kay is pressed: (xxx represents different configurations)

```
        gpio-keys {
                compatible = "gpio-keys";
                status = "okay";
                vol-down-key {
                        gpios = <&pio PE 1 GPIO_ACTIVE_LOW>;
                        linux,code = <114>;
                        label = "user key";
                        debounce-interval = <10>;
                        wakeup-source = <0x1>;
                };
          };
```

3.3.2. key test

- **View corresponding input device event information**

```
root@myd-yt113-i:~# cat /proc/bus/input/devices
I: Bus=0019 Vendor=0001 Product=0001 Version=0100
N: Name="sunxi-keyboard"
P: Phys=sunxikbd/input0
S: Sysfs=/devices/virtual/input/input0
U: Uniq=
H: Handlers=kbd event0
B: PROP=0
B: EV=3
B: KEY=1000 800 c0000 0 0 10000000

I: Bus=0019 Vendor=0001 Product=0001 Version=0100
N: Name="sunxi-tpadc"
P: Phys=
S: Sysfs=/devices/virtual/input/input1
U: Uniq=
H: Handlers=kbd event1
B: PROP=0
B: EV=100003
B: KEY=200000 0 0 0 0 0 0 0 8c0000 0 0 0

I: Bus=0019 Vendor=0001 Product=0001 Version=0100
N: Name="sunxi-gpadc0"
P: Phys=sunxigpadc0/input0
S: Sysfs=/devices/virtual/input/input2
U: Uniq=
H: Handlers=kbd event2
B: PROP=0
B: EV=100003
B: KEY=200000 0 0 0 0 0 0 0 8c0000 0 0 10000000

I: Bus=0019 Vendor=0001 Product=0001 Version=0100
N: Name="sunxi-ir"
P: Phys=sunxi-ir/input0
S: Sysfs=/devices/platform/soc@3000000/7040000.s_cir/rc/rc0/s_cir_rx
U: Uniq=
H: Handlers=kbd event3
B: PROP=20
```

```
B: EV=100017
B: KEY=2
B: REL=3
B: MSC=10

I: Bus=0019 Vendor=0001 Product=0001 Version=0100
N: Name="gpio-keys"
P: Phys=gpio-keys/input0
S: Sysfs=/devices/platform/gpio-keys/input/input5
U: Uniq=
H: Handlers=kbd event4
B: PROP=0
B: EV=3
B: KEY=40000 0 0 0
```

From the above, it can be seen that the corresponding device event for gpio keys is event4 (the user can determine the number of events based on the actual information by finding the gpio key name).

- **evtest test key information**

Execute the following command, operate the S1 key, and the serial terminal will print out the following information:

```
root@myd-yt113-i:~# evtest
No device specified, trying to scan all of /dev/input/event*
Available devices:
/dev/input/event0:      sunxi-keyboard
/dev/input/event1:      sunxi-tpadc
/dev/input/event2:      sunxi-gpadc0
/dev/input/event3:      sunxi-ir
/dev/input/event4:      gpio-keys
Select the device event number [0-4]: 4
Input driver version is 1.0.1
Input device ID: bus 0x19 vendor 0x1 product 0x1 version 0x100
Input device name: "gpio-keys"
```

```
Supported events:
  Event type 0 (EV_SYN)
  Event type 1 (EV_KEY)
    Event code 114 (KEY_VOLUMEDOWN)
Properties:
Testing ... (interrupt to exit)
Event: time 2583.205639, type 1 (EV_KEY), code 114 (KEY_VOLUMEDOWN), val
ue 1
Event: time 2583.205639, -------------- SYN_REPORT ------------
Event: time 2583.375614, type 1 (EV_KEY), code 114 (KEY_VOLUMEDOWN), val
ue 0
Event: time 2583.375614, -------------- SYN_REPORT ------------
Event: time 2583.865630, type 1 (EV_KEY), code 114 (KEY_VOLUMEDOWN), val
ue 1
Event: time 2583.865630, -------------- SYN_REPORT ------------
Event: time 2584.025630, type 1 (EV_KEY), code 114 (KEY_VOLUMEDOWN), val
ue 0
Event: time 2584.025630, -------------- SYN_REPORT ------------
Event: time 2584.495628, type 1 (EV_KEY), code 114 (KEY_VOLUMEDOWN), val
ue 1
Event: time 2584.495628, -------------- SYN_REPORT ------------
Event: time 2584.605620, type 1 (EV_KEY), code 114 (KEY_VOLUMEDOWN), val
ue 0
Event: time 2584.605620, -------------- SYN_REPORT -----------
```

Every time S1 is pressed, the current terminal will print the current event code value, indicating that the button press is normal.

3.3.3. Test with Rest key

The MYD-YT113X can be rebooted using the Reset button.

```
[29]HELLO! BOOT0 is starting!
[32]BOOT0 commit : 09b36272
[35]set pll start
```

```
[41]periph0 has been enabled
[44]set pll end
[45][pmu]: bus read error
[48]board init ok
[49]enable_jtag
[51]DRAM only have internal ZQ!!
[54]get_pmu_exist() = -1
[56]DRAM BOOT DRIVE INFO: V0.32
[59]DRAM CLK = 792 MHz
[62]DRAM Type = 3 (2:DDR2,3:DDR3)
[65]DRAMC ZQ value: 0x7b7bfb
......
```

3.4. USB

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

3.4.1. View the print information when the usb is inserted

- **View USB device information**

Connect the USB flash drive to the USB Host interface (J11) of the development board, and the prompt message is as follows:

```
root@myd-yt113-i:~# create /dev/sda
create /dev/sda1
root@myd-yt113-i:~# dmesg | grep sda
[   38.878204] sd 0:0:0:0: [sda] 60825600 512-byte logical blocks: (31.1 GB/29.
0 GiB)
[   38.879414] sd 0:0:0:0: [sda] Write Protect is off
[   38.879428] sd 0:0:0:0: [sda] Mode Sense: 03 00 00 00
[   38.880154] sd 0:0:0:0: [sda] No Caching mode page found
[   38.880166] sd 0:0:0:0: [sda] Assuming drive cache: write through
[   38.933146]  sda: sda1
[   38.937068] sd 0:0:0:0: [sda] Attached SCSI removable disk
[   39.040285] FAT-fs (sda1): Volume was not properly unmounted. Some dat
a may be corrupt. Please run fsck.
```

From the above information, it can be concluded that the device to be mounted is sda1, and it will be automatically mounted to the"*/run/media/sda1*" directory.

3.4.2. USB flash drive mount read and write

- **Write file**

A test.txt file needs to be created on the USB flash drive in advance.

```
root@myd-yt113-i:/run/media/sda1# time dd if=/dev/zero of=write_file bs=10
0M count=5 conv=fsync
5+0 records in
```

```
5+0 records out
524288000 bytes (524 MB, 500 MiB) copied, 30.9096 s, 17.0 MB/s

real	0m30.969s
user	0m0.001s
sys	0m5.876s
```

- **Read file**

```
root@myd-yt113-i:/run/media/sda1# time dd if=write_file of=/dev/null bs=100
M count=5 iflag=direct,nonblock
5+0 records in
5+0 records out
524288000 bytes (524 MB, 500 MiB) copied, 14.425 s, 36.3 MB/s

real	0m14.491s
user	0m0.001s
sys	0m0.555s
```

3.4.3. Unmount the USB flash drive

- **Uninstall operation**

When uninstalling a USB drive, exit the mounting directory.

```
root@myd-yt113-i:~# umount /run/media/sda1
```

3.5. USB-OTG

This section tests the host function of the typeC (J10) port of the development board by connecting a USB flash disk to the typeC-to-USB OTG cable, and the device mode of the typeC port by connecting a PC to the typeC-to-USB male cable to simulate the "*/dev/mmcblk0p8*" partition.

3.5.1. Test host function

- **Host mode**

```
root@myd-yt113-i:~# create /dev/sda1
create /dev/sda
root@myd-yt113-i:~# df -h
Filesystem           Size  Used Avail Use% Mounted on
/dev/root            992M  421M  556M  44% /
devtmpfs             103M     0  103M   0% /dev
tmpfs                113M     0  113M   0% /dev/shm
tmpfs                113M   56K  113M   1% /tmp
tmpfs                113M  188K  113M   1% /run
/dev/by-name/UDISK   2.5G  4.0K  2.5G   1% /mnt/UDISK
/dev/sda1             29G  117M   29G   1% /run/media/sda1
```

- **Write test**

```
root@myd-yt113-i:/run/media/sda1# time dd if=/dev/zero of=write_file bs=10
0M count=5  conv=fsync
5+0 records in
5+0 records out
524288000 bytes (524 MB, 500 MiB) copied, 31.4933 s, 16.6 MB/s

real    0m31.578s
user    0m0.001s
sys     0m5.856s
```

- **Read test**

```
root@myd-yt113-i:/run/media/sda1# time dd if=write_file of=/dev/null bs=100
M count=5 iflag=direct,nonblock
5+0 records in
5+0 records out
524288000 bytes (524 MB, 500 MiB) copied, 14.3796 s, 36.5 MB/s

real      0m14.450s
user      0m0.001s
sys       0m0.539s
```

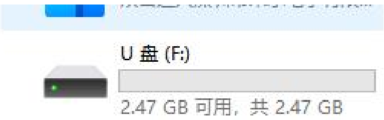
3.5.2. Test the device function

First, use a USB-TypeC cable to connect the development board to the host PC.

Then configure the development board as follows:

```
echo usb_device > /sys/devices/platform/soc@3000000/soc@3000000:usbc0@0
/otg_role
root@myd-yt113-i:~# echo none > /sys/kernel/config/usb_gadget/g1/UDC
root@myd-yt113-i:~# modprobe g_mass_storage stall=0 file=/dev/mmcblk0p8
removable=1  iSerialNumber="1234"
```

You can find the simulated mmcblk0p8 partition on the PC side as shown in Figure 3-3.



Figur 3-3. Simulated USB Flash Drive

3.6. Micro SD card

Micro SD Card, formerly known as Trans-flash Card(TF card), Micro SD card is a very small flash memory card. Compared with the standard SD card, the Micro SD card is more compact in appearance and is the smallest SD card type. Although the external size and interface shape of the Micro SD card are different from the original SD card, the interface specification remains the same to ensure compatibility. If Micro SD is inserted into a specific adapter card, it can be used as a standard SD card. SD card has become the most widely used memory card in consumer digital devices. It is a multi-functional memory card with large capacity, high performance and security. Micro SD card generally has 9 pins on the back, contains 4 data lines, and supports 1bit/4bit data transmission width.MYD-YT113X supports 3 channels of 1-bit or 4-bit SDMMC interfaces, and SDMMC0 is used to connect Micro SD on the development board.

3.6.1. Check the SD card capacity

The fdisk -l command can be used to query the SD card partition information and capacity.

```
root@myd-yt113-i:~# fdisk -l
Found valid GPT with protective MBR; using GPT

Disk /dev/mmcblk0: 7471104 sectors, 3648M
Logical sector size: 512
Disk identifier (GUID): ab6f3888-569a-4926-9668-80941dcb40bc
Partition table holds up to 8 entries
First usable sector is 73728, last usable sector is 7471070

Number  Start (sector)    End (sector)  Size Name
     1           73728          108165 16.8M boot-resource
     2          108166          110213 1024K env
     3          110214          112261 1024K env-redund
     4          112262          147461 17.1M boot
     5          147462          2244613 1024M rootfs
     6         2244614          2246661 1024K dsp0
     7         2246662          2279429 16.0M private
     8         2279430          7471070 2534M UDISK
Disk /dev/mmcblk1: 29 GB, 31266439168 bytes, 61067264 sectors
3786 cylinders, 256 heads, 63 sectors/track
Units: sectors of 1 * 512 = 512 bytes

Device         Boot StartCHS     EndCHS         StartLBA      EndLBA      Sectors
  Size Id Type
/dev/mmcblk1p1     0,0,0        1023,255,63       110214    61065214    609550
01 29.0G  c Win95 FAT32 (LBA)
```

3.6.2. Check the SD card partition information

Through the df-h command, you can query the SD card partition information, usage, mount directory and other information.

```
EMMC board type:
root@myd-yt113-i:~# df -h
Filesystem                  Size      Used Available Use% Mounted on
/dev/root                 991.9M    392.5M    583.4M  40% /
devtmpfs                  102.7M         0    102.7M   0% /dev
tmpfs                     112.4M         0    112.4M   0% /dev/shm
tmpfs                     112.4M    224.0K    112.2M   0% /tmp
tmpfs                     112.4M    180.0K    112.2M   0% /run
/dev/mmcblk0p8              2.4G      7.4M      2.2G   0% /media
/dev/mmcblk1p1             29.1G      5.3M     29.1G   0% /run/media/mmcb
lk1p1
```

- tmpfs: In-memory virtual filesystem, mounted in different directories
- devtmpfs: Used to create dev for the system
- /dev/mmcblk1p1: Directory of the first partition of the SD card

3.6.3. Performance test of SD card

Performance testing mainly assesses the read and write speeds of the SD card in a Linux system using the dd command. Here, we will mount the partition of the SD card to be tested. As an example, we will use the partition "*/dev/mmcblk1p1*" and mount it to the directory "*/run/media/mmcblk1p1*".

- **Write file test**

```
root@myd-yt113-i:/run/media/mmcblk1p1# time dd if=/dev/zero of=write_file
bs=100M count=5 conv=fsync
5+0 records in
5+0 records out
524288000 bytes (524 MB, 500 MiB) copied, 35.2864 s, 14.9 MB/s

real    0m35.338s
user    0m0.001s
sys     0m5.555s
```

- **Read feil test**

```
root@myd-yt113-i:/run/media/mmcblk1p1# time dd if=write_file of=/dev/null
bs=100M count=5 iflag=direct,nonblock
5+0 records in
5+0 records out
524288000 bytes (524 MB, 500 MiB) copied, 21.897 s, 23.9 MB/s

real    0m21.959s
user    0m0.001s
sys     0m0.470s
```

3.7. ADC

GPADC is a 12bit sampling precision analog-to-digital converter module, the analog input range is specified by the platform (up to 1.8V for MYD-YT113X platform), there is only one GPADC and four TPADC channels. the ADC is tested through the file system sysfs interface.

1). Read GPADC test value

First we select the gpadc0 channel and use the following command to read the default value directly.

```
root@myd-yt113-i:~# cat /sys/class/gpadc/data
1599
```

The data obtained is 1599, which is converted to a voltage value of: 1599/1000=1.599v.

```
root@myd-yt113-i:~# cat /sys/class/gpadc/data
3
```

Check the data again by connecting the gpadc0 channel to GND and get a voltage value of 000.3v.

```
root@myd-yt113-i:~# cat /sys/class/gpadc/data
1797
```

Check the data again by connecting the gpadc0 channel to 1.8 and get a voltage value of 1.797v

2). Read TPADC test values

There are a total of four tpadcs, and here we choose tpadc0 as an example

```
root@myd-yt113-i:~# echo 0 > /sys/devices/virtual/input/input1/channel_tpadc
```

Firstly, enable the tpadc0 channel.

```
root@myd-yt113-i:~# cat /sys/devices/virtual/input/input1/tpadc
1435
```

Then read the default value, and the obtained voltage is 1435/1000=1.435v.

```
root@myd-yt113-i:~# cat /sys/devices/virtual/input/input1/tpadc
0
```

Check the data again by connecting the tpadc0channel to GND and get a voltage value of 0v.

```
root@myd-yt113-i:~# cat /sys/devices/virtual/input/input1/tpadc
1797
```

Check the data again by connecting the tpadc0 channel to 1.8V and get a voltage value of 1.797v.

If other TPADCs need to be used, enable the channel first and then measure. The other three tpadc enable commands are as follows, and the measurement method is consistent with tpadc0.

```
root@myd-yt113-i:~# echo 1 > /sys/devices/virtual/input/input1/channel_tpadc
root@myd-yt113-i:~# echo 2 > /sys/devices/virtual/input/input1/channel_tpadc
root@myd-yt113-i:~# echo 3 > /sys/devices/virtual/input/input1/channel_tpadc
```

3.8. Display

This module consists of a display engine (DE) and various types of controllers (tcon). Input layers (layers) in the DE for display-related processing, through one or more interfaces output to the display device to display, in order to achieve a number of applications rendered layers synthesized in the display presented to the user to view the role of the DE has two independent units (can be referred to as de0, de1), which can be respectively accepted by the user input layer synthesis, the output to the different displays to achieve dual-display. DE has 2 independent units (can be referred to as de0, de1), which can accept user input layers for compositing and output to different displays to realize dual display. each independent unit of DE has 1-4 channels (typically, de0 has 4 on, de1 has 2), each channel can simultaneously process and accept 4 layers of the same format. sunxi platform has a video channel and the UI channel. 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:

- supports lcd output
- supports multi-layer overlay blending processing
- supports multiple display effect processing (alpha, colorkey, image enhancement, brightness/contrast/saturation/chroma adjustment)
- supports intelligent backlight adjustment

3.8.1. Device tree configuration information

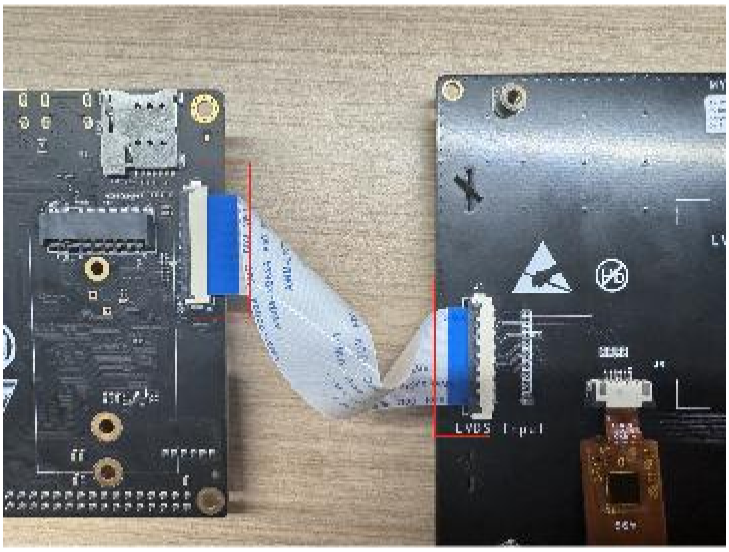
Open the accompanying device tree files "*device/config/chips/t113_i/config/myrir-image-yt113i-xxx/board.dts*" and "*uboot-board.dts*", and you will see two display schemes. (xxx represents different board configurations)

```
#include "sun8iw20p1.dtsi"
#include "myir-t113-lvds.dtsi"                7-inch LVDS screen display
//#include "myir-t113-lvds-dual.dtsi"         19 inch dual LVDS screen display
```

3.8.2. Display scheme combination

- **Default display**

MYD-YT113X defaults to 7-inch LVDS display, using 40pin double-ended reverse FPC cable, accessed with the blue side of the cable facing up at both ends of the development board and screen.



Figur 3-4. Connection between J6 and LVDS screen

- **19 inch dual LVDS screen display**

Before compilation, select the 19 inch dual LVDS display scheme in the device tree board.dts and uboot-board.dts, and annotate the 7-inch LVDS display scheme. (Cannot select two display schemes simultaneously)

```
//#include  "myir-t113-lvds.dtsi"
#include  "myir-t113-lvds-dual.dtsi"
```

3.9. Touch Panel

There are capacitive touch and resistive touch, MYD-YT113X development board hardware currently does not support resistive touch, but supports capacitive touch, MYiR Tech provides LVDS display accessories, see Table 1-2. you can buy your own accessories according to the actual needs. Capacitive screen is more sensitive in use, and rarely have 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. We are in use if the phenomenon of clicking on the software selection is not hit, there is generally only one situation: the screen has a problem. The following is a simple test through the evtest command to test the capacitive screen touch function of the test.

3.9.1. evtest command testing

Terminal execution “*evtest*” to enter the test interface. Select the test peripheral for the touch screen, the default here is to enter the interrupt “*4*”, the test interface select “*4*” press enter to start the test: (Users please select according to their own actual evtest display of the touch device).

```
root@myd-yt113-i:~# evtest
No device specified, trying to scan all of /dev/input/event*
Available devices:
/dev/input/event0:      sunxi-keyboard
/dev/input/event1:      sunxi-tpadc
/dev/input/event2:      sunxi-gpadc0
/dev/input/event3:      sunxi-ir
/dev/input/event4:      generic ft5x06 (79)
/dev/input/event5:      gpio-keys
Select the device event number [0-5]: 4
Input driver version is 1.0.1
Input device ID: bus 0x18 vendor 0x0 product 0x0 version 0x0
Input device name: "generic ft5x06 (79)"
```

```
Supported events:
  Event type 0 (EV_SYN)
  Event type 1 (EV_KEY)
    Event code 330 (BTN_TOUCH)
  Event type 3 (EV_ABS)
    Event code 0 (ABS_X)
      Value      0
      Min        0
      Max     1023
    Event code 1 (ABS_Y)
      Value      0
      Min        0
      Max      599
    Event code 47 (ABS_MT_SLOT)
      Value      0
      Min        0
      Max        4
    Event code 53 (ABS_MT_POSITION_X)
      Value      0
      Min        0
      Max     1023
    Event code 54 (ABS_MT_POSITION_Y)
      Value      0
      Min        0
      Max      599
    Event code 57 (ABS_MT_TRACKING_ID)
      Value      0
      Min        0
      Max    65535
Properties:
  Property type 1 (INPUT_PROP_DIRECT)
Testing ... (interrupt to exit)
```

```
Event: time 433.946640, type 3 (EV_ABS), code 57 (ABS_MT_TRACKING_ID), val
ue 0
Event: time 433.946640, type 3 (EV_ABS), code 53 (ABS_MT_POSITION_X), valu
e 344
Event: time 433.946640, type 3 (EV_ABS), code 54 (ABS_MT_POSITION_Y), valu
e 703
Event: time 433.946640, type 1 (EV_KEY), code 330 (BTN_TOUCH), value 1
Event: time 433.946640, type 3 (EV_ABS), code 0 (ABS_X), value 344
Event: time 433.946640, type 3 (EV_ABS), code 1 (ABS_Y), value 703
Event: time 433.946640, -------------- SYN_REPORT ------------
Event: time 433.965434, type 3 (EV_ABS), code 53 (ABS_MT_POSITION_X), valu
e 342
Event: time 433.965434, type 3 (EV_ABS), code 54 (ABS_MT_POSITION_Y), valu
e 701
Event: time 433.965434, type 3 (EV_ABS), code 0 (ABS_X), value 342
Event: time 433.965434, type 3 (EV_ABS), code 1 (ABS_Y), value 701
Event: time 433.965434, -------------- SYN_REPORT ------------
Event: time 433.985361, type 3 (EV_ABS), code 53 (ABS_MT_POSITION_X), valu
e 346
Event: time 433.985361, type 3 (EV_ABS), code 0 (ABS_X), value 346
Event: time 433.985361, -------------- SYN_REPORT ------------
Event: time 434.006843, type 3 (EV_ABS), code 54 (ABS_MT_POSITION_Y), valu
e 700
Event: time 434.006843, type 3 (EV_ABS), code 1 (ABS_Y), value 700
Event: time 434.006843, -------------- SYN_REPORT ------------
Event: time 434.024685, type 3 (EV_ABS), code 57 (ABS_MT_TRACKING_ID), val
ue -1
Event: time 434.024685, type 1 (EV_KEY), code 330 (BTN_TOUCH), value 0
Event: time 434.024685, -------------- SYN_REPORT ------------
Event: time 434.585329, type 3 (EV_ABS), code 57 (ABS_MT_TRACKING_ID), val
ue 1
```

It can be seen from the above that it mainly displays coordinate values and key values, and the specific information is as follows:

- EV_SYN：Synchronization events
- EV_KEY：Key event; for example, BTN_TOUCH indicates a touch key
- EV_ABS：bsolute coordinates, such as those reported by the touch screen
- BTN_TOUCH：Touch the key
- ABS_MT_TRACKING_ID:indicates the beginning of collecting information, and the following ABS_MT_TRACKING_ID indicates the end of collecting information

Single touch information is carried in ABS and sent in a certain order, such as:

- ABS_X: Is the absolute coordinate X relative to the screen
- ABS_Y: Is the absolute coordinate Y relative to the screen

Multi-touch messages, on the other hand, are carried in ABS_MT and sent in a certain order, such as:

- ABS_MT_POSITION_X: Indicates the x-coordinate position of the center point of the screen contact surface.
- ABS_MT_POSITION_Y: This indicates the y-coordinate position of the center point of the screen contact.

3.10. Audio

Using the system comes with the music and script for playback, you need to access the playback device in the J3 interface, in the directory "*/etc/myir_test*" under the playback script myir_audio_play as follows:

```
root@myd-yt113-i:~# cat /etc/myir_test/myir_audio_play
#!/bin/bash


default_audio="/usr/share/myir/Music/myir_audio.wav"


play_audio() {
    local audio_file="$1"
    tinyplay "$audio_file"
}

while true; do

    read -p "Do you want to play the default audio file? (y/n/quit): " choice


    case $choice in
        [Yy]*)
            play_audio "$default_audio"
            ;;
        [Nn]*)
            read -p "Enter the path of the audio file you want to play: " custom_audio

            if [ -f "$custom_audio" ]; then
                play_audio "$custom_audio"
            else
                echo "Error: The specified audio file does not exist."
            fi
            ;;
        [Qq]*|quit)
            echo "Exiting the script."
            exit 0
            ;;
        *)
            echo "Invalid choice. Please enter 'y', 'n', or 'quit'."
            ;;
    esac
done
```

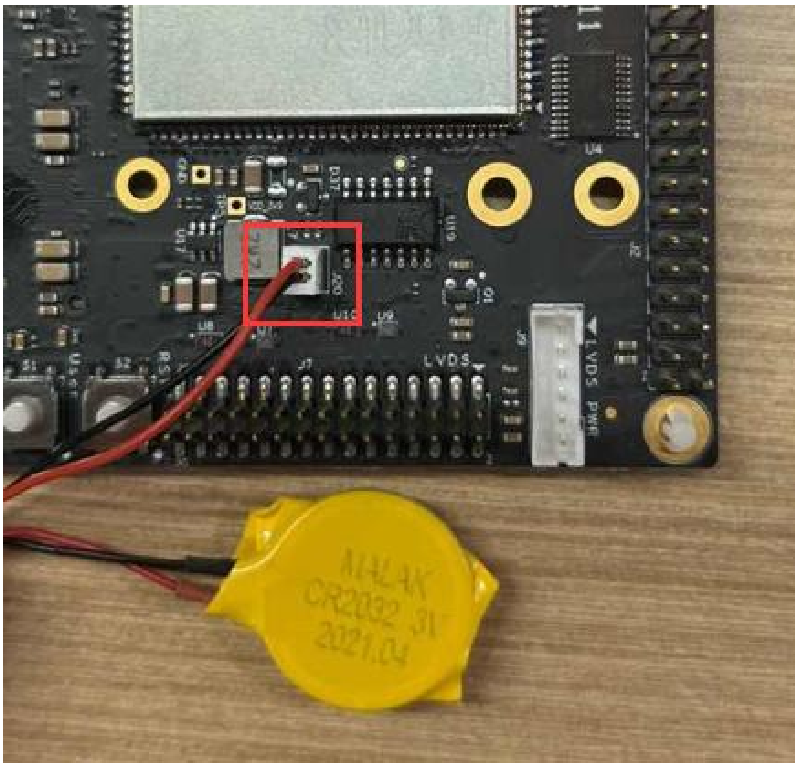
- **Play music in wav format**

```
root@myd-yt113-i:~# /etc/myir_test/myir_audio_play
Do you want to play the default audio file? (y/n/quit): n
Enter the path of the audio file you want to play: /usr/share/myir/Music/myir_audio.wav
playing '/usr/share/myir/Music/myir_audio.wav': 2 ch, 48000 hz, 16 bit
```

3.11. RTC

RTC (Real-time clock) itself is a clock, used to record the real time, when the software system is shut down to retain the system time and continue to timekeeping, the system reopened in the time synchronized into the software system.MYD-YT113X has an external RTC (RX8025), if the actual product on the RTC power requirements are not very high, the RTC test is usually used in conjunction with the hwclock and date commands commonly used in Linux systems. If the actual product does not require high power consumption of RTC, the RTC test is usually carried out with the hwclock and date commands commonly used in the Linux system. The following test will write the system time into the RTC, read the RTC time and set it as the system time, and carry out the test of keeping the power-down time.

Test the external RTC needs to be in the development board J20 interface, access to the RTC battery, as shown in Figure 3-5:



Figur 3-5. RTC battery diagram

- **Checking the system RTC device:**

```
root@myd-yt113-i:~#  ls /dev/rtc* -al
```

```
lrwxrwxrwx 1 root root        4 Jan  1 08:00 /dev/rtc -> rtc0
crw-rw---- 1 root root 252, 0 Jan  1 08:00 /dev/rtc0
```

- **Setting the system time:**

Set the system time to "*2024-07-22 10:51:10*".

```
root@myd-yt113-i:~# date -s "2024-07-22 10:51:10"
Mon Jul 22 10:51:10 UTC 2024
```

- **Writing system time to RTC:**

After setting the system time with the date command, the next step would be to write this time to the RTC device.

```
root@myd-yt113-i:~# hwclock -w -f /dev/rtc0
```

- **Reading external RTC time**

```
root@myd-yt113-i:~# hwclock -r -f /dev/rtc0
Mon Jul 22 10:51:30 2024  0.000000 seconds
```

- **Retaining RTC Time during Power-off**

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

```
root@myd-yt113-i:~# hwclock -r -f /dev/rtc0
Mon Jul 22 10:56:19 2024  0.000000 seconds
```

After rebooting, if the RTC time is approximately 5 minutes later than the time set previously, it indicates that the RTC is functioning correctly. For detailed accuracy testing of the RTC, you can extend the power-off duration, such as to 24 hours, and compare the RTC time with standard time.

- **Synchronizing System Time with RTC Time:**

```
root@myd-yt113-i:~# hwclock -s -f /dev/rtc0
root@myd-yt113-i:~# date
Mon Jul 22 11:12:30 UTC 2024
```

If you add the command hwclock -s to the startup script, it will ensure that the system time and RTC time are synchronized each time the system boots up.

3.12. Watchdog

The Linux kernel contains the Watchdog subsystem, and the hardware design process can generally utilize the watchdog timer inside the chip or use an external watchdog chip to realize the Watchdog function for monitoring the system operation. When the system is unable to feed the dog when an abnormal situation occurs the system will be able to carry out an automatic reset.MYD-YT113X development board has an external watchdog, the internal watchdog has been turned off, this chapter will explain the test method of the external watchdog under linux.

3.12.1. Testing the Watchdog

- **Watchdog Application Testing:**

```
root@myd-yt113-i# watchdog
Usage: wdt_driver_test <timeout> <sleep> <test>
timeout: value in seconds to cause wdt timeout/reset
sleep: value in seconds to service the wdt
test: 0 - Service wdt with ioctl(), 1 - with write()
```

The command starts the watchdog timer with a specified timeout period. The <timeout> parameter indicates the watchdog's timeout period, during which it waits for a system response. The <sleep> parameter determines how often the watchdog is "*fed*" (notified that the system is operating normally), in seconds. If the watchdog does not receive a response from the system within this timeout period, it forces a system restart. Initially, set the timeout to 10 seconds and the feed time to 5 seconds. You can observe that the system continues to run normally.

```
root@myd-yt113-i# watchdog 10 5 0
Starting wdt_driver (timeout: 10, sleep: 5, test: ioctl)
Trying to set timeout value=10 seconds
The actual timeout was set to 10 seconds
Now reading back -- The timeout is 10 seconds
```

Resetting the timeout to 4 seconds and the feeding interval to 5 seconds means that if the watchdog is not fed after 4 seconds, the system will restart directly due to timeout.

```
root@myd-yt113-i# watchdog 4 5 0
Starting wdt_driver (timeout: 4, sleep: 5, 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
[28]HELLO! BOOT0 is starting!
[31]BOOT0 commit : 069ed30b88
[34]set pll start
[40]periph0 has been enabled
[43]set pll end
[45][pmu]: bus read error
[47]board init ok
[49]enable_jtag
[50]get_pmu_exist() = -1
[53]DRAM BOOT DRIVE INFO: V0.34
[56]DRAM CLK = 792 MHz
[58]DRAM Type = 3 (2:DDR2,3:DDR3)
[61]DRAMC ZQ value: 0x7b7bfb
[64]DRAM ODT value: 0x42.
[67]ddr_efuse_type: 0x0
[70]DRAM SIZE = 256 MB
[76]DRAM simple test OK.
[79]rtc standby flag is 0x0, super standby flag is 0x0
[84]dram size =256
[87]card no is 2
[88]sdcard 2 line count 4
[90][mmc]: mmc driver ver 2021-05-21 14:47
[100][mmc]: Wrong media type 0x0, but host sdc2, try mmc first
[105][mmc]: ***Try MMC card 2***
```

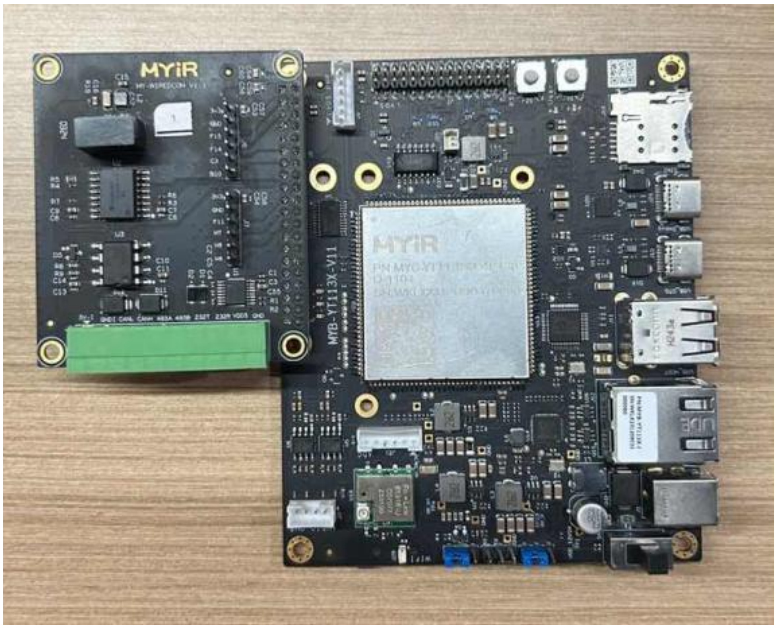
```
[130][mmc]: RMCA OK!
[132][mmc]: mmc 2 bias 0
[137][mmc]: MMC 5.1
[138][mmc]: HSSDR52/SDR25 4 bit
......
```

4. Expand peripheral interface

MYD-YT113X development board provides a wealth of peripheral interfaces, in addition to the basic peripheral interfaces, can also be connected to a variety of external expansion modules. Make the user's development more flexible and convenient. The following describes the test steps for several optional modules introduced by Mill. Users are free to purchase according to their needs. For details of the optional modules, refer to Table 1-2 Optional Modules List.

4.1. MY-WiredCom module

MY-WiredCom module is a form of Raspberry Pi peripheral interface introduced by MYiR, including RS232/RS485/CAN/SPI/I2C peripheral interfaces. The module needs to be purchased by the user according to the need, please refer to Table 1-2 Optional Module List for module details. Before testing, users need to connect the module to the J2 interface of the development board as shown in Figure 4-1:



Figur 4-1. J2 connected to Raspberry Pi

1). RS485 test

This section will use the Linux API to configure the RS485 send/receive function of the development board. The Linux serial device file is usually named /dev/ttySn(n=0,1,2,3.......). n means the device number of the serial port in Linux system, and "*ttyS*" is the name of the serial port device defined by the kernel. This section is to test the J2 interface on the MY-WiredCom expansion board as an example, the numbered device node of the J2 interface is ttyS4. Its test configuration is shown in the following table:

Table 4-1. RS485 Interface Configuration

Test points	MYD-YT113X	Windows 10
hardware interface	RS485	USB-RS485 module
Device Node	ttyS4	com12（According to the actual situation）
Test Software	uart_test_rs485_read uart_test_rs485_write	sscom

(uart_test_rs485_read, uart_test_rs485_write are stored in the "*/etc/myir_test directory*").

Connect the 485A and 485B of RS485 to the 485A and 485B of the testing equipment, as shown in Figure 4-2:

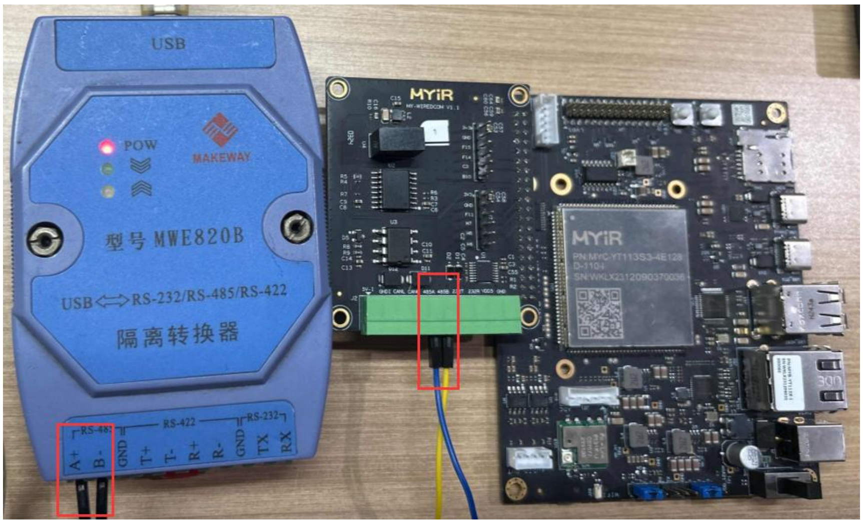
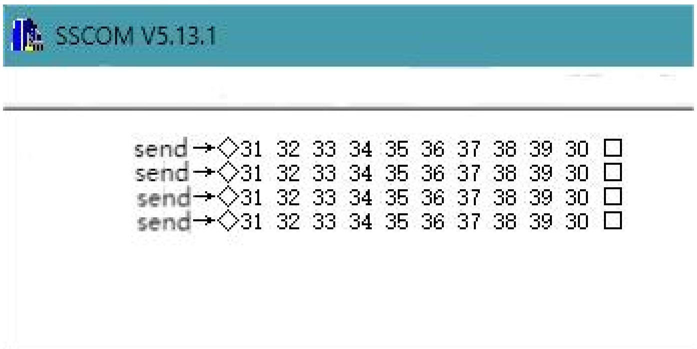


Figure 4-2. RS485 connection

- **Test development board RS485 data reception**

Firstly, set up the SSCOM serial port tool under Windows to send strings at regular intervals, as shown in Figure 4-3:



```
send→◇31 32 33 34 35 36 37 38 39 30 □
send→◇31 32 33 34 35 36 37 38 39 30 □
send→◇31 32 33 34 35 36 37 38 39 30 □
send→◇31 32 33 34 35 36 37 38 39 30 □
```

Figure 4-3. SSCOM sends data

The development board is ready to receive data. Execute the following command on the development board to receive the data.

After the command execution is completed, the interrupt will enter a blocking state, waiting for the data sent from the computer serial port to be received. When the data is received from the computer serial port, the received string will be printed and displayed on the terminal.

```
root@myd-yt113-i:~# /etc/myir_test/uart_test_rs485_read
Please enter the device name (ttyS4):
ttyS4
Set the baud rate. You can select from [230400, 115200, 57600, 38400, 19200,
 9600, 4800, 2400, 1200, 300]:
115200
read:
RECV[10]: 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x39 0x30
RECV[10]: 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x39 0x30
RECV[10]: 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x39 0x30
RECV[10]: 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x39 0x30
```

- **Test development board RS485 data transmission**

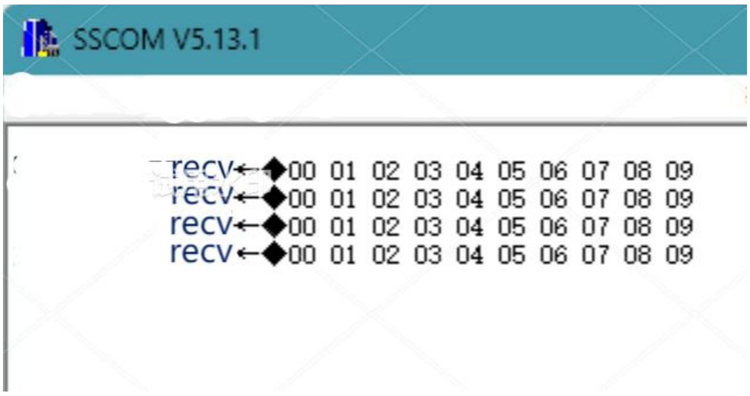
Executing the following command on the development board will send data to UART4, and the serial port tool under Windows will receive the data.

Then execute the send command:

```
root@myd-yt113-i:~# /etc/myir_test/uart_test_rs485_write
Please enter the device name (ttyS4):
ttyS4
```

```
Set the baud rate. You can select from [230400, 115200, 57600, 38400, 19200,
 9600, 4800, 2400, 1200, 300]:
115200
write:
SEND[10]: 0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09
SEND[10]: 0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09
SEND[10]: 0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09
SEND[10]: 0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09
```

The data sent and received on the development board corresponds to the data sent and received on SSCOM under Windows, that is, the RS485 data received by the development board is normal, as shown in Figure 4-4:



```
recv←◆00 01 02 03 04 05 06 07 08 09
recv←◆00 01 02 03 04 05 06 07 08 09
recv←◆00 01 02 03 04 05 06 07 08 09
recv←◆00 01 02 03 04 05 06 07 08 09
```

Figure 4-4. SSCOM receives data

2). CAN test

This section uses the commonly used cansend and candump commands in Linux systems to conduct communication testing for SocketCAN. The two development boards used for testing here are docked for testing.

Here, the CANH and CANL pins of the MY WiredCom module J2 socket are connected to the same type of board CANH and CANL, as shown in Figure 4-5:

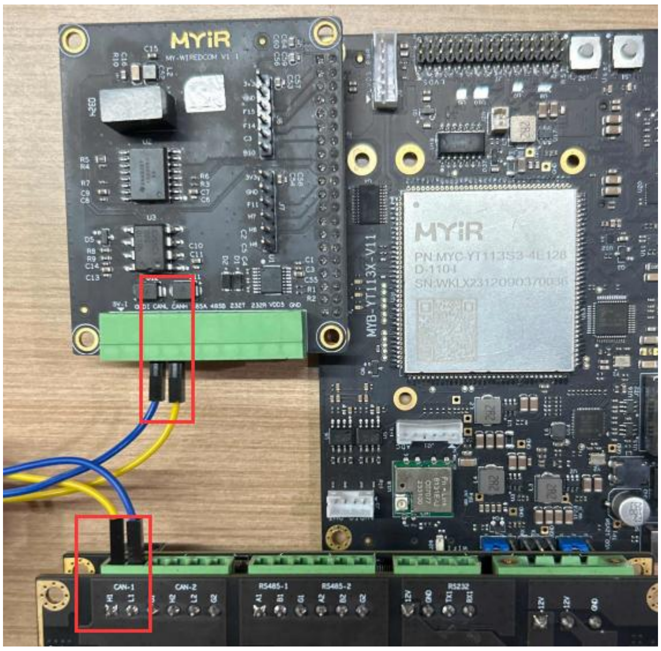


Figure 4-5. Can connect

- **Initialize CAN network interface**

Here the CAN node name is awlink0, set the CAN baud rate, use awlink0 need to set the baud rate first, and open the CAN network interface. Refer to the following commands to set the data baud rate of the two development boards to 20KHz respectively, and turn on the CAN function: (the highest tested 1M)

```
root@myd-yt113-i:~# ifconfig awlink0 down
root@myd-yt113-i:~# ip link set awlink0 up type can bitrate 500000
```

At this point, the CAN function is activated.

- **send data**

Set one of the boards to send and use cansend to send data:

```
root@myd-yt113-i:~# cansend awlink0 123#EDEDEDED
root@myd-yt113-i:~# cansend awlink0 123#EDEDEDED
root@myd-yt113-i:~# cansend awlink0 123#EDEDEDED
```

- **receive data**

Set another board as the receiver and use candump to view the CAN receiving data:

```
root@myd-yt113-i:~# candump awlink0
  awlink0  123   [4]  ED ED ED ED
  awlink0  123   [4]  ED  ED  ED  ED
  awlink0  123   [4]  ED  ED  ED  ED
```

- **Statistics of awlink0**

After CAN data is sent and received, the CAN device details and send/receive statistics are displayed, where the value of “*clock*” represents the CAN clock, the value of “drop” represents packet loss, the value of “overrun” represents overflow, and the value of “error” represents bus error. The value of “*clock*” represents the clock of the can, the value of “drop” represents the packet loss, the value of “overrun” represents the overflow, and the value of “*error*” represents the bus error.

```
root@myd-yt113-i:~# ip -details -statistics link show awlink0
2: awlink0: <NOARP,ECHO> mtu 16 qdisc noop state DOWN mode DEFAULT group default qlen 10
    link/can  promiscuity 0 minmtu 0 maxmtu 0
    can state STOPPED (berr-counter tx 0 rx 0) restart-ms 0
          sunxi-awlink: tseg1 1..16 tseg2 2..8 sjw 1..4 brp 2..1024 brp-inc 1
          clock 24000000
          re-started bus-errors arbit-lost error-warn error-pass bus-off
          0          0          0          0          0          0          numtxqueues 1 numrxqueues 1 gso_max_size 65536 gso_max_segs 65535
    RX: bytes  packets  errors  dropped overrun mcast
    0          0        0       0       0       0
    TX: bytes  packets  errors  dropped carrier collsns
    0          0        0       0       0       0
```

5. Network and Communication Interface

This chapter is designed to test and evaluate the connection, configuration, and use of network devices such as Ethernet, Wi-Fi, and (optionally) mobile networks.

5.1. Network Devices

The MYD-YT113X development board contains a Gigabit Ethernet interface (eth0), a Wi-Fi module, and a 4G/5G module interface J15 (on the back of the board), and the configurations of these three network devices are described below.

5.1.1. Ethernet

There are many tools for Linux network configuration,common net-tools, iproute2, systemd-networkd,network manager and connman, etc.which can be customized according to the actual needs of the system to choose.The MYD-YT113X development board uses systemd-networkd to manage the network.

5.1.2. Configure Ethernet IP address

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

First view the network device information by passing the ifconfig command as follows:

```
root@myd-yt113-i:~# ifconfig
eth0      Link encap:Ethernet  HWaddr C2:77:C8:5F:E1:47
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:38
```

```
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 (0.0 B)  TX bytes:0 (0.0 B)

wlan0       Link encap:Ethernet  HWaddr 54:F2:9F:50:21:0D
            UP BROADCAST MULTICAST  MTU:1500  Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Here is the method to manually configure the IP address 192.168.0.100 for eth0. The command is as follows:

```
root@myd-yt113-i:~# ifconfig eth0 192.168.0.100 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 a broadcast address of 192.168.0.255 configured by default and activated with the up parameter as shown below:

```
root@myd-yt113-i:~# ifconfig eth0 192.168.0.100 netmask 255.255.255.0 up
root@myd-yt113-i:~# ifconfig eth0
eth0        Link encap:Ethernet  HWaddr C2:77:C8:5F:E1:47
            inet addr:192.168.0.100  Bcast:192.168.0.255  Mask:255.255.255.0
            UP BROADCAST MULTICAST  MTU:1500  Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
```

```
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:38
```

- **Manual configuration of the network using the ip command from the iproute2 toolkit**

The method of manually setting the IP address using the ifconfig command can also be replaced by using IP addr and IP link. For more information, please refer to https://wiki.linuxfoundation.org/networking/iproute2 Explanation in.

```
root@myd-yt113-i:~# ip addr flush dev eth0
root@myd-yt113-i:~# ip addr add 192.168.0.101/24 brd + dev eth0
root@myd-yt113-i:~# ip link set eth0 up
```

If you have already configured an IP address before, the IP address configured with ip addr add will become the second ary address, so use ip addr flush to clear the previous address before configuring and activating it. After completing the configuration, view the eth0 information through the ip addr show command as follows:

```
root@myd-yt113-i:~# ip addr show eth0
3: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast
state DOWN group default qlen 1000
    link/ether c2:77:c8:5f:e1:47 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.101/24 brd 192.168.0.255 scope global eth0
       valid_lft forever preferred_lft foreverr
```

5.1.3. Change Mac address

The method to manually modify the Mac address 00:0C:29:36:97:20 is as follows:

```
root@myd-yt113-i:~# ifconfig eth0 down
root@myd-yt113-i:~# ifconfig eth0 hw ether 00:0C:29:36:97:20
root@myd-yt113-i:~# ifconfig eth0 up
root@myd-yt113-i:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0C:29:36:97:20
          inet addr:192.168.0.101  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:38
```

5.1.4. Wi-Fi

This section introduces the configuration and use of Wi-Fi under Linux. usually Wi-Fi module can support two working modes, STA mode and AP mode, and some devices also support STA and AP modes at the same time. STA mode allows the device to connect to external Wi-Fi hotspots, and AP mode turns the device into a Wi-Fi hotspot for other devices to connect to.

5.1.4.1. STA mode connects to WiFi hotspot

1). Using shell scripts to connect to Wi Fi

```
root@myd-yt113-i:~#/etc/myir_test/wifi_on_sta -ssid MYIR_WH -passwd Myir@
2016
SSID:MYIR_WH PASSWD:Myir@2016 DRIVER:nl80211
killall: udhcpc: no process killed
killall: wpa_supplicant: no process killed
killall: hostapd: no process killed
killall: udhcpd: no process killed
find phy0 enable it
udhcpc: started, v1.33.2
udhcpc: sending discover
udhcpc: sending select for 192.168.30.212
udhcpc: lease of 192.168.30.212 obtained, lease time 60000
deleting routers
adding dns 192.168.30.1
```

2). Manually connect to WiFi

Below try to manually connect to a nearby Wi-Fi hotspot SSID, this is a Wi-Fi hotspot with WPA2 encryption, please configure it yourself.

Make sure the wlan0 network device is active.

```
root@myd-yt113-i:~# ifconfig wlan0 up
```

- **Scan nearby WiFi hotspots**

Scan nearby Wi Fi hotspots and obtain the following list of nearby Wi Fi hotspots:

```
root@myd-yt113-i:~# iw dev wlan0 scan | grep SSID
        SSID: MYIR_WH
        SSID: MYIR_mobile
        SSID: BSPTEST_24G
        SSID: 360_2.4g
        SSID: OBOX-5G-87D0
        SSID: BSPTEST_5G
        SSID: agingtest_5G
        SSID: MYIR_MES_5g
        SSID: MYIR_5G
```

- **Wpa_passphrase: Set WiFi name and password**

```
root@myd-yt113-i:~# wpa_passphrase ssid password >> /etc/wpa_supplicant.conf
root@myd-yt113-i:~# cat /etc/wpa_supplicant.conf
ctrl_interface=/var/run/wpa_supplicant
ap_scan=1

network={
  key_mgmt=NONE
}
network={
        ssid="MYIR_WH"
        #psk="Myir@2016"
        psk=8a24681161052b0e89883c5e9256f049fe4e39d28bac65e44df4bbcfcb0ae0b9
}
```

Generate a WPA PSK from the ASCII cipher of an SSID for encryption operations.

- **Turn off the wpa_supplicant process**

Before using wpa_supplicant to connect and configure WIFI, you need to turn off the wpa_supplicant process:

```
root@myd-yt113-i:~# killall wpa_supplicant
```

- **Initializing wpa_supplican**

wpa_supplicant is a tool for connecting and configuring WIFI, its main job is to interact with the driver and report data to the user layer via socket, while the user layer can also send commands to wpa_supplicant via socket to mobilize the driver to operate on the WiFi chip. It usually runs in the background, as shown below:

```
root@myd-yt113-i:~# wpa_supplicant -B -Dnl80211 -c /etc/wpa_supplicant.conf
 -i wlan0
Successfully initialized wpa_supplicant
```

- ➢ -B: running daemon in the background
- ➢ -D: name of the driver
- ➢ -c: path to configuration information
- ➢ -i: wifi interface to listen on

- **Get ip address**

```
root@myd-yt113-i:~# udhcpc -b -i wlan0 -R
udhcpc: started, v1.33.2
udhcpc: sending discover
udhcpc: sending select for 192.168.30.212
udhcpc: lease of 192.168.30.212 obtained, lease time 60000
deleting routers
adding dns 192.168.30.1
```

- **Ping Baidu check if you can connect normally**

```
root@myd-yt113-i:~# ping www.baidu.com -I wlan0
PING www.baidu.com (153.3.238.102): 56 data bytes
64 bytes from 153.3.238.102: seq=0 ttl=53 time=14.994 ms
```

```
64 bytes from 153.3.238.102: seq=1 ttl=53 time=15.698 ms
64 bytes from 153.3.238.102: seq=2 ttl=53 time=13.924 ms
64 bytes from 153.3.238.102: seq=3 ttl=53 time=14.128 ms
```

5.1.4.2. AP mode to turn on hotspot

Hostapd is a wireless access point program with encryption, a more convenient tool for building a wireless access point on Linux OS, supporting IEEE 802.11 protocol and IEEE 802.1X/WPA/WPA2/EAP/RADIUS encryption. The board acts as a WiFi hotspot, and you need to assign IP, route, and other network parameters to each terminal (e.g., cell phone) that accesses the hotspot. For example, create a wireless wifi hotspot with SSID myd_t113i_test_wifi and PASSWD 12345678. The following describes the script to open the hotspot and manual configuration to open the hotspot.

Note: MYD-YT113X development board does not support WIFI STA mode and AP mode at the same time.

1). Using shell scripts to activate hotspots

We will manually configure the Wi-Fi process organized into a script "*/etc/myir_test/wifi_on_ap*" to provide users with reference to the implementation of this script can be configured to AP mode, the implementation of the success of the user can be connected to the MYIR_TEST hotspot using other devices to test.

```
root@myd-yt113-i:~# /etc/myir_test#./wifi_on_ap
killall:udhcpc:no process killed
killall:wpa_supplicant:no process killed
killall:hostapd:no process killed
killall:udhcpd:no process killed
find phy0 enable it
Configuration file:/etc/hostapd.conf
Using interface wlane with hwaddr 54:f2:9f:50:1f:50 and ssid "myd_t113i_test_w
ifi'
wlan0:interface state UNINITIALIZED->ENABLED
wlane:AP-ENABLED
```

After connecting to the development board hotspot with a mobile device, perform a network ping packet.

```
root@myd-yt113-i:~# ping 192.168.1.5 -I wlan0
PING 192.168.0.5 (192.168.1.5): 56 data bytes
64 bytes from 192.168.1.5: seq=0 ttl=64 time=4.433 ms
64 bytes from 192.168.1.5: seq=1 ttl=64 time=5.338 ms
64 bytes from 192.168.1.5: seq=2 ttl=64 time=13.632 ms
64 bytes from 192.168.1.5: seq=3 ttl=64 time=27.319 ms
64 bytes from 192.168.1.5: seq=4 ttl=64 time=5.427 ms
^C
--- 192.168.1.5 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 4.433/11.229/27.319 ms
```

2). Turning on a hotspot manually

To customize the hotspot, you need to configure the following manually.

- **Configure the IP address of wlan0**

When using AP mode, you need to activate wlan0 and configure a static IP address, here configure a default IP address: 192.168.1.10

```
root@myd-yt113-i:~# ifconfig wlan0 192.168.1.10 up
```

- **Use wlan0 to run DHCP service.**

wlan0 works in AP mode. when other devices connect to this AP hotspot, it needs to dynamically assign IP addresses to other devices through wlan0, so you need to use wlan0 to run the DHCP service program udhcpd. udhcpd corresponds to the configuration file "*/etc/udhcpd.conf*".

```
# File: /etc/udhcpd.conf
# Sample udhcpd configuration file (/etc/udhcpd.conf)

# The start and end of the IP lease block

start           192.168.1.10      #default: 192.168.1.20
```

```
end                 192.168.1.254    #default: 192.168.1.254



# The interface that udhcpd will use

interface         wlan0             #default: eth0



# The maximim number of leases (includes addressesd reserved
# by OFFER's, DECLINE's, and ARP conficts

#max_leases       254               #default: 254



# If remaining is true (default), udhcpd will store the time
# remaining for each lease in the udhcpd leases file. This is
# for embedded systems that cannot keep time between reboots.
# If you set remaining to no, the absolute time that the lease
# expires at will be stored in the dhcpd.leases file.

#remaining        yes               #default: yes



# The time period at which udhcpd will write out a dhcpd.leases
# file. If this is 0, udhcpd will never automatically write a
# lease file. (specified in seconds)

#auto_time        7200              #default: 7200 (2 hours)



# The amount of time that an IP will be reserved (leased) for if a
# DHCP decline message is received (seconds).
```

```
#decline_time   3600            #default: 3600 (1 hour)


# The amount of time that an IP will be reserved (leased) for if an
# ARP conflct occurs. (seconds

#conflict_time  3600            #default: 3600 (1 hour)


# How long an offered address is reserved (leased) in seconds

#offer_time     60              #default: 60 (1 minute)

# If a lease to be given is below this value, the full lease time is
# instead used (seconds).

#min_lease      60              #defult: 60


# The location of the leases file

#lease_file     /var/lib/misc/udhcpd.leases     #defualt: /var/lib/misc/udhcpd.leases

# The location of the pid file
#pidfile        /var/run/udhcpd.pid     #default: /var/run/udhcpd.pid

# Everytime udhcpd writes a leases file, the below script will be called.
# Useful for writing the lease file to flash every few hours.

#notify_file                            #default: (no script)
```

```
#notify_file     dumpleases        # <--- usefull for debugging

# The following are bootp specific options, setable by udhcpd.

#siaddr            192.168.1.22                #default: 0.0.0.0

#sname              zorak                        #default: (none)

#boot_file        /var/nfs_root                #default: (none)

# The remainer of options are DHCP options and can be specifed with the
# keyword 'opt' or 'option'. If an option can take multiple items, such
# as the dns option, they can be listed on the same line, or multiple
# lines. The only option with a default is 'lease'.

#Examles
opt       dns       8.8.8.8
option    subnet    255.255.255.0
opt       router    192.168.1.1
opt       wins      192.168.1.10
option    dns       129.219.13.81    # appened to above DNS servers for a total
of 3
option    domain    local
option    lease     864000           # 10 days of seconds


# Currently supported options, for more info, see options.c
#subnet
#timezone
#router
#timesvr
```

```
#namesvr
#dns
#logsvr
#cookiesvr
#lprsvr
#bootsize
#domain
#swapsvr
#rootpath
#ipttl
#mtu
#broadcast
#wins
#lease
#ntpsrv
#tftp
#bootfile
```

When the AP mode configuration is complete and other devices connect to this hotspot, they will get the IP address in the above address pool through wlan0, with the address range of 192.168.1.10~192.168.1.254, subnet mask of 255.255.255.0, default gateway of 192.168.1.1, and DNS of 8.8.8.8.

- **Run the hostapd service using wlan0**

The most critical step in configuring AP mode is of course to start the hostapd service. Before starting the service, you need to configure the AP mode ssid, password, encryption algorithm, driver type, working mode and so on via "*/etc/hostapd.conf*". For the complete parameter configuration instructions, see http://w1.fi/cgit/hostap/plain/hostapd/hostapd.conf,The content is as follows:

```
#  File:  /etc/hostapd.conf
interface=wlan0
driver=nl80211
ssid=myd_t113i_test_wifi
channel=6
```

```
hw_mode=g
ignore_broadcast_ssid=0
auth_algs=1
wpa=2
wpa_passphrase=12345678
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

After the configuration file is ready, you need to shut down the wpa_supplicant process first. Then execute the following command to start the hostapd service, you can use the above configured hotspot normally.

```
root@myd-yt113-i:~# killall wpa_supplicant
root@myd-yt113-i:~# killall hostapd
root@myd-yt113-i:~# killall udhcpd
root@myd-yt113-i:~# killall udhcpd.conf
root@myd-yt113-i:~# ifconfig wlan0 192.168.1.10 up
root@myd-yt113-i:~# udhcpd /etc/udhcpd.conf
root@myd-yt113-i:~# hostapd -B /etc/hostapd.conf
Configuration file: /etc/hostapd.conf
Using interface wlan0 with hwaddr 54:f2:9f:50:23:f2 and ssid "myd_t113i_test_w
ifi"
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
```

5.1.5. 4G/5G

Linux devices can also be connected to external 4G or 5G modules to dial-up the Internet, MYD-YT113X development board uses the EM05-CE 4G module and RM500Q-CN 5G module.

Dialing methods are pppd, gobinet and qmi_wwan 3 ways, of which pppd is more common, gobinet is not used, does not support qmi_wwan, and can be used to do the 5G module connection, the following EM05-CE module as an example for illustration, before testing the module to be connected to the development board's J16 interface, in addition to the need to give the 5G module connected to the IPEX4 generation of IPEX4 interface, the 5G module. Before testing, the module should be connected to the J16 interface of the development board, and the 5G module should be connected to the antenna of IPEX4 generation.

MYD-YT113X has two SIM card slots, respectively J16 and J17. 4G module can only use SIM 1 (J16). 5G module can access two SIM cards at the same time, but can not be used at the same time SIM 1 and SIM 2 dialing, but only through the command to switch the SIM card, choose to use SIM 1 or SIM 2, we will introduce how to switch the use of SIM cards.

1). View VID and PID

```
root@myd-yt113-i:~# lsusb
Bus 001 Device 001: ID 1d6b:0002
Bus 001 Device 004: ID 2c7c:0800
Bus 001 Device 002: ID 1a40:0101
Bus 002 Device 001: ID 1d6b:0001
Bus 001 Device 003: ID 0bda:b733
```

- 2c7c: 0125: VID and PID information of EM05-CE.
- 2c7c: 0800: VID and PID information of RM500Q-CN.

2). Check the kernel identification module

If the kernel increases the VID and PID configuration for this module, then "*/dev/ttyUSB**" nodes will be generated:

```
root@myd-yt113-i:~# ls -l /dev/ttyUSB*
crw-rw----    1 root     dialout   188,   0 Jan  1 00:00 /dev/ttyUSB0
crw-rw----    1 root     dialout   188,   1 Jan  1 00:00 /dev/ttyUSB1
crw-rw----    1 root     dialout   188,   2 Jan  1 00:00 /dev/ttyUSB2
crw-rw----    1 root     dialout   188,   3 Jan  1 00:00 /dev/ttyUSB3
```

3). Use script testing

```
root@myd-yt113-i:~# /etc/myir_test/myir_dial
try 1...

pppd options in effect:
debug           # (from /etc/ppp/peers/quectel-ppp)
nodetach                # (from /etc/ppp/peers/quectel-ppp)
dump            # (from /etc/ppp/peers/quectel-ppp)
noauth          # (from /etc/ppp/peers/quectel-ppp)
user test               # (from /etc/ppp/peers/quectel-ppp)
password ??????         # (from /etc/ppp/peers/quectel-ppp)
remotename 3gppp                # (from /etc/ppp/peers/quectel-ppp)
/dev/ttyUSB3            # (from /etc/ppp/peers/quectel-ppp)
115200          # (from /etc/ppp/peers/quectel-ppp)
lock            # (from /etc/ppp/peers/quectel-ppp)
connect chat -s -v -f /etc/ppp/peers/quectel-chat-connect               # (fro
m /etc/ppp/peers/quectel-ppp)
......
rcvd [IPCP ConfAck id=0x5 <addr 10.67.53.10> <ms-dns1 111.48.8.188> <ms-
dns2 111.48.10.188>]
Could not determine remote IP address: defaulting to 10.64.64.64
not replacing existing default route via 192.168.40.1
local  IP address 10.67.53.10
remote IP address 10.64.64.64
primary   DNS address 111.48.8.188
```

```
secondary DNS address 111.48.10.188
ppp connected!
```

- **View ppp0 node**

```
root@myd-yt113-i:~# ifconfig ppp0
ppp0      Link encap:Point-to-Point Protocol
          inet addr:10.67.53.10  P-t-P:10.64.64.64  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:7 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:124 (124.0 B)  TX bytes:148 (148.0 B)
```

- **Ping external network test**

```
root@myd-yt113-i:~# ping www.baidu.com -I ppp0
PING www.baidu.com (36.155.132.76): 56 data bytes
64 bytes from 36.155.132.76: seq=0 ttl=50 time=474.417 ms
64 bytes from 36.155.132.76: seq=1 ttl=50 time=223.872 ms
64 bytes from 36.155.132.76: seq=2 ttl=50 time=313.756 ms
64 bytes from 36.155.132.76: seq=3 ttl=50 time=378.418 ms
64 bytes from 36.155.132.76: seq=4 ttl=50 time=268.207 ms
64 bytes from 36.155.132.76: seq=5 ttl=50 time=283.357 ms
64 bytes from 36.155.132.76: seq=6 ttl=50 time=353.639 ms
64 bytes from 36.155.132.76: seq=7 ttl=50 time=398.406 ms
^C
--- www.baidu.com ping statistics ---
8 packets transmitted, 8 packets received, 0% packet loss
round-trip min/avg/max = 223.872/336.759/474.417 ms
```

4). manual test

Using the AT command can be convenient to query the signal strength, whether the SIM card is inserted, whether the SIM card is currently searching for the operator, you can also use AT to make a phone call to test the current card function. Here for AT communication also need to know which device is the communication port, here you need to query the module file, EM05 and RM500Q using ttyUSB2 for AT communication. Here microcom example, you can also use minicom. such as interrupt the execution of microcom "*/dev/ttyUSB2*" command to enter the AT command mode, the keyboard press ctrl + x to exit the AT command mode.

- **Query signal quality**

```
root@myd-yt113-i:~# microcom /dev/ttyUSB2
at+csq
+CSQ: 27,99

OK
```

➢ 27,99: 27 is the signal quality, ranging from 0 to 31 (99 means no signal) Higher numbers mean stronger signal.

- **Check if it can be operated**

```
at+cpin?
+CPIN: READY

OK
```

➢ +CPIN:READY :READY stands for ready.

- **Check the carrier**

```
at+cops?
+COPS: 0,0,"CHN-UNICOM",7

OK
```

➢ CHN-UNICOM,7: CHN-UNICOM stands for Unicom, 7 stands for adopting 2G, 3G, 4G, or 5G according to the module manual.

If the above three steps can be normal, you can dial up the Internet, here also describes how to call and send text messages, further verification.

- **Check card slot**

Sometimes sim card is inserted in slot one (j16) but recognized in slot two (j17), then dialing will fail, the following commands are to check the current slot position and switch the slot.

Note: Only the 5G module needs to switch the card slot, the 4G module can only use the SIM1 card slot and does not need to switch the card slot.

Query Card Slot

```
root@myd-yt113-i:~# microcom /dev/ttyUSB2
AT+QUIMSLOT?
+QUIMSLOT: 1

OK
```

Switch card slot to 1

```
root@myd-yt113-i:~# microcom /dev/ttyUSB2

AT+QUIMSLOT=1
OK

+CPIN: NOT READY

+CPIN: READY

+QUSIM: 1

+QIND: SMS DONE

+QIND: PB DONE
```

- **PPP dial-up test**

Here we use the pppd dialing command that comes with the development board:

```
root@myd-yt113-i:~# pppd call quectel-ppp &
[1] 1731
```

```
root@myd-yt113-i:~# pppd options in effect:
debug            # (from /etc/ppp/peers/quectel-ppp)
nodetach                 # (from /etc/ppp/peers/quectel-ppp)
dump             # (from /etc/ppp/peers/quectel-ppp)
noauth           # (from /etc/ppp/peers/quectel-ppp)
user test                # (from /etc/ppp/peers/quectel-ppp)
password ??????          # (from /etc/ppp/peers/quectel-ppp)
remotename 3gppp                 # (from /etc/ppp/peers/quectel-ppp)
/dev/ttyUSB3             # (from /etc/ppp/peers/quectel-ppp)
115200           # (from /etc/ppp/peers/quectel-ppp)
lock             # (from /etc/ppp/peers/quectel-ppp)
connect chat -s -v -f /etc/ppp/peers/quectel-chat-connect                # (fro
m /etc/ppp/peers/quectel-ppp)
disconnect chat -s -v -f /etc/ppp/peers/quectel-chat-disconnect          # (fro
m /etc/ppp/peers/quectel-ppp)
nocrtscts                # (from /etc/ppp/peers/quectel-ppp)
modem            # (from /etc/ppp/peers/quectel-ppp)
hide-password            # (from /etc/ppp/peers/quectel-ppp)
novj             # (from /etc/ppp/peers/quectel-ppp)
novjccomp                # (from /etc/ppp/peers/quectel-ppp)
ipcp-accept-local                # (from /etc/ppp/peers/quectel-ppp)
ipcp-accept-remote               # (from /etc/ppp/peers/quectel-ppp)
ipparam 3gppp            # (from /etc/ppp/peers/quectel-ppp)
noipdefault              # (from /etc/ppp/peers/quectel-ppp)
ipcp-max-failure 30              # (from /etc/ppp/peers/quectel-ppp)
defaultroute             # (from /etc/ppp/peers/quectel-ppp)
usepeerdns               # (from /etc/ppp/peers/quectel-ppp)
noccp            # (from /etc/ppp/peers/quectel-ppp)
abort on (BUSY)
abort on (NO CARRIER)
abort on (NO DIALTONE)
abort on (ERROR)
abort on (NO ANSWER)
timeout set to 30 seconds
send (AT^M)
expect (OK)
AT^M^M
OK
 -- got it

send (ATE0^M)
......
sent [IPCP ConfReq id=0x2 <addr 10.108.21.1> <ms-dns1 111.48.8.188> <ms-dns2 111.48.10.188>]
rcvd [IPCP ConfReq id=0x1]
sent [IPCP ConfAck id=0x1]
rcvd [IPCP ConfAck id=0x2 <addr 10.108.21.1> <ms-dns1 111.48.8.188> <ms-dns2 111.48.10.188>]
Could not determine remote IP address: defaulting to 10.64.64.64
not replacing existing default route through wwan0
local  IP address 10.108.21.1
remote IP address 10.64.64.64
primary   DNS address 111.48.8.188
secondary DNS address 111.48.10.188
```

It can be seen that the connection is normal and the IP address can be obtained.

```
root@myd-yt113-i:~# ifconfig
eth0      Link encap:Ethernet  HWaddr C2:87:FD:A8:00:AB  
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000 
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:38 
```

```
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 (0.0 B)  TX bytes:0 (0.0 B)

ppp0         Link encap:Point-to-Point Protocol
            inet addr:10.108.21.1  P-t-P:10.64.64.64  Mask:255.255.255.255
            UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:
1
            RX packets:4 errors:0 dropped:0 overruns:0 frame:0
            TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:3
            RX bytes:52 (52.0 B)  TX bytes:58 (58.0 B)

wlan0        Link encap:Ethernet  HWaddr 54:F2:9F:50:23:F2
            UP BROADCAST MULTICAST  MTU:1500  Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

wwan0         Link encap:Ethernet  HWaddr 52:78:21:37:ED:BA
            inet addr:169.254.146.44  Bcast:169.254.255.255  Mask:255.255.0.0
            inet6 addr: fe80::da9f:5535:7ea:d535/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:18 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
```

```
          RX bytes:0 (0.0 B)  TX bytes:2496 (2.4 KiB)
```

- ppp0: ppp0 is the dial-up network card device, and the IP address is obtained normally.

- **Ping external network test**

```
root@myd-yt113-i:~# ping www.baidu.com -I ppp0
PING www.baidu.com (112.80.248.76): 56 data bytes
64 bytes from 112.80.248.76: seq=0 ttl=55 time=56.226 ms
64 bytes from 112.80.248.76: seq=1 ttl=55 time=54.201 ms
64 bytes from 112.80.248.76: seq=2 ttl=55 time=61.772 ms
64 bytes from 112.80.248.76: seq=3 ttl=55 time=30.343 ms
64 bytes from 112.80.248.76: seq=4 ttl=55 time=41.040 ms
^C
--- www.baidu.com ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
```

5.2. Network applications

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

5.2.1. PING

PING is primarily used to test network connectivity, but it can also be used to test network latency and packet loss rates. Once you've configured your Ethernet connection as described in 4.1.1, you can simply test your network connection using PING.

1). Wiring and information output

Connect the development board to the switch or router through the network cable, the console will display the connection information output by the kernel, as follows: (through the test CAT6 type network cable speed stability, using other network cable speed may not be stable).

```
root@myd-yt113-i:~# dmesg
[ 1360.246251] sunxi-gmac 4500000.gmac0 eth0: Link is Up - 1Gbps/Full - flo
w control off
```

2). Test external sites

```
root@myd-yt113-i:~# ping www.baidu.com
PING www.baidu.com (153.3.238.110): 56 data bytes
64 bytes from 153.3.238.110: seq=0 ttl=54 time=12.720 ms
64 bytes from 153.3.238.110: seq=1 ttl=54 time=12.572 ms
64 bytes from 153.3.238.110: seq=2 ttl=54 time=13.014 ms
64 bytes from 153.3.238.110: seq=3 ttl=54 time=12.517 ms
64 bytes from 153.3.238.110: seq=4 ttl=54 time=12.896 ms
64 bytes from 153.3.238.110: seq=5 ttl=54 time=12.286 ms
^C
--- www.baidu.com ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max = 12.286/12.667/13.014 ms
```

Note: ping the public network needs to ensure that the DNS is working properly.

The above results show that the IP address of www.baidu.com after domain name resolution is 153.3.238.110, and icm p_seq represents the number of icmp packets. If the numbers are consecutive, it indicates that there is no packet loss; Time represents the delay time of the response, and of course, the shorter the time, the better. In addition to testing Ethernet, the ping command can also be used to test Wi Fi.

5.2.2. SSH

SSH stands for Secure Shell and was developed by the Network Working Group of the IETF. SSH is a security protocol built on the application layer and is designed to provide secure remote login sessions and other network services. It is a reliable protocol for this purpose. On Linux platforms, OpenSSH or Dropbear is commonly used to implement SSH server and client functionalities. Below, we will test the use of SSH client and server over an Ethernet connection.The current factory default includes the client and service programs provided by openssh 7.6p1 (http://www.openssh.com/). First configure the development board Ethernet interface to SSH server connection, the configured Ethernet card address is as follows:

```
root@myd-yt113-i:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 3A:55:34:46:7E:12
          inet addr:192.168.40.96  Bcast:192.168.40.255  Mask:255.255.255.0
          inet6 addr: fe80::871c:39be:9002:d90/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:184023 errors:3953 dropped:0 overruns:0 frame:0
          TX packets:279 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:91522179 (87.2 MiB)  TX bytes:63527 (62.0 KiB)
          Interrupt:387
```

The IP address of the SSH server is 192.168.40.96.After testing the connection between the device and the SSH server with the ping command, the following tests can be performed.

1). SSH client testing

The development board connects to the SSH server as an SSH client, and the ssh command is used to log in to the SSH server on the device. The command and results are as follows:

```
root@myd-yt113-i:~# ssh szy@192.168.40.20
The authenticity of host '192.168.40.20 (192.168.40.20)' can't be established.
ECDSA key fingerprint is SHA256:QeyhIflWyl9VP8QnBzKaHdZbg8G+lJNEuQYth8
NZyww.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.40.20' (ECDSA) to the list of known hos
ts.
szy@192.168.40.20's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-107-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2025.
*** System restart required ***
Last login: Fri Aug  2 17:27:29 2024 from 192.168.40.41
szy@bsp:~$
```

where szy is the user name on the server.

After successfully logging in, you will be automatically entered into the console console on the SSH server, and you will be able to control the remote server from the client with the privileges of the szy user. If you need to exit, just execute the "*exit*" command on the console..

2). SSH server test

The development board acts as an SSH server, and other devices connect to the development board remotely.Since the SSH server is also enabled on the development board by default, we can also use the SSH command to log in to the current development board on any other device that has an ssh client. The command and results are as follows:

```
szy@bsp:/media/szy$ ssh root@192.168.40.96
The authenticity of host '192.168.40.96 (192.168.40.96)' can't be established.
ECDSA key fingerprint is SHA256:T3IXeI/6JRNoNXzHhxD0nctgzjBj//z5omfNL2I7e
gc.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.40.96' (ECDSA) to the list of known hos
ts.
COLUMNS=186;LINES=28;export COLUMNS LINES;
root@myd-yt113-i:~#
```

In the above example, we have logged in to the board from a remote location with the szy account and accessed the console, which allows us to control the board with szy privileges, and if we need to log out, we can simply execute the “*exit*” command from the console. If you need to exit, just execute the “*exit*” command directly from the console. openSSH is the primary connection tool for remote login using the SSH protocol. It encrypts all traffic to eliminate eavesdropping, connection hijacking and other attacks. In addition, OpenSSH offers a large set of secure tunneling features, multiple authentication methods, and sophisticated and flexible configuration options. Users can modify the configuration files ssh_config and sshd_config located in the /etc/ssh/ directory of the computer's host, according to their needs.

5.2.3. SCP

SCP is the abbreviation of Secure Copy, it is the linux system based on SSH protocol secure remote file copy command, in the system debugging phase is very practical.we have already discussed examples of remote login using the SSH protocol with SSH clients and servers. Here, we will now introduce an example of file remote copying using the SCP command:

1). Copy files from local to development board

```
root@myd-yt113-i:~# scp szy@192.168.40.20:/media/szy/test.txt ./
szy@192.168.40.20's password:
```

This file can be seen in the current directory of the development board as follows:

```
root@myd-yt113-i:~# ls test.txt
test.txt
```

2). Copy files from the development board to the local system

```
szy@bsp:/media/szy$ scp root@192.168.40.96:/root/test.txt ./
test.txt                                         100%    6      0.1KB/s    00:00
```

During the copying process, you need to follow the prompts and after successful verification, the file will be copied from the development board to the directory of the specified account on the server.

```
szy@bsp:/media/szy$ ls test.txt
test.txt
```

It is also possible to copy directories by adding the "*-r*" flag; see the scp command for help.

5.2.4. TFTP

TFTP uses client and server software to connect and transfer files between two devices, but the difference is that TFTP uses UDP protocol, does not have the login function, it is very simple, especially suitable for transferring and backup firmware, configuration files and other information in the device and server. For example, the common u-boot supports TFTP protocol, which can load the server-side Linux system through the network and realize the function of network boot.The default image file contains the tftp client program provided by busybox, which has the following command syntax:

```
root@myd-yt113-i:~# tftp --help
```

```
BusyBox v1.33.2 (2024-07-19 14:42:46 CST) multi-call binary.

Usage: tftp [OPTIONS] HOST [PORT]

Transfer a file from/to tftp server

        -l FILE Local FILE
        -r FILE Remote FILE
        -g      Get file
        -p      Put file
        -b SIZE Transfer blocks in bytes
```

The detailed parameters are as follows:

- -g: Get file
- -p: Upload file
- -l: Local file
- -r: Remote file
- HOST: Remote host IP address

TFTP server can choose Linux platform TFTP-hpa, also can choose Windows platform tftpd32/64(http://tftpd32.jounin.net/tftpd32_download.html). The following ubuntu platform as an example to explain the tftp server configuration.

- **Install TFTP server**

```
$: sudo apt-get install tftp-hpa tftpd-hpa
```

- **Configure TFTP service**

Create the TFTP server working directory and open the TFTP service configuration file as follows:

```
$: mkdir -p <WORKDIR>/tftpboot
$: chmod -R 777 <WORKDIR>/tftpboot
$: sudo vi /etc/default/tftpd-hpa
```

Modify or add the following fields:

```
TFTP_DIRECTORY="<WORKDIR>/tftpboot"
```

```
TFTP_OPTIONS="-l -c -s"
```

- **Restart the TFTP service**

```
$: sudo service tftpd-hpa restart
```

After configuring the tftp server, place a test file test.txt in the <WORKDIR>/tftpboot/ directory configured above, and you can use the tftp client to download and upload files on the target machine.

```
root@myd-yt113-i:~# tftp -g -r zImage -l zImage 192.168.40.207
```

The above command will download the zImage in the tftp server/tftpboot directory to the current directory of the development board.

```
root@myd-yt113-i:~# tftp -p -l config -r config_01 192.168.40.207
```

The above command will upload the config file from the current directory on the development board to the <WORKDIR>/tftpboot directory previously configured by the tftp server and rename it to config_01.

5.2.5. DHCP

DHCP (Dynamic Host Configuration Protocol) is a network protocol for LANs. It refers to a range of IP addresses that are controlled by a server, and clients can automatically obtain the IP address and subnet mask assigned by the server when they log in to the server. dhcp also contains both server-side and client-side roles. dhcp server-side mode was tested to assign an IP address to a connected WiFi device when configuring WiFi's AP mode in 5.1.4.2. Here is another introduction to manually obtaining an IP address using the udhcpc command, which is convenient for users to use when debugging the network.

- **Configuring an IP Address Using the udhcpc Command**

```
root@myd-yt113-i:~# udhcpc -i eth0
udhcpc: started, v1.33.2
udhcpc: sending discover
udhcpc: sending select for 192.168.40.220
udhcpc: lease of 192.168.40.220 obtained, lease time 60000
deleting routers
route: SIOCADDRT: Network is unreachable
adding dns 114.114.114.114
adding dns 218.106.127.114
```

Regardless of the method used, the IP address, gateway, subnet mask, DNS, and other information can ultimately be configured for eth0 as follows:

```
root@myd-yt113-i:~#  ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 3A:55:34:46:7E:12
          inet addr:192.168.40.220  Bcast:192.168.40.255  Mask:255.255.255.0
          inet6 addr: fe80::871c:39be:9002:d90/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:250662 errors:5259 dropped:0 overruns:0 frame:0
          TX packets:479 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:122676959 (116.9 MiB)  TX bytes:88754 (86.6 KiB)
          Interrupt:38
```

5.2.6. Iptables

iptables is an administrative tool for IPv4 packet filtering and NAT. It is used to set up, 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 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 then processes various packets such as accept, reject, and drop based on the methods 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 specific command usage see:https://linux.die.net/man/8/iptables.

1). Configure the development board Iptables

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

```
root@myd-yt113-i:~# iptables -A INPUT -p icmp --icmp-type 8 -j DROP
root@myd-yt113-i:~# 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

Pinging the development board on the development host and specifying deadline as 10 results in the following:

```
C:\Users\40938>ping 192.168.40.220

Pinging 192.168.40.220 with 32 bytes of data:
Request timeout.
Request timeout.
Request timeout.
Request timeout.

Ping statistics for 192.168.40.220.
    Packets: sent = 4, received = 0, lost = 4 (100% lost).,
```

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

Delete the corresponding firewall rule.

```
root@myd-yt113-i:~# iptables -F
root@myd-yt113-i:~# iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
```

- **Test pinging the board again**

```
C:\Users\40938>ping 192.168.40.220

Pinging 192.168.40.220 with 32 bytes of data.
Reply from 192.168.40.220: bytes=32 time=74ms TTL=64
Reply from 192.168.40.220: bytes=32 time=26ms TTL=64
Reply from 192.168.40.220: bytes=32 time=13ms TTL=64
Reply from 192.168.40.220: bytes=32 time=10ms TTL=64

Ping stats from 192.168.40.220.
    Packets: Sent = 4, Received = 4, Lost = 0 (0% lost).
Estimated round-trip travel time in milliseconds.
    Minimum = 10ms, Maximum = 74ms, Average = 30ms
```

After clearing the iptables rules, ping the board again from the development host and it pings through. The above example is just a simple demonstration, in fact, iptables with a variety of rules can achieve a very powerful function, we will not go into detail here.

5.2.7. iPerf3

iPerf3 is a tool to actively measure the maximum achievable bandwidth over IP networks. It supports tuning various parameters such as test time, buffer size, and protocol (TCP, UDP, SCTP for IPV4 and IPV6). iPerf3 can be divided into server-side mode or client-side mode according to its role. We can use it to test and view network bandwidth, TCP window value, retransmission probability, etc., in TCP mode, and also test packet loss rate, delay and jitter under specified UDP bandwidth.

We open Windows PowerShell on the development host, the host computer with a Gigabit NIC acts as the server of iperf3, and the development board under test acts as the client to test the performance of TCP and UDP of the board's NIC respectively. First, install iperf3 on the host computer as follows:

Connect the server and the development board directly via CAT cable and configure their respective I addresses. For example, we set the server IP to 192.168.40.20 and the development board IP to 192.168.40.220, 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 device.

1). Test TCP performance

- **Server side (192.168.40.20)**

The on-server iperf3 uses the -s parameter to indicate that it works in server-side mode.

```
$: iperf3 -s -i 2
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

- **Client(192.168.40.220)**

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

- -c 192.168.40.220: Working on client side, connecting to server side 192.168.40.20
- -i 2: Test results are reported at a time interval of 2 seconds
- -t 10: The total test duration is 10 seconds

```
root@myd-yt113-i:~# iperf3  -c 192.168.40.20 -i 2 -t 10
Connecting to host 192.168.40.20, port 5201
[  5] local 192.168.40.220 port 35712 connected to 192.168.40.20 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-2.01   sec   176 MBytes   736 Mbits/sec    0    185 KBytes

[  5]   2.01-4.00   sec   175 MBytes   736 Mbits/sec    0    212 KBytes

[  5]   4.00-6.01   sec   176 MBytes   737 Mbits/sec    0    223 KBytes

[  5]   6.01-8.00   sec   175 MBytes   736 Mbits/sec    0    223 KBytes

[  5]   8.00-10.01  sec   176 MBytes   737 Mbits/sec    0    355 KBytes


- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.01  sec   879 MBytes   736 Mbits/sec    0             sender
[  5]   0.00-10.01  sec   879 MBytes   736 Mbits/sec                  receive
r

iperf Done.
```

Client after 10 seconds after the end of the test and display the above test results, indicating that the TCP bandwidth of 736Mbits or so, there is no retransmission, the test TCP window value of about 223KBytes at the same time as the server also displays the results of the test are as follows, and then continue to listen to the port waiting for the client to connect:

```
pc$ $ iperf3 -s
-----------------------------------------------------------
```

```
Server listening on 5201
-----------------------------------------------------------
Accepted connection from 192.168.40.220, port 35710
[  5] local 192.168.40.20 port 5201 connected to 192.168.40.220 port 35712
[ ID] Interval           Transfer     Bitrate
[  5]   0.00-1.00   sec  87.5 MBytes   734 Mbits/sec
[  5]   1.00-2.00   sec  87.8 MBytes   736 Mbits/sec
[  5]   2.00-3.00   sec  87.8 MBytes   736 Mbits/sec
[  5]   3.00-4.00   sec  87.8 MBytes   737 Mbits/sec
[  5]   4.00-5.00   sec  87.8 MBytes   737 Mbits/sec
[  5]   5.00-6.00   sec  87.9 MBytes   737 Mbits/sec
[  5]   6.00-7.00   sec  87.7 MBytes   736 Mbits/sec
[  5]   7.00-8.00   sec  87.7 MBytes   736 Mbits/sec
[  5]   8.00-9.00   sec  87.8 MBytes   737 Mbits/sec
[  5]   9.00-10.00  sec  87.9 MBytes   737 Mbits/sec
[  5]  10.00-10.01  sec  1.17 MBytes   691 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate
[  5]   0.00-10.01  sec   879 MBytes   736 Mbits/sec                  receiver
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

2). **Test UDP performance**

- **Server (192.168.40.20)**

Continue running iperf3 on the server using the -s parameter to indicate working in server-side mode.

```
$ iperf3 -s -i 2
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

3). Client (192.168.40.220)

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

- -u : Works in UDP mode
- -c 192.168.40.220 : Working on client side, connecting to server side 192.168.40.20
- -i 2 : Test results are reported at a time interval of 2 seconds
- -t 10 : The total test duration is 10 seconds
- -b 100M : Set UDP transmission bandwidth to 100Mbps.

```
root@myd-yt113-i:~# iperf3 -c 192.168.40.20 -u -i 2 -t 10 -b 100M
Connecting to host 192.168.40.20, port 5201
[  5] local 192.168.40.220 port 33388 connected to 192.168.40.20 port 5201
[ ID] Interval           Transfer     Bitrate         Total Datagrams
[  5]   0.00-2.00   sec  23.8 MBytes   100 Mbits/sec  17261
[  5]   2.00-4.00   sec  23.8 MBytes   100 Mbits/sec  17265
[  5]   4.00-6.00   sec  23.8 MBytes   100 Mbits/sec  17265
[  5]   6.00-8.00   sec  23.8 MBytes   100 Mbits/sec  17265
[  5]   8.00-10.00  sec  23.8 MBytes   100 Mbits/sec  17265
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagra
ms
[  5]   0.00-10.00  sec   119 MBytes   100 Mbits/sec  0.000 ms  0/86321
(0%)  sender
[  5]   0.00-10.00  sec   119 MBytes   100 Mbits/sec  0.025 ms  0/86321
(0%)  receiver

iperf Done.
```

The client completes the test after 10 seconds and displays the above test result, indicating that UDP has no packet loss at the specified bandwidth of 100 MBPS.

The server displays the following test result and continues listening on port 5201 for client connections:

```
pc$ $ iperf3 -s
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
Accepted connection from 192.168.40.220, port 47232
[  5] local 192.168.40.20 port 5201 connected to 192.168.40.220 port 33388
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagra
ms
[  5]   0.00-1.00   sec  11.9 MBytes  99.9 Mbits/sec  0.013 ms  0/8628 (0%)

[  5]   1.00-2.00   sec  11.9 MBytes   100 Mbits/sec  0.042 ms  0/8633 (0%)

[  5]   2.00-3.00   sec  11.9 MBytes   100 Mbits/sec  0.039 ms  0/8632 (0%)

[  5]   3.00-4.00   sec  11.9 MBytes   100 Mbits/sec  0.033 ms  0/8633 (0%)

[  5]   4.00-5.00   sec  11.9 MBytes   100 Mbits/sec  0.064 ms  0/8630 (0%)

[  5]   5.00-6.00   sec  11.9 MBytes   100 Mbits/sec  0.029 ms  0/8635 (0%)

[  5]   6.00-7.00   sec  11.9 MBytes   100 Mbits/sec  0.026 ms  0/8633 (0%)

[  5]   7.00-8.00   sec  11.9 MBytes   100 Mbits/sec  0.037 ms  0/8632 (0%)

[  5]   8.00-9.00   sec  11.9 MBytes   100 Mbits/sec  0.029 ms  0/8633 (0%)

[  5]   9.00-10.00  sec  11.9 MBytes   100 Mbits/sec  0.025 ms  0/8632 (0%)

- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagra
ms
```

```
[  5]   0.00-10.00  sec   119 MBytes   100 Mbits/sec  0.025 ms  0/86321
(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:

```
root@myd-yt113-i:~# iperf3 -u -c 192.168.40.20 -i 2 -t 10 -b 1000M
Connecting to host 192.168.40.20, port 5201
[  5] local 192.168.40.220 port 37838 connected to 192.168.40.20 port 5201
[ ID] Interval           Transfer     Bitrate         Total Datagrams
[  5]   0.00-2.00   sec  74.5 MBytes   313 Mbits/sec  53980
[  5]   2.00-4.00   sec  74.3 MBytes   311 Mbits/sec  53772
[  5]   4.00-6.00   sec  75.1 MBytes   315 Mbits/sec  54376
[  5]   6.00-8.00   sec  75.5 MBytes   317 Mbits/sec  54676
[  5]   8.00-10.00  sec  75.2 MBytes   315 Mbits/sec  54434
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagra
ms
[  5]   0.00-10.00  sec   375 MBytes   314 Mbits/sec  0.000 ms  0/271238
(0%)  sender
[  5]   0.00-10.00  sec   375 MBytes   314 Mbits/sec  0.028 ms  0/271238
(0%)  receiver

iperf Done.-
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagra
ms
[  5]   0.00-10.00  sec   905 MBytes   759 Mbits/sec  0.000 ms  0/655554
(0%)  sender
```

```
[  5]   0.00-10.04  sec   905 MBytes   756 Mbits/sec  0.014 ms  146/655554  (0.022%)  receiver

iperf Done.
```

There are many parameters of iperf3 that can be configured in the process of testing. Users can adjust the test according to the actual application needs. For example, we can increase the -t parameter to stress over a long period of time, or specify the -P parameter to stress test multiple connections concurrently, and so on. For more information about the iperf3 test, see:https://iperf.fr/iperf-doc.php#3doc

6. graphics system

6.1. QT

QT is a cross-platform C++ graphical user interface application development framework. It can be used to develop both GUI and non-GUI programs, such as console tools and servers. Qt is an object-oriented framework that uses special code generation extensions and some macros. Qt is easy to extend and allows for real component programming.

6.1.1. Get information about qt

First check the QT version supported by the current system, as follows:

```
root@myd-yt113-i:~# ls /usr/local/
Qt_5.12.5
```

6.1.2. QT running environment introduction

When running Qt applications, the Qt operating environment, such as platform plug-ins, display parameters, input devices and cursor Pointers, can be properly configured according to different software and hardware requirements.

- **S70MXAPP2 Service**

On embedded Linux systems, multiple platform plug-ins can be used: EGLFS, LinuxFB, DirectFB, or Wayland. However, the availability of these plug-ins depends on the characteristics of the actual hardware platform and how Qt is configured, with the linuxfb plug-in used on MY D-YT113X platform.

In the MYD-YT113X platform, we load the environment variables required to run the QT program through the S70MXAPP2 service, the script is as follows:

```
root@myd-yt113-i:~# cat /etc/init.d/S70MXAPP2
#!/bin/sh
#
# Start the mxapp2
#
```

```
case "$1" in
    start)
                if [ ! -f "/usr/bin/mxapp2" ];then
                        exit 1
                fi

                if [ -d "/usr/local/Qt_5.12.5" ];then
                        export  QTDIR=/usr/local/Qt_5.12.5
                else
                        export  QTDIR=/usr/lib
                fi

                if [ -d $QTDIR ];then

                        export  QT_ROOT=$QTDIR
                        export  PATH=$QTDIR/bin:$PATH
                        export  LD_LIBRARY_PATH=$QTDIR/lib:/usr/lib/cedarx/:
$LD_LIBRARY_PATH

                        export QT_QPA_PLATFORM_PLUGIN_PATH=$QT_ROOT/
plugins
                        export QT_QPA_PLATFORM=linuxfb:tty=/dev/fb0
                        export QT_QPA_FONTDIR=$QT_ROOT/fonts

                        TouchDevice=gt9xxnew_ts
                        for InputDevices in /sys/class/input/input*
                        do
                                DeviceName=`cat $InputDevices/name`
                                if [ "$DeviceName" == "$TouchDevice" ];then
                                        TouchDeviceNum=${InputDevices##*in
put}
```

```
                export QT_QPA_EVDEV_TOUCHSCREEN_PARAMETERS=/dev/input/event$TouchDeviceNum
                echo "add "/dev/input/event$TouchDeviceNum "to Qt Application."
                break
            fi
        done
        if [ ! -n "$TouchDeviceNum" ]; then
            echo "Error:Input device $TouchDevice can not be found,plz check it!"
        fi

        if [ -d "/usr/local/Qt_5.12.5" ];then
            export QT_QPA_PLATFORM=linuxfb
            export QT_QPA_GENERIC_PLUGINS=evdevtouch
            export QT_QPA_EGLFS_INTEGRATION=eglfs_mali
            export QML_IMPORT_PATH=$QTDIR/qml
            export QML2_IMPORT_PATH=$QTDIR/qml
        else
            export QT_QPA_FONTDIR=/usr/lib/fonts
            export QT_QPA_GENERIC_PLUGINS=tslib
            #export QT_QPA_GENERIC_PLUGINS=evdevmouse:/dev/input/event4
            export TSLIB_FBDEVICE=/dev/fb0
            export TSLIB_CONSOLEDEVICE=none
            export TSLIB_TSDEVICE=/dev/input/event4
            export TSLIB_CONFFILE=/etc/ts.conf
            export TSLIB_CALIBFILE=/etc/pointercal
            export TSLIB_PLUGINDIR=/usr/lib/ts
        fi
```

```
                        export QWS_MOUSE_PROTO=
                        ulimit -c unlimited
                        mxapp2 &
                fi
                        sleep 3
                        echo -ne "\033[?25h"
        ;;
    stop)]
        ;;
    *)
        echo "Usage: $0 {start}"
        exit 1
        ;;
esac

exit 0
```

- **Display parameter configuration**

QT applications can write applications that match the screen by obtaining screen display parameters from the QScreen class or QDesktopWidget. Getting the screen resolution through the QScreen or QDesktopWidget, the color depth is generally fine, but sometimes the physical size obtained is not necessarily correct due to display drivers. In this case, you can configure and adjust the following parameters to make the elements displayed on the actual interface fit the size of the display screen.

In general, the default configuration can be used, but if there is a display element and the actual screen does not match the situation, you can make appropriate adjustments to the relevant parameters according to the above description.

- **Enter the peripheral configuration**

When there is no windowing system on an embedded Linux device (such as XWindow or Weston), the mouse, button, or eye device obtains input device information by reading evdev directly or using another intermediate library, such as libinput or tslib. The eglfs and linuxfb platform plug-ins include both types of input. About Qt5 Input device configuration is as follows:

linuxfb platform plug-in default is the use of EvdevTouch input processor, this way is often used to deal with capacitive touch, capacitive touch driver reported event coordinates and the actual screen area coordinates exactly corresponding, there is no need to do additional processing, if the reverse, you can make some adjustments through the environment variables, The EvdevTouch input handler supports the following additional parameters.

As can be seen from the above code, linuxfb platform plug-in default is the use of EvdevTouch input processor, this way is often used to deal with capacitive touch, capacitive touch driver reported event coordinates and actual screen area coordinates exactly corresponding, there is no need to do additional processing, if the reverse, Some adjustments can be made through environment variables, and the EvdevTouch input handler supports the following additional parameters:

Table 6-1. Parameters of environment variables related to the QT Linuxfb plug-in EVDEV touch handler

argument	Description
/dev/input/...	Specify the name of the input device. If not specified, Qt will look for a suitable device either by libudev or by traversing the available nodes.
rotate	On some touch screens, the coordinates must be rotated by setting the coordinates rotate to 90, 180, or 270.
invertx/inverty	Specifies the parameter used to invert the X or Y coordinates in the input event.

For example, if QT_QPA_EVDEV_TOUCHSCREEN_PARAMETERS passes the following value to the platform plug-in before launching the application, the touch device is explicitly specified as "*/dev/inputt/event6*" and its coordinates are flipped 180 degrees. This is useful when the orientation of the actual screen and touch screen do not match.

```
export QT_QPA_EVDEV_TOUCHSCREEN_PARAMETERS=/dev/input/event6:rotate=
180
```

To enable tslib support, you need to set the QT_QPA_EGLFS_TSLIB (for eglfs) or QT_QPA_FB_TSLIB (for linuxfb) environment variable to 1. For more information on how to use tslib, please refer to:https://github.com/libts/tslib/blob/master/README.md.

Note: The tslib input handler is often used for resistive touches, generates mouse events and supports only single touch, and requires screen calibration for initial use.

6.1.3. Start Qt program

When we need to run our own QT program on the MYD-YT113X platform, we need to modify the qtenv script file, add the executable QT program path to the script file, so that the correct environment variables can be configured after running the script. To run the QT program qt_test, make the following changes at the end of the "*/etc/init.d/S70MXAPP2*" script.

```
 export QWS_MOUSE_PROTO=
         ulimit -c unlimited
         #mxapp2 &
         /etc/qt_test &
fi
```

Here we comment out the previous default QT program mxapp2 and add our own QT program qt_test path to the script.

Since mxapp2 has been started by default before, if you want to run your own Qt application at this time, you need to terminate mxapp2 before starting other applications.

You can exit a process directly by running the kill command.

```
root@myd-yt113-i:~# killall mxapp2
```

7. Multimedia applications

7.1. Video playback

- **xplayerdemo tool**

The allwinner system comes with a player xplayerdemo that can decode and play videos.

```
root@myd-yt113-i:/# xplayerdemo
[1970-01-01 00:14:43] WARNING: awplayer <cdx_log_set_level:30>: Set log lev
el to 3
[1970-01-01 00:14:43] ERROR   : awplayer <ReadPluginEntry:198>: read plugin
entry adecoder-14 fail!
[1970-01-01 00:14:43] ERROR   : awplayer <ReadPluginEntry:198>: read plugin
entry vdecoder-10 fail!
INFO    : cedarc <CedarPluginVDInit:80>: register h264 decoder success!
INFO    : cedarc <CedarPluginVDInit:84>: register mjpeg decoder success!
INFO    : cedarc <CedarPluginVDInit:86>: register mpeg2 decoder success!
[1970-01-01 00:14:44] WARNING: awplayer <DlOpenPlugin:112>: Invalid plugin,
function CedarPluginVDInit not found.
INFO    : cedarc <CedarPluginVDInit:98>: register mpeg4dx decoder success!
INFO    : cedarc <CedarPluginVDInit:79>: register mpeg4H263 decoder success!
INFO    : cedarc <CedarPluginVDInit:90>: register mpeg4Normal decoder succ
ess!
[1970-01-01 00:14:44] ERROR   : awplayer <DlOpenPlugin:103>: dlopen 'libaww
mv3.so' fail: libawwmv3.so: cannot open shared object file: No such file or dir
ectory
[1970-01-01 00:14:44] ERROR   : awplayer <CdxPluginLoadList:235>: load vdec
oder id vdecoder.wmv3 fail!
INFO    : cedarc <CedarPluginVDInit:85>: register h265 decoder success!
INFO    : cedarc <CedarPluginVDInit:73>: register vp8 decoder success!
```

```
[1970-01-01 00:14:44] ERROR   : awplayer <ReadPluginEntry:198>: read plugin
entry plugin-0 fail!
[1970-01-01 00:14:44] DEBUG   : awplayer <AwStreamInit:109>: aw stream init...

*****************************************************************************
* This program implements a simple player,
* you can type commands to control the player.
* To show what commands supported, type 'help'.
* Inplemented by Allwinner ALD-AL3 department.
****************************************************************************
[1970-01-01 00:14:44] DEBUG   : awplayer <XPlayerCreate:221>: XPlayerCreate.
[1970-01-01 00:14:44] DEBUG   : awplayer <LogVersionInfo:34>:
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> CedarX <<<<<<<<<<<<<<<<
<<<<<<<<<<<<<<<<
tag    : cedarx
branch: stable_v2.8_common
commit: 967535b8ff6a073cb4f38e85a4ae5fa6008014d8
date   : Mon, 15 May 2017 01:30:22 +0000 (09:30 +0800)
author:
------------------------------------------------------------------


[1970-01-01 00:14:44] DEBUG   : awplayer <LayerCreate_DE:876>: LayerCreate.
[1970-01-01 00:14:44] INFO    : awplayer <LayerCreate_DE:906>: screen_id = 0,
output_type = 1
[1970-01-01 00:14:44] INFO    : awplayer <LayerCreate_DE:909>: the output ty
pe: 0
[1970-01-01 00:14:44] DEBUG   : awplayer <LayerCreate_DE:924>: screen:w 102
4, screen:h 600
[1970-01-01 00:14:44] DEBUG   : ionAlloc <CdxIonOpen:30>: begin ion_alloc_o
pen

INFO    : cedarc <VeInitialize:1495>: *** ic_version = 0x1301000010210,
[1970-01-01 00:14:44] DEBUG  : TinaSoundControl <TinaSoundDeviceInit:146>: TinaSoundDeviceInit()
[1970-01-01 00:14:44] DEBUG  : awplayer <XPlayerSetVideoSurfaceTexture:611>: setVideoSurfaceTexture, surface = 0x940f80
[1970-01-01 00:14:44] DEBUG  : awplayer <XPlayerThread:2016>: process message XPLAYER_COMMAND_SET_SURFACE.
[1970-01-01 00:14:44] DEBUG  : awplayer <XPlayerThread:2077>: ==== process message XPLAYER_COMMAND_SET_SUBCTRL.
[1970-01-01 00:14:44] DEBUG  : awplayer <XPlayerSetDeinterlace:732>: set deinterlace
[1970-01-01 00:14:44] DEBUG  : awplayer <XPlayerThread:2092>: ==== process message XPLAYER_COMMAND_SET_SUBCTRL.

demoPlayer#
```

Check if the player supports operation commands:

```
demoPlayer# help

********************************************************************************
* This is a simple media player, when it is started, you can input commands to tell
* what you want it to do.
* Usage:
*    # ./demoPlayer
*    # set url: http://www.allwinner.com/ald/al3/testvideo1.mp4
*    # show media info
*    # play
*    # pause
*    # stop
*
* Command and it param is seperated by a colon, param is optional, as below:
*        Command[: Param]
```

```
*
* here are the commands supported:
*      help:
*                     show this help message.
*      quit:
*                     quit this program.
*      set url:
*                     set url of the media, for example, set url: ~/testfile.mkv.
*      play:
*                     start playback.
*      pause:
*                     pause the playback.
*      stop:
*                     stop the playback.
*      set speed:
*                     set seek with n times speed.
*      set fspeed:
*                     set seek with float times speed.
*      seek to:
*                     seek to specific position to play, position is in unit of second,
 ex, seek to: 100.
*      show media info:
*                     show media information of the media file.
*      show duration:
*                     show duration of the media file.
*      show position:
*                     show current play position,in unit of second.
*      switch audio:
*                     switch audio to a track, for example, switch audio: 2, track is
start counting from 0.
*      loop:
*                     set looping.
*     set play speed:
*                     set playback with n times speed.
*
**********************************************************
```

7.1.2. Play video

Let's play the mounted u disk h264_output.mp4 as an example:

```
demoPlayer# set url:/run/media/sda1/T113/1080p_30fps_h265.mp4
[1970-01-01 00:02:13] DEBUG  : awplayer <XPlayerSetDataSourceUrl:476>: set
DataSource(url), url='/run/media/sda1/T113/1080p_30fps_h265.mp4'
[1970-01-01 00:02:13] INFO   : awplayer <XPlayerThread:1853>: process mess
age XPLAYER_COMMAND_SET_SOURCE.
[1970-01-01 00:02:13] DEBUG  : awplayer <XPlayerPrepare:781>: prepare
[1970-01-01 00:02:13] DEBUG  : awplayer <XPlayerThread:2106>: process mess
age XPLAYER_COMMAND_PREPARE. mPriData->mStatus: 1
[1970-01-01 00:02:13] DEBUG  : demuxComponent <DemuxThread:1816>: pro
cess message DEMUX_COMMAND_PREPARE.
[1970-01-01 00:02:13] DEBUG  : awplayer <CdxParserPrepare:908>: source uri
'file:///run/media/sda1/T113/1080p_30fps_h265.mp4'
[1970-01-01 00:02:13] DEBUG  : awplayer <__FileStreamConnect:426>: (6/0/402
17864) path:'file:///run/media/sda1/T113/1080p_30fps_h265.mp4'
[1970-01-01 00:02:13] DEBUG  : awplayer <CdxParserCreate:847>: Good, it's '
mov'
[1970-01-01 00:02:13] DEBUG  : demuxComponent <PrintMediaInfo:469>: ****
*****PrintMediaInfo begin*********
[1970-01-01 00:02:13] DEBUG  : demuxComponent <PrintMediaInfo:483>: fileS
ize = 40217864, bSeekable = 1, duration = 30634, audioNum = 1, videoNum
 = 1, subtitleNum = 0
[1970-01-01 00:02:13] DEBUG  : demuxComponent <PrintMediaInfo:503>: ***V
ideo[0]*** eCodecFormat = 0x116, nWidth = 1920, nHeight = 1080, nFrameRa
te = 30000, nFrameDuration = 0, bIs3DStream = 0, bSecureFlag = 0
```

```
[1970-01-01 00:02:13] DEBUG   : demuxComponent <PrintMediaInfo:521>: ***Audio[0]*** eCodecFormat = 0x4, eSubCodecFormat = 0x0, nChannelNum = 2, nBitsPerSample = 16, nSampleRate = 44100
[1970-01-01 00:02:13] DEBUG   : demuxComponent <PrintMediaInfo:537>: *********PrintMediaInfo end*********
INFO    : cedarc <log_set_level:71>: Set log level to 5 from /vendor/etc/cedarc.conf
ERROR   : cedarc <DebugCheckConfig:360>: now cedarc log level:5
[1970-01-01 00:02:13] DEBUG   : awplayer <VideoDecCompSetVideoStreamInfo:260>: ++++++++ pVconfig->bGpuBufValid = 1,nGpuAlignStride = 32
[1970-01-01 00:02:13] DEBUG   : awplayer <VideoDecCompSetVideoStreamInfo:304>: enable afbc mode for 4k
[1970-01-01 00:02:13] DEBUG   : awplayer <__LayerResetNativeWindow:728>: LayerResetNativeWindow : (nil)
[1970-01-01 00:02:13] DEBUG   : awplayer <VideoRenderCompSetDeinterlace:363>: video render component setting deinterlace: 0x163818
[1970-01-01 00:02:13] DEBUG   : awplayer <PlayerConfigDropDelayFrame:1822>: PlayerConfigDropDelayFrame

[1970-01-01 00:02:13] DEBUG   : awplayer <PlayerConfigDropDelayFrame:1825>: VideoDecCompSetDropDelayFrames

[1970-01-01 00:02:13] DEBUG   : awplayer <AudioRenderCompSetAudioSink:195>: audio render component setting AudioSink
setDataSource end
[1970-01-01 00:02:13] DEBUG   : awplayer <XPlayerPrepareAsync:761>: prepareAsync
[1970-01-01 00:02:13] DEBUG   : awplayer <XPlayerThread:2106>: process message XPLAYER_COMMAND_PREPARE. mPriData->mStatus: 4
[1970-01-01 00:02:13] INFO    : awplayer <XPlayerThread:2142>: xxxxxxxxxx video size: width = 1920, height = 1080
++++ video width: 1920, height: 1080
```

```
[1970-01-01 00:02:13] DEBUG  : awplayer <CallbackForAwPlayer:392>: info : p
reared
info: prepare ok.
preparing...
```

This indicates that the playback is ready. Run the play command to start the playback.

```
demoPlayer# play
[1970-01-01 00:02:45] DEBUG  : awplayer <XPlayerStart:811>: start
[1970-01-01 00:02:45] DEBUG  : awplayer <XPlayerThread:2314>: process mess
age XPLAYER_COMMAND_START.
[1970-01-01 00:02:45] DEBUG  : awplayer <PlayerStart:740>: player start
[1970-01-01 00:02:45] DEBUG  : awplayer <BaseCompPostAndWait:61>: video
decoder receive cmd: start
[1970-01-01 00:02:45] DEBUG  : awplayer <BaseCompPostAndWait:61>: audio
decoder receive cmd: start
(Allwinner Audio Middle Layer),line(971) : Create Decoder!!=====
[1970-01-01 00:02:45] DEBUG  : awplayer <handleStart:1099>: Create libadeco
der success...
(Allwinner Audio Middle Layer),line(603) : AudioDec_Installaudiolib ok
(Allwinner Audio Middle Layer),line(606) : audio decoder init start ...
(AllwinnerAlibs),line(50) :
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> Audio <<<<<<<<<<<<<<<<
<<<<<<<<<<<<<<<
tag    : audiocodec-v1.2
branch: new
commit: 3ba65962c01cbf1280ddda19d843009b6ef8ce85
date   : Tue Jan 8 16:25:27 2019 +0800
......
```

7.2. Audio

This chapter is about testing playback audio.

7.2.1. Debugging tool

- **Tinyplay audio tool**

Put the audio file into the USB flash drive, and then connect the speaker to the J3 port.

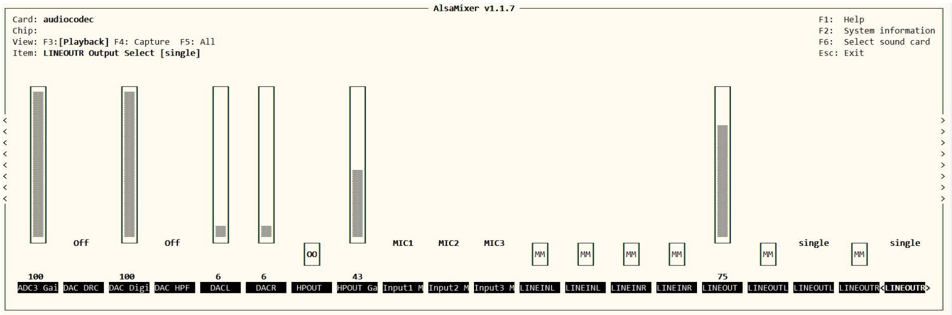
```
Usage: tinyplay file.wav [-D card] [-d device] [-p period_size] [-n n_periods] [-T capture time]
[root@myir:/mnt/usb/sda1]# tinyplay 01+Singalongsong.wav
playing '01+Singalongsong.wav': 2 ch, 48000 hz, 16 bit
```

- **alsamixer**

Volume adjustment tool. Run the alsamixer command to adjust the left and right channels and volume.

```
root@myd-yt113-i:~# alsamixer -a
alsamixer: option requires an argument -- 'a'
Usage: alsamixer [options]
Useful options:
  -h, --help              this help
  -c, --card=NUMBER       sound card number or id
  -D, --device=NAME       mixer device name
  -V, --view=MODE         starting view mode: playback/capture/all
Debugging options:
  -g, --no-color          toggle using of colors
  -a, --abstraction=NAME  mixer abstraction level: none/basic
```

The following figure so, use the direction “←” “→” keys to control the cursor to select the modification items, use “↑” “↓ ” to adjust the parameter size.



```
AlsaMixer v1.1.7
Card: audiocodec                                                          F1:  Help
Chip:                                                                     F2:  System information
View: F3:[Playback] F4: Capture  F5: All                                  F6:  Select sound card
Item: LINEOUTR Output Select [single]                                     Esc: Exit
```

Figure 7-1. Player parameters adjust the UI

8. References

- **Linux kernel Open source community**

https://www.kernel.org/

- **Allwinner Development community**

https://www.aw-ol.com

- **Linux kernel watchdog introduction**

https://www.kernel.org/doc/html/latest/watchdog/index.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