MYD-YT507H Linux System Evaluation Guide

File Status : [ ] Draft [√] Release	FILE ID :	MYIR-MYD-YT507H-SW-DG-ZH-L4.9.170
	VERSION :	V1.0(DOC)
	AUTHOR :	Zhaoy
	CREATED :	2022-03-30
	UPDATED :	2022-04-26

Revision History

version	The author	participants	The date of	note
V1.0	Zhaoy		2022.03.21	The initial release

CONTENT

1. overview

The Linux Software Evaluation Guide is used to describe the testing procedures and evaluation methods for core and peripheral resources running in the open Source Linux system on the Mir development board.This article can be used as an early assessment guide or as a test guide for general system development.

1.1. Hardware resources

MYD-YT507H board card of Mir Electronics is composed of core plate MYC-YT507H and bottom plate MYB-YT507H. The core plate and bottom plate are welded by stamp hole.In addition, MYIR provides a wealth of software resources and documentation.For details on hardware configuration parameters, please refer to MYD-YT507H Product Manual.At the same time, users will use some accessories in the evaluation test process, see the following list.

Table 1-1. Optional modules

accessories	The interface way	Description and Links
camera	MIPI/ parallel	MIPI camera : http://www.myir-tech.com/product/my_cam003m.htm Parallel camera : http://www.myir-tech.com/product/my_cam011b.htm
LCD screen	LVDS interface 7 inch	7 inch LVDS screen: http://www.myir-tech.com/product/my-lvds070c.htm
4 g module	PCIE	EC20CEFDKG： https://www.quectel.com/cn/product/ec20r21minipcIe.htm
Expansion board module	Raspberry PI interface	MY-WiredCom： http://www.myir-tech.com/product/my-wiredcom.htm
WiFi module	SDIO	MY-WF005S： http://www.myir-tech.com/product/my-wf05s.htm

1.2. Software resources

The BSP of MYD-YT507H development board is based on the transplantation and modification of the official Open source community edition Linux BSP, and the system image is built by the Buildroot project.Bootloader, Kernel and all parts of the file system software resources are open in the form of source code. For details, please refer to MYD-YT507H_SDK Release Notes. The development board has burned a mirror image according to the core board model when leaving the factory. You only need to power it on to use it.

1.3. Document resources

Depending on the user's different stages of using the development board, the SDK contains different types of documents and manuals for each stage, such as release instructions, evaluation guides, development guides, application notes, common questions and answers, etc.For details, see Table 2-4 in MYD-YT507H_SDK Release Notes.

1.4. Environment to prepare

Before you start evaluating the development board software, you need to do the necessary preparation and configure the basic environment for the development board, including the correct hardware wiring, configuring the debugging serial port, setting up the startup steps, and so on.For detailed steps, see myD-Y T507 Quick Start Guide. The following sections focus on how to evaluate and test the hardware resources and interfaces as well as the software functions of the system. Mainly use some common tools and commands under Linux, as well as their own applications to test.Software evaluation guide is divided into several parts to describe, including: core resources, peripheral resources, network applications, multimedia applications, development support applications, system tools and other categories.The following chapters will give a comprehensive explanation of each part and describe in detail the specific evaluation methods and steps for each part of resources.

2. core resources

On Linux systems, a 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 following will specifically read and test the parameters of CPU, memory, eMMC, RTC and other core resources.

2.1. CPU

1) Command to view CPU information

Read CPU provider and parameter information in the system, can be obtained from the /proc/cpuinfo file.

```
[root@myir:/]# cat /proc/cpuinfo
processor       : 0
BogoMIPS: 48.00
Features        : fp asimd aes pmull sha1 sha2 crc32
CPU implementer : 0x41
CPU architecture: 8
CPU variant     : 0x0
CPU part        : 0xd03
CPU revision    : 4

processor       : 1
BogoMIPS: 48.00
Features        : fp asimd aes pmull sha1 sha2 crc32
CPU implementer : 0x41
CPU architecture: 8
CPU variant     : 0x0
CPU part        : 0xd03
CPU revision    : 4

processor       : 2
```

```
BogoMIPS: 48.00
Features          : fp asimd aes pmull sha1 sha2 crc32
CPU implementer : 0x41
CPU architecture: 8
CPU variant       : 0x0
CPU part          : 0xd03
CPU revision      : 4

processor         : 3
BogoMIPS: 48.00
Features          : fp asimd aes pmull sha1 sha2 crc32
CPU implementer : 0x41
CPU architecture: 8
CPU variant       : 0x0
CPU part          : 0xd03
CPU revision      : 4
```

- Processor: The number of a logical processing core in a system. For multi-core processors, this can be a physical core or a virtual logical core using hyperthreading technology
- Modelname: indicates the name and number of the CPU
- BogoMIPS: when the system is the kernel starts a rough measure of the CPU running one million instructions per second (MillionInstructionsPerSecond)

2) Viewing CPU Usage

To view the CPU usage of T507 series chips, perform the following operations:

```
[root@myir:/]# top
Mem: 174156K used, 829584K free, 84K shrd, 2244K buff, 62640K cached
CPU:   0% usr   2% sys   0% nic  97% idle   0% io   0% irq   0% sirq
Load Average: 0.00 0.00 1/126 10461
  PID  PPID USER     STAT   VSZ %VSZ %CPU COMMAND
10453  1890 root     R      2580   0%   2% top
 1523     1 root     S      156m  16%   0% adbd
 1694     1 root     S      148m  15%   0% /etc/video2lcd
 1890     1 root     S      3628   0%   0% -/bin/sh
```

```
 1335     1 root     S     2716   0%   0% dbus-daemon --system
    1        0 root     S     2580   0%   0% init
 1306     1 root     S     2580   0%   0% /sbin/syslogd -n
 1311     1 root     S     2580   0%   0% /sbin/klogd -n
 1408     1 root     S     2328   0%   0% /usr/sbin/dropbear -R
 1388     1 root     S     2192   0%   0% /sbin/dhcpcd -f /etc/dhcpcd.conf
 1447     1 root     S     2160   0%   0% /usr/sbin/tftpd -c -l -s /var/lib/tf
tp
   34        2 root     SW       0   0%   0% [kworker/u8:2]
  587        2 root     SW       0   0%   0% [vsync proc 0]
    7        2 root     SW       0   0%   0% [rcu_preempt]
 1228     2 root     SW       0   0%   0% [cec thread]
  916        2 root     SW       0   0%   0% [kworker/0:1]
  640        2 root     SW       0   0%   0% [kworker/2:1]
 1226     2 root     SW       0   0%   0% [hdmi proc]
 1234     2 root     SW       0   0%   0% [tve detect]
 1495     2 root     SW       0   0%   0% [mali-simple-pow]
```

- %usr: indicates CPU usage of user-space programs (not scheduled by NICE)
- %sys: indicates the CPU usage of system space, mainly kernel programs
- % NIC: indicates the CPU usage of the user space and programs scheduled by NICE
- %idle: indicates the idle CPU
- % irQ: number of hard interrupts processed by the CPU
- % sirQ: number of soft interrupts processed by the CPU

3) Obtain CPU temperature information

The built-in temperature sensor of the CPU can be used to collect the CPU temperature.

```
[root@myir:/]# cat /sys/class/thermal/thermal_zone0/temp
40967
```

The number shown above is thousandths of a degree, divided by 1,000 to give the current temperature.

```
[root@myir:/]# echo "scale=5000;4*a(1)" | bc -l -q &
[2], 3300
[1]    Done(127)                    echo "scale=5000;4*a(1)" | bc-l-q
```

The above command will compute the PI in the background and be accurate to 5000 decimal places.The calculation process takes a while.At this point, we can check the CPU utilization change with the top command, as shown below:

```
[root@myir:/]# top
Mem: 174980K used, 828760K free, 84K shrd, 2244K buff, 62704K cached
CPU:  25% usr   2% sys   0% nic  72% idle   0% io   0% irq   0% sirq
Load Average: 0.69 0.37 0.15 3/128 14112
  PID  PPID USER      STAT    VSZ %VSZ %CPU COMMAND
12581  1890 root      R       1940   0%  25% bc -l -q
```

About 3 minutes later, the PI result is calculated.If the CPU usage reaches 100% and no exception occurs, the CPU pressure test passes.You can also continue to increase the exact value, which can further improve the test pressure.

```
Root @ myir: / # 3.14159265358979323846264338327950288419716939937510
5820 97494459230781640628620899862803482534211706798214808651328230
66470, 9384460955058,
.
74351362222477158915049530984448933309634087807693259939780541934l4,
 47377441842631298608099888687413260472O,
```

2.2. GPU

Graphics Processing Unit (GPU), also known as display core, visual processor, and display chip, is a microprocessor that specializes in performing image and graphic-related operations on personal computers, workstations, game consoles, and some mobile devices (such as tablet computers, smart phones, etc.).As the core of the display system, the GRAPHICS processor has powerful data computing capability, and realizes the functions of 2D/3D (Two Dimensions /Three Dimensions, 2D/3D) graphics processing, image processing and display control in the form of hardware accelerator.

MYD-YT507H chip internal GPU module, support 2D, 3D acceleration, OpenGL ES1.1,2.0,3.0,3.1, OpenCL 1.2, and QT graphics system.

For details, see 6.1 Graphics and Image Processing.

2.3. Memory

MYD-YT507H memory is available in 1GByte and 2GByte versions. The system will divide the memory into device memory (CMA) and system memory (MEM).Device memory is the contiguous space used by the driver, and system memory is the space allocated to the user mode.

1) Viewing Memory Information

Read the memory parameter information in the system, can be obtained from the /proc/meminfo file.

```
[root@myir:/]# cat /proc/meminfo
MemTotal:        1003740 kB
MemFree:          828948 kB
MemAvailable:     886756 kB
Buffers:            2244 kB
Cached:            62704 kB
SwapCached:            0 kB
Active:            22300 kB
Inactive:          57392 kB
.
```

- MemTotal: All available RAM size, physical memory minus reserved and kernel usage
- MemFree: LowFree + HighFree
- Buffers: Size used to cache block devices
- Cached: indicates the buffer size of the file
- SwapCached: memory that has been swapped out.Associated with I/O
- Active: frequently (recently) used memory
- Inactive: memory that is not used recently

2) Obtaining memory Usage

You can use the free command to read the memory usage. The -m parameter stands for MByte.

```
[root@myir:/]# free -m
             total       used       free     shared    buffers     cached
Mem:           980        170        809          0          2         61
-/+ buffers/cache:        107        872
Swap:            0          0          0
```

- Total: indicates the total memory capacity
- Used: Indicates the amount of memory used
- Free: indicates the available memory

3) Memory stress test

Given the size and times of the test memory, the existing memory of the system can be tested on pressure.You can use the system tool memtester to test. For example, set the memory size to 512MB, test times to 10, and test command to memtester 512M 10.

The following uses 512MB of memory as an example:

```
[root@myir:/]# memtester  512M
Memtester version 4.3.0 (64 - bit)
Copyright (C) 2001-2012 Charles Cazabon.
Licensed under the GNU General Public License version 2 (only).

pagesize is 4096
pagesizemask is 0xfffffffffffff000
want 200MB (209715200 bytes)
got  200MB (209715200 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.
```

2.4. eMMC

This section describes eMMC testing, which is suitable for development boards with eMMC memory. EMMC is a data storage device that includes a MultiMediaCard (MMC) interface and a NAND Flash component.Its cost, small size, Flash technology independence, and high data throughput make it an ideal choice for embedded products.

1) View the eMMC capacity

You can run the fdisk-l command to query information and capacity of eMMC partitions.

```
[root@myir:/]# fdisk -l
Found valid GPT with protective MBR; using GPT

Disk /dev/mmcblk0: 15269888 sectors, 3360M
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 15269854

Number  Start (sector)    End (sector)  Size  Name
     1           73728          139263  32.0M  boot-resource
     2          139264          172031  16.0M  env
     3          172032          303103  64.0M  boot
     4          303104         4497407  2048M  rootfs
     5         4497408         8691711  2048M  rootfsbak
     6         8691712         8922783   112M  recovery
     7         8922784         8955551  16.0m  private
     8         8955552        15269854  3083M  UDISK
Disk /dev/mmcblk0boot1: 8 MB, 8388608 bytes, 16384 sectors
256 cylinders, 4 heads, 16 sectors/track
Units: sectors of 1 * 512 = 512 bytes
```

```
Disk /dev/mmcblk0boot1 doesn't contain a valid partition table
Disk /dev/mmcblk0boot0: 8 MB, 8388608 bytes, 16384 sectors
256 cylinders, 4 heads, 16 sectors/track
Units: sectors of 1 * 512 = 512 bytes
Disk /dev/mmcblk0boot0 doesn't contain a valid partition table
```

2) View eMMC partition information

You can run the df command to query information about eMMC partitions, usage, and mount directories.

```
[root@myir:/]# df -h
Filesystem                Size      Used Available Use% Mounted on
/dev/mmcblk0p4 1.9g 803.6m 1.1g 41% /
TMPFS 490.1m 64.0k 490.0m 0% / TMP
TMPFS 490.1m 20.0k 490.1m 0% /run
Devtmpfs 480.7m 0 480.7m 0% /dev
/dev/mmcblk0p8 2.9g 9.0m 2.7g 0% /media
/dev/shm /dev/shm /dev/shm /dev/shm
```

- /dev/mmcblk0p4: indicates the root file system, mounted to the root directory.
- TMPFS: memory virtual file system, mounted to different directories.
- Devtmpfs: used to create dev for the system.
- /dev/mmcblk0p8: Can be used for user partition

3) Performance testing of eMMC

The performance test mainly tests the speed of eMMC reading and writing files in Linux. It is generally combined with the time and DD commands.

- **Write file test**

```
[root@myir dir1]# time dd if=/dev/zero of=write_file bs=100M count=1 conv
=fsync
1+0 records in
1+0 recor      ds out

The real 0 m5. 071 s
```

```
The user 0 m0. The 001 s
Sys 0 m0. The 407 s
```

The eMMC write speed is 19.71 MB/s.

- **Read file test**

```
[root@myir dir1]# time dd if=/dev/zero of=write_file bs=100M count=1 conv
=fsync
1+0 records in
1+0 recor      ds out

Real 0 m3. The 451 s
The user 0 m0. The 001 s
Sys 0 m0. The 407 s
```

The read speed of data directly from the disk is 28.98 MB/s.

2.5. RTC

Real-time clock (RTC) is a clock used to record the Real time. After the software system is shut down, the system time is retained and the timing continues. After the software system is restarted, the time is synchronized to the software system. MYD-YT507H has internal RTC and external RTC (RX8025). If the actual product does not require very high POWER consumption of RTC and the power outage time is required within a month, The RTC test is performed using the hwclock and date commands commonly used in Linux. In the following test, the system time is written to the RTC, the RTC time is read, the SYSTEM time is set to the system time, and the power failure duration test is performed.

1) View RTC devices

```
root@myir:~# ls /dev/rtc* -al
lrwxrwxrwx 1 root root        4 Sep 20 10:24 /dev/rtc -> rtc0
crw------- 1 root root 251, 0 Sep 20 10:24 /dev/rtc0
```

2) Setting the System Time

Mon Feb 7 09:28:30 UTC 2022

```
/ root @ myir: / # date 020709282022.00
Mon Feb  7 09:28:30 UTC 2022
```

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

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

```
root@myir:~#  hwclock -w
```

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

```
[root@myir:/]# hwclock -r
Mon Feb  7 09:29:03 UTC 2022
```

- **Maintain RTC duration during power failure**

Shut down the development board and disconnect the power supply. After a few minutes or so, power it on again.View the RTC time and system time:

```
[root@myir:/]# hwclock -r
The 2022-02-07 09:47:34.524613 + 00:00
```

The RTC time and system time displayed after the restart are about 20 minutes longer than those previously set, indicating that the RTC is working properly.If you need to test the accuracy of the RTC in detail, you can extend the power outage for example 24 hours to test the difference between the RTC time and the standard time.

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

```
root@myir:~# hwclock -s
root@myir:~# date
Mon Feb 7 09:49:03 UTC 2022
```

If you add the hwclock-s command to the startup script, the system time can be synchronized with the RTC time at each startup.

2.6. Watchdog

Linux kernel contains Watchdog subsystem. During the hardware design process, Watchdog timer inside the chip or external Watchdog chip can be used to realize Watchdog function, which is used to monitor the operation of the system.The system will automatically reset when the system is abnormal and cannot feed the dog.Quan Zhi T507 series chip has 3 watchdog.It has the following features: The MPU has two independent watchdog dogs (IWDG1 and IWDG2).IWDG1 is on the secure bus and can only be used by secure applications in a secure environment.Independent watchdogs (IWDG1 and IWDG2) are clocked by a low speed clock (LSI), so they remain active even if the master clock fails.As such, they are best suited for applications that require the watchdog to run as a completely independent process outside of the main application.IWDGs is best suited for recovering from unexpected software or hardware failures.The MYD-YT507H platform uses watchdog IWDG2 by default.The MCU end has a watchdog WWDG1, which is clocked by APB1 and provides reset and early interrupt signals.The early interrupt provided by WWDG1 (WWDG1_ IT) is connected to both MPU (GIC) and MCU (NVIC) interrupt controllers.It allows the application to decide which processor should handle urgent tasks when needed. When using a watchdog, pay attention to the following features:

NOWAYOUT feature: If you want to start the watchdog without stopping it, that is, any upper-layer actions to close the watchdog are not supported, you can use the NOWAYOUT function, for example: CONFIG_WATCHDOG_NOWAYOUT=y With this enabled, this configuration needs to be supported in the specific WDT driver by calling wa tchdog_set_nowayout () to set the use of WDOG_NO_WAY_OUT.The MYD-YT507H watchdog driver supports this feature.

Magic Close feature: When the user wants to stop the watchdog, a character "V" is written into the WDT node to stop the watchdog.MYD-YT507H Watchdog driver does not support closing the watchdog in this way. This section demonstrates how to use watchdog, tests the watchdog system reset function by simulating a kernel crash, and provides an example for setting the watchdog timeout period.

1) User space test door Watchdog

- **Simulated kernel crash**

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

```
[root@myir:/]# echo c > /proc/sysrq-trigger
[9548.559530] SYSRq: sySRq: Trigger a crash
[9548.564224] Internal error: Accessing user space Memory outside uaccess. H
 ROUTINES: 96000045 [#1
[9548.575374] Modules Linked in: Mali_kbase (O)
[9548.580193] CPU: 2 PID: 1890 Comm: sh Tainted: G O 4.9.170 #1
[9548.587915] Sun50IW9 MYIR-YT507H (DT)
[9548.593399] Task: ffffFFC038BC0e80 task.stack: ffffFFC039570000
[9548.600065] PC is at SYSRq_HANDLE_CRASH +0x28/0x34
[9548.605353] LR is at SYSRq_HANDLE_CRASH +0x14/0x34
[9548.610640] PC: [< FFFFFF80084E7B00 >] LR: [< FFFFFF80084E7AEC >] PStat
e: 60400145
```

2) application tests the watchdog

The following uses the application to set the timeout period for the open door dog and the feeding time for the dog.

- **Set the watchdog timeout period**

The timeout period is realized by using the ioctl command: WDIOC_SETTIMEOUT. A parameter, timeout, is required.Example:

```
ioctl(fd, WDIOC_SETTIMEOUT, &timeout);
```

The above is the reference code for setting the current timeout period of the watchdog.Fd is the file handle of the watchdog device.

- **Watchdog application test**

Compile the production file watchdog and copy it to the development board. Run the following command:

```
root@myir:~# ./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()
```

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

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

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

2.7. PMIC

This section 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 are triggered by writing "Freeze" and "mem" to the "/sys/power/state" file in user space, respectively.MYD-YT507H supports freeze and MEM.

1) View the modes supported by the current development board

```
[root@myir:/]# cat /sys/power/state
freeze mem
```

2) Method of writing in user space

```
[root@myir:/]# echo "freeze" > /sys/power/state
[root@myir:/]# echo "mem" > /sys/power/state
```

- **Mem dormancy**

After you run the hibernation command, the development board hibernates, and the debugging serial port cannot be entered again. In this case, the system and device states are saved to the memory (in self-refresh mode, and their contents have been reserved), and all devices enter the low-power mode.

```
[root@myir:/]# echo "mem" > /sys/power/state
[557.683513] PM: Suspend Entry 1970-01-01 00:09:17.527363512 UTC
[557.690227] PM: Syncing filesystems... done.
[557.697346] PM: Preparing System for Sleep (MEM)
[557.896162] Freezing user space processes... (0.001 seconds elapsed) done.
[557.905635] Freezing remaining Freezable Tasks... (0.001 seconds elapsed) do
ne.
[557.915669] PM: Champion System (MEM)
```

- **Freeze dormancy**

After you run the hibernation command, the development board hibernates, and the debugging serial port cannot be entered any more. In this case, the user space is frozen, all I/O devices enter the low-power state, and the processor enters the idle state.

```
[root@myir:/]# echo "freeze" > /sys/power/state
[1507.369691] PM: Suspend Entry 1970-01-01 00:25:07.213720350 UTC
[1507.376418] PM: Syncing filesystems...
[1507.381029] SUNXI-MMC SDC2: SDC set ios: CLK 0Hz BM PP PM UP VDD
22 Width 1 Timing LEGACY(SDR12) DT B
[1507.405864] SUNXI-MMC SDC2: SDC set ios: CLK 400000Hz BM PP PM ON
VDD 22 Width 1 Timing LEGACY(SDR12) DT B
[1507.432562] SUNXI - MMC SDC2: SDC set ios: CLK 400000Hz BM OD PM O
N VDD 22 Width 1 Timing LEGACY(SDR12) DT B
[1507.443546] SUNXI - MMC SDC2: SDC set ios: CLK 400000Hz BM OD PM O
N VDD 22 Width 1 Timing LEGACY(SDR12) DT B
[1507.456967] SUNXI - MMC SDC2: SDC set ios: CLK 400000Hz BM OD PM O
N VDD 22 Width 1 Timing LEGACY(SDR12) DT B
[1507.520768] SUNXI-MMC SDC2: SDC set ios: CLK 400000Hz BM PP PM ON
VDD 22 Width 1 Timing LEGACY(SDR12) DT B
[1507.534474] SUNXI - MMC SDC2: SDC set ios: CLK 400000Hz BM PP PM O
N VDD 22 Width 8 Timing LEGACY(SDR12) DT B
[1507.547795] SUNXI - MMC SDC2: SDC set ios: CLK 400000Hz BM PP PM O
N VDD 22 Width 8 Timing MMC-HS200 dt B
[1507.558740] SUNXI - MMC SDC2: SDC set ios: CLK 100000000Hz BM PP P
M ON VDD 22 Width 8 Timing MMC-HS200 dt B
[1507.571001] done.
[1507.573248] PM: Preparing System for Sleep (freeze)
```

At this point, press user key S2 to wake up the system:

```
[root@myir:/]# [1558.783587] libPHY: gMAC1: probed
[1558.790185] libphy: gMAC0: probed
[1558.790609] SUNXI-gMAC gMAC0 eth0: eth0: Type(7) PHY ID 0000010A at
0 IRQ poll (gMAC0-0:00)
```

```
[1558.813892] SUNXI-gMAC GMAC1 eth1: eth1: Type(6) PHY ID 00000128 at 0 IRQ poll (gMAC1-0:00)
[1558.865964] Error Reading temperature for GPU Thermal zone: -11
[1560.872832] sunxi-gmac gMAC0 eth0: Link is up-1Gbps/full-flow control off
```

The debugging serial port can be re-entered.

3. Basic peripheral interface

3.1. GPIO

1) Set the gpioset command

The gpioset command is used to set the value of the specified GPIO line.

In fact, starting with Linux 4.8, it is no longer recommended to use the SYSFS interface (/sys/class/ GPIO) to manipulate GPIO. Instead, it is recommended to use character devices in user space. Libgpiod is a library for manipulating GPIO character devices. Convenient for developers to debug.

Libgpiod contains a series of commands. Besides gpioset, there are gpioget, gpiodetect, gpioinfo, gpiofind, and gpiomon commands.

- **Grammar:**

```
gpioset [OPTIONS] <chip name/number> <offset1>=<value1> <offset2>=<value2> ...
```

- **Options:**
  - -l, --active-low: sets the low level to the active level
  - - B - bias = [as-is | disable | the pull - down | pull - up] : set bias, (the default use as-is)
  - - D - drive = [push - pull | open - drain | tapping the open - source] : set the drive mode (the default use the push - pull)
  - - m - mode = [exit | wait | time | signal] : set after the completion of the mode of action
  - -s, -- SEC =SEC: Specifies the time to wait when using the --mode=time option (in seconds).
  - -u, --usec= usec: Specifies the time to wait when using the --mode=time option (in microseconds)
  - -b, --background: Separated from the control terminal after setting
  - -h, --help: View the help and exit
  - -v, --version: Displays the version information and exits
  - Parameters: Specify gpiochip and offset (inline offset), and the corresponding value. Multiple lines can be specified at the same time.

- **Example:**

Check MYD-YT507H system gpiochip0 partial line information.

```
[root@myir:/]# gpioinfo gpiochip0
gpiochip0 - 32 lines:
        line   0:      unnamed         unused   input  active-high
        line   1:      unnamed         unused   input  active-high
        line   2:      unnamed         unused   input  active-high
        line   3:      unnamed         unused   input  active-high
        line   4:      unnamed         unused   input  active-high
        line   5:      unnamed         unused   input  active-high
        line   6:      unnamed         unused   input  active-high
        line   7:      unnamed         unused   input  active-high
        line   8:      unnamed         unused   input  active-high
        line   9:      unnamed         unused   input  active-high
        line  10:      unnamed         unused   input  active-high
```

Read the value of gpiochip0 line0.

```
[root@myir:/]# gpioget gpiochip0 0
1
```

Set line 0 of Gpiochip1 to 0 (low level).

```
sudo gpioset gpiochip1 0=0
```

Read line 0 of gpiochip1 again.

```
gpioget gpiochip0 0
0
```

2) gpio -s command is configured

You can also use the gPIO -s command directly to control gPIO Settings based on pin numbers.To set GPIOF14, run the following command.

```
[root@myir:/]# gpio -s PD25 1 0 0 0
set PD25 function=1
set PD25 data=0
set PD25 dlevel=0
set PD25 pull=0
```

Parameters:

- Function: reuse
- Data: indicates the level data
- Plevel: Drive ability
- Pull: pull up and down

3.2. LED lights

Linux provides an independent subsystem to facilitate the operation of LED devices from user space. This subsystem provides an interface for LED devices in the form of files.These interfaces are located in the/SYS /class/leds directory.In the hardware resources list, we have listed all the leds on the development board.The following tests LED by reading and writing sysFS commands.The following commands are general commands and are general methods for manipulating leds.

1) directory for operating leds is/SYS /class/leds

```
root@myir:/sys/class/leds#ls
blue
```

Through to the/sys/class/leds/bule/brightness to destroy the light of different values can change heart light duty ratio.

2) Take heartbeat lamp 1 as an example to test LED

- **Turn off the heartbeat light**

```
root@myir:/sys/class/leds#  echo none > /sys/class/leds/blue/trigger
```

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

- **Put out the LED**

```
root@myir:/sys/class/leds# echo 1 > /sys/class/leds/blue/brightness
```

- **Light up the LED**

```
root@myir:/sys/class/leds# echo 0 > /sys/class/leds/blue/brightness
```

- **Enable LED trigger mode**

After the heartbeat mode is enabled, the LED flashes at 1Hz by default with a duty cycle of 50% :

```
[root@myir:/]# echo heartbeat > /sys/class/leds/blue/trigger
```

3.3. Key

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

1) Device tree configuration information

Open the supporting device tree file myir-YT507H.dtsi, you can see the node with S2 Poweroff&on button:

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

2) Key Test

- **View the input event information**

```
[root@myir:/]# 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=event0
```

```
B: PROP=0
B: EV=3
B: KEY=100000000800 c000000000000 10000000

I: Bus=0000 Vendor=0000 Product=0000 Version=0000
N: Name="axp2101-pek"
P: Phys=m1kbd/input2
S: Sysfs=/devices/platform/soc/twi5/i2c-5/5-0036/axp2101-pek.0/input/input1
U: Uniq=
H: Handlers=event1
B: PROP=0
B: EV=100003
B: KEY=10000000000000 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=event2
B: PROP=0
B: EV=11
B: MSC=10

I: Bus=0019 Vendor=0001 Product=0001 Version=0100
N: Name="sunxi-gpadc1"
P: Phys=sunxigpadc1/input0
S: Sysfs=/devices/virtual/input/input3
U: Uniq=
H: Handlers=event3
B: PROP=0
B: EV=11
B: MSC=10
```

```
I: Bus=0019 Vendor=0001 Product=0001 Version=0100
N: Name="sunxi-gpadc2"
P: Phys=sunxigpadc2/input0
S: Sysfs=/devices/virtual/input/input4
U: Uniq=
H: Handlers=event4
B: PROP=0
B: EV=11
B: MSC=10

I: Bus=0019 Vendor=0001 Product=0001 Version=0100
N: Name="sunxi-gpadc3"
P: Phys=sunxigpadc3/input0
S: Sysfs=/devices/virtual/input/input5
U: Uniq=
H: Handlers=event5
B: PROP=0
B: EV=11
B: MSC=10

I: Bus=0019 Vendor=0001 Product=0001 Version=0100
N: Name="sunxi-ir"
P: Phys=sunxi-ir/input0
S: Sysfs=/devices/platform/soc/7040000.s_cir/rc/rc0/input6
U: Uniq=
H: Handlers=event6
B: PROP=0
B: EV=100013
B: KEY=2
B: MSC=10
```

The device event corresponding to GPIO-keys is event1.

- **Evtest Tests key information**

Run the following command to press S2. The serial port terminal displays the following information:

```
[root@myir:/]# evtest
No device specified, trying to scan all of /dev/input/event*
Available devices:
/dev/input/event0:        sunxi-keyboard
/dev/input/event1:        axp2101-pek
/dev/input/event2:        sunxi-gpadc0
/dev/input/event3:        sunxi-gpadc1
/dev/input/event4:        sunxi-gpadc2
/dev/input/event5:        sunxi-gpadc3
/dev/input/event6:        sunxi-ir
Select the device event number [0-6]: 1
Input Driver Version is 1.0.1
Input device ID: bus 0x0 vendor 0x0 product 0x0 version 0x0
Input device name: "axp2101-pek"
Supported events:
  Event type 0 (EV_SYN)
  Event type 1 (EV_KEY)
    Event code 116 (KEY_POWER)
Key repeat handling:
  Repeat type 20 (EV_REP)
    Repeat code 0 (REP_DELAY)
      Value      250
    Repeat code 1 (REP_PERIOD)
      Value       33
Properties:
Testing ... (interrupt to exit)
Event: time 1930.466183, Type 1 (EV_KEY), code 116 (KEY_POWER), value 1
Event: time, 1930.466183 -- -- -- -- -- -- -- -- -- -- -- -- -- -- SYN_REPORT -
- -- -- -- -- -- -- -- -- -- -- --
Event: Time 1930.632878, Type 1 (EV_KEY), code 116 (KEY_POWER), value 0
```

```
Event: time, 1930.632878 -- -- -- -- -- -- -- -- -- -- -- -- -- SYN_REPORT -
- -- -- -- -- -- -- -- -- -- --
Event: time 1932.242150, Type 1 (EV_KEY), Code 116 (KEY_POWER), value 1
Event: time, 1932.242150 -- -- -- -- -- -- -- -- -- -- -- -- -- SYN_REPORT -
- -- -- -- -- -- -- -- -- -- --
Event: Time 1932.449797, Type 1 (EV_KEY), Code 116 (KEY_POWER), value 0
Event: time, 1932.449797 -- -- -- -- -- -- -- -- -- -- -- -- -- SYN_REPORT -
- -- -- -- -- -- -- -- -- -- --
Event: time 1933.384171, Type 1 (EV_KEY), code 116 (KEY_POWER), value 1
Event: time, 1933.384171 -- -- -- -- -- -- -- -- -- -- -- -- -- SYN_REPORT -
- -- -- -- -- -- -- -- -- -- --
Event: time 1933.562458, Type 1 (EV_KEY), code 116 (KEY_POWER), value 0
Event: time, 1933.562458 -- -- -- -- -- -- -- -- -- -- -- -- -- SYN_REPORT -
- -- -- -- -- -- -- -- -- -- --
```

Every time you press S2, the current terminal will print the current event code value, that is, the key is normal.

3.4. USB

This section describes how to verify the feasibility of the USB Host driver by using related commands, hot swap, and USB HUB to implement the USB flash drive reading and writing functions and USB enumeration functions.

1) View the printed messages inserted into the USB, anyway

- **View USB device information**

Connect the USB disk to the DEVELOPMENT board USB Host interface (J7), and the kernel prompt is as follows:

```
[3.624780] SCSI 0:0:0:0: direct-access General UDisk 5.00 PQ: 0 ANSI: 2
[3.626654] SD 0:0:0:0: [sda] 122880000 512-Byte Logical Blocks: (62.9GB / 58.6 gib)
[3.627373] SD 0:0:0:0: [sda] Write Protect is off
[3.627381] SD 0:0:0:0: [sDA] Mode Sense: 0B 00 00 08
[3.628101] SD 0:0:0:0: [sda] No Caching mode page found
[3.628107] SD 0:0:0:0: [sda] Assuming Drive Cache: Write through
[3.635124] sda: sda1
[3.638741] SD 0:0:0:0: [sda] Attached SCSI Removable Disk
[3.678742] IR NEC Protocol Handler initialized
[3.683880] IR RC5(X/SZ) Protocol Handler Initialized
[3.690272] sunxi_IR_startup: Get IR protocol failed
[3.695749] sCIr supply ir0 not found, using dummy regulator
[3.703386] Registered IR keymap RC_MAP_SUNXI
As [3.708760] input: sunxi - ir/devices/platform / / 7040000. The soc s_cir/rc/rc0 / input4
As [3.717408] rc rc0: sunxi - ir/devices/platform / / 7040000. The soc s_cir/rc/rc0
[3.727294] USBCore: Registered New Interface Driver uvcVideo
[3.733799] USB Video Class Driver (1.1.1)
[3.738432] Cedar Version 0.1
[3.742562] VE: Install start!!
```

According to the preceding information, the device to be mounted is SDA1.

2) Usb disk mount read and write

- **Mount the U disk**

```
root@myir:~#mount /dev/sda1 /mnt/
```

- **Read the file**

You need to create a test. TXT file on the USB flash drive in advance.

```
root@myir:~# ls /mnttest.txt
root@myir:~# cat /mnt/test.txt
helloworld!
```

- **Write files**

```
root@myir:~# touch test.txt
Root @ myir: ~ # echo "helloworld!!!!!!!!!" > test.txt
root@myir:~# cp test.txt /mnt
root@myir:~# cat /mnt/test.txt
helloworld!!!
```

After writing files, you need to run the sync command to ensure that data is fully written into the USB flash drive before uninstalling the USB flash drive.

3) Uninstall the U disk

- **Unloading operations**

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

3.5. 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 microSD card is smaller in appearance and the smallest in size.Although the size and interface shape of the microSD card are different from those of the original SD card, the interface specification remains unchanged to ensure compatibility.If the Micro SD is inserted into a specific conversion card, it can be used as a standard SD card. SD card has become the most widely used memory card in current consumer digital equipment, with large capacity, high performance, security and other features of multi-functional memory card.The microsd card has nine pins on the back, including four data cables, and supports 1bit or 4bit data transmission widths. MYD-YT507H supports 3-channel 8bit SDMMC interface. SDMMC1 is used to connect microSD on the starter board. The hardware specifications of this interface are as follows:

Support 1bit/4bit SDMMC interface,Supports SDHC Class 10 MicroSD card supports the first generation UHS bus interface (UHS-1 speed Class U3), and does not support UHS-II.The maximum transmission speed (theoretical value) of UHS-I is 104MB/s.The letter I indicates that the device (SD card or card reader) supports uHS-I interfaces.The letter U contains 3, indicating that the read/write speed of the device is U3. supports SDHC card (>2GB to 32GB), SDXC card (>32GB to 2TB). This section explains how to view and operate a TF card in Linux.

1) View the TF card capacity

You can run the fdisk-l command to query the partition information and capacity of the TF card:

```
[root@myir:/]# fdisk -l
slightly
GPT PMBR size mismatch (3145727 != 7806975) will be corrected by write.
Disk /dev/mmcblk1: 3.74gib, 3997171712 bytes, 7806976 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
```

```
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: ED5A32D1-2945-4300-A5AF-5DD822C77BC1
Device                Start     End     Sectors  Size  Type
/dev/mmcblk1p1        34        545     512      256K  Linux reserved
/dev/mmcblk1p2        546       1057    512      256K  Linux reserved
/dev/mmcblk1p3        1058      5153    4096     2M    Linux reserved
/dev/mmcblk1p4        5154      46113   40960    20M   Linux filesystem
/dev/mmcblk1p5        46114     66593   20480    10M   Linux filesystem
/dev/mmcblk1p6 66594 2539617 2473024 1.2g Linux filesystem
/dev/mmcblk1p7        2539618   3145694 606077   296M  Linux filesystem
```

2) View partition information of the TF card

You can run the df command to query information about eMMC partitions, usage, and mount directories.

```
[root@myir:/]# df -h
Filesystem                Size      Used Available Use% Mounted on
/dev/mmcblk1p4            1.9G    475.9M      1.5G  24% /
tmpfs                   993.4M     52.0K    993.3M   0% /tmp
tmpfs                   993.4M     20.0K    993.3M   0% /run
devtmpfs                984.0M         0    984.0M   0% /dev
tmpfs                   993.4M         0    993.4M   0% /dev/shm
```

- /dev/mmcblk1p4: the root file system, mount to the root directory
- TMPFS: memory virtual file system that mounts to a different directory
- Devtmpfs: used to create dev for the system

3) TF card performance test

The performance test mainly tests the speed of eMMC reading and writing files in Linux. It is generally combined with the time and DD commands.Mount the TF card partition to be tested. In this example, the last partition /dev/mmcblk1p7 is mounted to the /usr/local directory.

- **Write file test**

```
[root@myir:/mnt/lost+found]# time dd if=/dev/zero of=test_file_w bs=1M co
unt=500 conv=fsync
500+0 records in
500+0 records out
The real 0 m18. 316 s
The user 0 m0. The 010 s
Sys 0 m3. The 527 s
```

The disk write speed tested here is 27.29m/s.

- **Read file test**

```
[root@myir:/mnt/lost+found]# time dd if=test_file_w of=test_file_r bs=1M c
ount=500
49.43 m/s
500+0 records in
500+0 records out

The real 0 m10. 115 s
The user 0 m0. The 007 s
Sys 0 m4. The 053 s
```

The speed of reading data directly from the SD card is 49.43m/s.

3.6. ADC

GPADC is an analog-to-digital conversion module with 12bit sampling accuracy, supporting four channels, and the specific range of analog input is determined by the platform (T507 platform is 1.8V).ADC tests are implemented through the sysFS interface of the file system. The following uses ADC1 channel 0 as an example.J28 port GPADC0 pin.

1) Can make the ADC

```
[root@myir:/]# echo gpadc0,1 > /sys/class/gpadc/status
```

2) Read the ADC's SYSFS interface

Use the following command to view the ADC read interface:

```
[root@myir:/]# evtest
No device specified, trying to scan all of /dev/input/event*
Available devices:
/dev/input/event0:        sunxi-keyboard
/dev/input/event1:        axp2101-pek
/dev/input/event2:        sunxi-gpadc0
/dev/input/event3:        sunxi-gpadc1
/dev/input/event4:        sunxi-gpadc2
/dev/input/event5:        sunxi-gpadc3
/dev/input/event6:        sunxi-ir
Select the device event number [0-6]:
```

The EVENT2-6 interface can correspond to the TEST ADC1-5.

3) Read the ADC test value

Here we take reading full value as an example, first we select GPADC0 channel, input full value corresponding voltage 1.8V in the pin, check the conversion value:

```
[root@myir:/]# evtest
No device specified, trying to scan all of /dev/input/event*
```

```
Available devices:
/dev/input/event0:        sunxi-keyboard
/dev/input/event1:        axp2101-pek
/dev/input/event2:        sunxi-gpadc0
/dev/input/event3:        sunxi-gpadc1
/dev/input/event4:        sunxi-gpadc2
/dev/input/event5:        sunxi-gpadc3
/dev/input/event6:        sunxi-ir
Select the device event number [0-6]: 2
Input Driver Version is 1.0.1
Input device ID: bus 0x19 vendor 0x1 product 0x1 version 0x100
Input device name: "sunxi-gpadc0"
Supported events:
  Event type 0 (EV_SYN)
  Event type 4 (EV_MSC)
    Event code 4 (MSC_SCAN)
Properties:
Testing ... (interrupt to exit)
Event: time, 753.920019 -- -- -- -- -- -- -- -- -- -- -- -- -- -- SYN_REPORT --
 -- -- -- -- -- -- -- -- -- -- --
Event: time 753.921021, Type 4 (EV_MSC), Code 4 (MSC_SCAN), value fec
Event: time, 753.921021 -- -- -- -- -- -- -- -- -- -- -- -- -- -- SYN_REPORT --
 -- -- -- -- -- -- -- -- -- -- --
Event: Time 753.922018, Type 4 (EV_MSC), Code 4 (MSC_SCAN), value fe3
Event: time, 753.922018 -- -- -- -- -- -- -- -- -- -- -- -- -- -- SYN_REPORT --
 -- -- -- -- -- -- -- -- -- -- --
Event: time 753.923018, Type 4 (EV_MSC), Code 4 (MSC_SCAN), value fe0
Event: time, 753.923018 -- -- -- -- -- -- -- -- -- -- -- -- -- -- SYN_REPORT --
 -- -- -- -- -- -- -- -- -- -- --
Event: Time 753.924018, Type 4 (EV_MSC), Code 4 (MSC_SCAN), Value Feb
Event: time, 753.924018 -- -- -- -- -- -- -- -- -- -- -- -- -- -- SYN_REPORT --
 -- -- -- -- -- -- -- -- -- -- --
Event: time 753.925018, Type 4 (EV_MSC), Code 4 (MSC_SCAN), value fe2
```

The conversion value of the ADC0 path is read continuously, and the output is a hexadecimal value.Take value fec as an example to test the decimal value 4076.Plug into the formula (4076/4096) *1.8 and the test value is 1.79V.

The ADC of MYD-T507 is 12 bits, and the maximum access voltage is 1.8V. Under the full range, the theoretically read value is 4096. There is some error in the current read value, which belongs to a reasonable range.

3.7. Display

This module consists of display engine (DE) and each type of controller (TCON).After inputting layers for display related processing in DE, the layers are output to display device through one or more interfaces, so as to achieve the function of compositing multiple layers of application rendering and presenting them to users on display.DE has two independent units (can be referred to as DE0, DE1), which can respectively accept the user input layers for synthesis, output to different displays, to achieve double display.Each individual cell of DE has 1-4 channels (typically, DE0 has 4 and DE1 has 2), and each channel can handle four layers of the same format at the same time.Sunxi platform has video channel and UI channel.The video channel is powerful and supports YUV format and RGB layers.The UI channel only supports RGB layers.In brief, the main functions of the display module are as follows:

- Supports LCD (HV/LVDS/CPU/DSI) output
- Supports dual-display output
- Support multi-layer overlay mixed processing
- Support multiple display effect processing (alpha, Colorkey, image enhancement, brightness/contrast/saturation/chroma adjustment)
- Supports intelligent backlight adjustment
- Support multiple image data format input (ARGB, YUV)
- Supports image zooming • Supports screen capture • Supports image conversion

1) Device tree configuration information

Open the supporting device tree file myir-YT507H.dtSI, you can see a variety of display scheme combinations.

```
/*display*/
#include "display/myir-hdmi-1920x1080-1lvds-7-1024x600.dtsi"
//#include "display/myir-lcd-1lvds-7-1024-600.dtsi"
/ / # include "display/myir - - an LVDS LCD - 10.1-1280-800. The dtsi"
//#include "display/myir-lcd-2lvds-7-1024-600.dtsi"
//#include "display/myir-lcd-2lvds-21-1920-1080.dtsi"
```

```
//#include "display/myir-hdmi.dtsi"
//#include "display/myir-tv.dtsi"
```

2) Displays solution combination

- **default display**

MYD-YT507H uses HDMI and LVDS0 display combination by default.

At this point, no matter which device is connected, the display can be normal. If the display device is connected at the same time, switch the display device and perform the following operations:

```
[root@myir:/]# mount -t debugfs none /sys/kernel/debug/
[root@myir:/]# cd /sys/kernel/debug/dispdbg/
[root@myir:/sys/kernel/debug/dispdbg]# echo disp0 > name
[root@myir:/sys/kernel/debug/dispdbg]# echo switch > command
[root@myir:/sys/kernel/debug/dispdbg]# echo 1 0 > param
[root@myir:/sys/kernel/debug/dispdbg]# echo 1 > start
```

  - Name: DISP0/1/2 -- indicates that channel 0/1/2 is displayed
  - Command: switch -- Runs the switch command
  - param: type mode -- type: 0(none),1(lcd),2(tv),4(hdmi),8(vga)
  - Start: Enter 1 to run the command

- **21 "dual LVDS display**

dtsi comments out other display schemes in device tree myir-YT507H.dtsi before compilation, and uncomments of the following schemes can be compiled to generate dual-channel LVDS images.

```
#include "display/myir-lcd-2lvds-7-1024-600.dtsi"
```

- **LVDS0 + LVDS1 display**

dtsi comments out other display schemes in device tree myir-YT507H.dtsi before compilation, and uncomments of the following schemes can be compiled to generate dual-channel LVDS images.

```
#include "display/myir-lcd-2lvds-7-1024-600.dtsi"
```

J12 and J13 interfaces are respectively connected to LVDS0 and LVDS1 screens. In this case, only LVDS0 screen screen will be displayed. The following operations are required to display LVDS1 screen asynchronously:

```
[root@myir:/]# mount -t debugfs none /sys/kernel/debug/
[root@myir:/]# cd /sys/kernel/debug/dispdbg/
[root@myir:/sys/kernel/debug/dispdbg]# echo disp0 > name
[root@myir:/sys/kernel/debug/dispdbg]# echo switch > command
[root@myir:/sys/kernel/debug/dispdbg]# echo 1 0 > param
[root@myir:/sys/kernel/debug/dispdbg]# echo 1 > start
```

- **CVBS display**

After connecting the J30 port to the CVBS device, you need to perform the following operations to display the J30 port properly:

```
cat /sys/class/disp/disp/attr/sys
mount -t debugfs none /sys/kernel/debug
cd /sys/kernel/debug/dispdbg/
echo disp1 > name
echo switch > command
echo 2 1 > param
echo 1 > start
echo 1 > /sys/class/disp/disp/attr/disp
echo 8 > /sys/class/disp/disp/attr/colorbar
```

3.8. Touch Panel

There are capacitive touch and resistance touch. The hardware of MYD-YT507H development board does not support resistance touch at present, but capacitive touch is supported.According to the actual needs to buy their own accessories.Capacitive screen in use is more sensitive, few problems.In addition, the capacitance screen does not need to be accurate. Because according to the principle of capacitive screen, capacitive screen can accurately identify the position of finger and screen contact in use, with high sensitivity.If we click on the software in use, there is generally only one case: there is a problem with the screen.Here is a simple test to test the touch function of the capacitive screen using the evtest command

1) Touch screen connection

Connect the MY-LVDS070C_V1.2 LVDS screen to the development board as per section 3.7.

2) Evtest Tests the evtest command

The terminal runs evtest to go to the test page.Select the test peripheral as touch screen, in which imer thinks that input interrupt 0, select "0" on the test interface and press Enter to start the test:

```
[root@myir:/]# evtest
No device specified, trying to scan all of /dev/input/event*
Available devices:
/dev/input/event0:      sunxi-keyboard
/dev/input/event1:      axp2101-pek
/dev/input/event2:      sunxi-gpadc0
/dev/input/event3:      sunxi-gpadc1
/dev/input/event4:      sunxi-gpadc2
/dev/input/event5:      sunxi-gpadc3
/dev/input/event6:      sunxi-ir
/dev/input/event7:      ft5x_ts
```

```
Select the device event number [0-7]: 7
Input Driver Version is 1.0.1
Input device ID: bus 0x0 vendor 0x0 product 0x0 version 0x0
Input device name: "ft5x_ts"
Supported events:
  Event type 0 (EV_SYN)
  Event type 1 (EV_KEY)
    Event code 330 (BTN_TOUCH)
  Event type 3 (EV_ABS)
    Event code 48 (ABS_MT_TOUCH_MAJOR)
      Value      0
      Min        0
      Max      255
    Event code 50 (ABS_MT_WIDTH_MAJOR)
      Value      0
      Min        0
      Max      200
    Event code 53 (ABS_MT_POSITION_X)
      Value      0
      Min        0
      Max     1024
    Event code 54 (ABS_MT_POSITION_Y)
      Value      0
      Min        0
      Max      600
    Event code 57 (ABS_MT_TRACKING_ID)
      Value      0
      Min        0
      Max        4
Properties:
  Property type 1 (INPUT_PROP_DIRECT)
Testing ... (interrupt to exit)
Event: Time 467.761457, Type 1 (EV_KEY), Code 330 (BTN_TOUCH), value 1
```

```
Event: time 467.761457, Type 3 (EV_ABS), code 57 (ABS_MT_TRACKING_ID), val
ue 0
Event: Time 467.761457, Type 3 (EV_ABS), code 48 (ABS_MT_TOUCH_MAJOR),
value 20
Event: Time 467.761457, Type 3 (EV_ABS), code 53 (ABS_MT_POSITION_X), val
ue 81
Event: Time 467.761457, Type 3 (EV_ABS), code 54 (ABS_MT_POSITION_Y), valu
e 21
Event: Time 467.761457, Type 3 (EV_ABS), code 50 (ABS_MT_WIDTH_MAJOR),
value 30
Event: the time 467.761457, + + + + + + + + + + + + + + + SYN_MT_REPOR
T + + + + + + + + + + + +
Event: time, 467.761457 -- -- -- -- -- -- -- -- -- -- -- -- -- SYN_REPORT --
 -- -- -- -- -- -- -- -- -- -- --
Event: time 467.781182, Type 3 (EV_ABS), code 57 (ABS_MT_TRACKING_ID), val
ue 0
Event: Time 467.781182, Type 3 (EV_ABS), code 48 (ABS_MT_TOUCH_MAJOR),
value 20
Event: Time 467.781182, Type 3 (EV_ABS), code 53 (ABS_MT_POSITION_X), val
ue 80
Event: Time 467.781182, Type 3 (EV_ABS), code 54 (ABS_MT_POSITION_Y), valu
e 17
Event: Time 467.781182, Type 3 (EV_ABS), code 50 (ABS_MT_WIDTH_MAJOR),
value 30
Event: the time 467.781182, + + + + + + + + + + + + + + + SYN_MT_REPOR
T + + + + + + + + + + + +
Event: time, 467.781182 -- -- -- -- -- -- -- -- -- -- -- -- -- SYN_REPORT --
 -- -- -- -- -- -- -- -- -- -- --
```

3.9. Ethernet

There are many network configuration tools in Linux, such as Net-tools, Iproute2, Systemd-Networkd, Network Manager and Connman, etc., which can be selected according to actual needs during system customization.The MYD-YT507H has two network ports: eth0 and eth1.This section uses eth0 as an example to describe common Ethernet configurations.

1) Configure an Ethernet IP address

- **Use ifconfig in the Net-Tools tool package to manually configure the network**

Run the ifconfig command to check the network device information as follows:

```
[root@myir:/sys/kernel/debug/dispdbg]# ifconfig
eth0      Link encap:Ethernet  HWaddr 36:C9:E3:F1:B8:05
          Inet addr: 192.168.1.98 Bcast: 192.168.1.255 Mask: 255.255.255.0
          inet6 addr: fe80::dd12:1815:4621:b249/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:16773 errors:0 dropped:0 overruns:0 frame:0
          TX packets:306 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1367859 (1.3 MiB) TX bytes:24861 (24.2 KiB)
          Interrupt:65

eth1      Link encap:Ethernet  HWaddr 36:C9:E3:F1:B8:05
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0b) TX bytes:780 (780.0b)
          Interrupt:66

lo        Link encap:Local Loopback
          Inet addr: 127.0.0.1 Mask: 255.0.0.0
```

```
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:0 (0.0b) TX bytes:0 (0.0b)
```

To manually configure the IP address 192.168.0.100 for eth0, run the following command:

```
[root@myir:/] # ifconfig eth0 192.168.1.100 netmask 255.255.255.0 Up
```

Run the preceding command to manually set the IP address of eth0 to 192.168.0.100, subnet mask to 255.255.255.0, and default broadcast address 192.168.0.255, and activate the IP address with the up parameter, as shown in the following figure:

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

- **Manually configure the network using the IP command in the Iproute2 toolkit**

The ifconfig command can also be used to manually set the IP address using IP addr and IP link. More information please view the instructions in the https://wiki.linuxfoundation.org/networking/iproute2.

```
[root@myir:/]# ip addr flush dev eth0
[root@myir:/]# IP addr add 192.168.0.101/24 BRD + dev eth0
[root@myir:/]# ip link set eth0 up
```

If the IP address has been configured before, the IP address configured with IP addr add will become the Second ary address. Therefore, use IP Addr flush to flush the previous IP address before configuring and activating the IP address.After the configuration, run the IP addr show command to view the following information:

```
[root@myir:/]#  ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast st
ate UP group default qlen 1000
    link/ether 00:0c:29:36:97:20 brd ff:ff:ff:ff:ff:ff
    Inet 192.168.1.195/24 BRD 192.168.1.255 Scope Global noprefixroute eth0
       valid_lft forever preferred_lft forever
    Inet 192.168.1.101/24 BRD 192.168.1.255 Scope Global Secondary eth0
       valid_lft forever preferred_lft forever
```

2) Changing a Mac Address

To manually change the Mac address 00:0C:29:36:97:20, run the following command:

```
[root@myir:/]# ifconfig eth0 down
[root@myir:/]# ifconfig eth0 hw ether 00:0C:29:36:97:20
[root@myir:/]# ifconfig eth0 up
[root@myir:/]# ifconfig eth0
eth0        Link encap:Ethernet  HWaddr 00:0C:29:36:97:20
            Inet addr: 192.168.1.197 Bcast: 192.168.1.255 Mask: 255.255.255.0
            inet6 addr: fe80::dd12:1815:4621:b249/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:17112 errors:0 dropped:0 overruns:0 frame:0
            TX packets:338 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:1395663 (1.3 MiB) TX bytes:27773 (27.1 KiB)
            Interrupt:65
```

4. Extended peripheral interface

MYD-YT507H development board provides a wealth of peripheral interfaces, in addition to the basic peripheral interface, but also can be connected to various external expansion modules.The development of users is more flexible and convenient.Here are the test steps for several optional modules introduced by MYIR.Users buy according to their needs.For details about optional modules, see Table 1-2 Optional modules.

4.1. MY - WiredCom module

My-wiredcom module is the raspberry PI peripheral interface form launched by Mir, including RS232/RS485/CAN/ SPI/I2C peripheral interfaces.You need to purchase this module by yourself. For details about this module, see Table 1-2 Optional modules.Before testing, users need to connect the module to the development board J25 interface.

1) RS232 test

This section uses the Linux API to configure the transceiver function of the DEVELOPMENT board RS232.Linux serial device files are usually named /dev/ttysn (n=0,1,2,3.....).N indicates the serial port device number in Linux. ttyS is the serial port device name defined by the kernel.This section uses the J2 interface on the My-Wiredcom expansion board as an example. The numbered device node of the J2 interface is ttyS2.The test configuration is as follows:

Table 4-1. RS232 interface configuration

intreface	MYD-YT507H	Windows 10
Hardware interface	RS232	USB-RS232 module
Device node	ttyS2	com12
The test software	uart_test	sscom

Here, RXD and TXD of My-Wiredcom module J2 are connected with RXD and TXD of USB-RS232 converter respectively.

- **Test development board RS232 transceiver data**

First, configure the serial port tool sscom in Windows to send the string periodically, for example, ni hao at an interval of 1000ms.

When the development board is ready to receive data, execute the following command on the development board to receive data.

After the command is executed, the interrupt enters a blocking state, waiting to receive data from the serial port of the computer. When receiving data from the serial port of the computer, the interrupt sends back the string "Hello world" and prints the received string.

```
[root@myir:/]# ./eth/uart/uart.out /dev/ttyS2
total_send is 11
my total len = 6
my data: ni hao
my total len = 12
my data: ni hao
my total len = 18
my data:ni hao
my total len = 24
my data:ni hao
```

In Windows, sSCOM.exe is displayed

```
[22:46:53. 902] closed to derive hello world
[22:46:54. 117] - > filled ni hao
[22:46:54. 415] - > filled ni hao
[22:46:54. 482] - > filled ni hao
[22:46:54. 661] - > filled ni hao
[22:46:54. 850] - > filled ni hao
```

Data sent and received on the development board corresponds to data sent and received by sSCOM.exe under Windows, that is, data received by RS232 on the development board is normal.

2) RS485 test

This example demonstrates how to use Linux API test development board RS485 send and receive data function, device node is ttyS5.The RS485 port on the J2 module is used as an example for the test. The hardware port configurations are as follows:

Table 4-2. RS485 port configuration

inerface	MYD-YT507H	Windows 10
Hardware interface	RS485	USB-RS485 module
Device node	ttyS5	com12
The test software	uart_test	sscom

Here, 485A and 485B of J2 module of MY-Wiredcom are connected with 485A and 485B of USB-RS485 converter respectively.

- **Test development board RS485 transceiver data**

First, configure the serial port tool sscom in Windows to send the string periodically, for example, ni hao at an interval of 1000ms.

When the development board is ready to receive data, execute the following command on the development board to receive data.

After the command is executed, the interrupt enters a blocking state, waiting to receive data from the serial port of the computer. When receiving data from the serial port of the computer, the interrupt sends back the string "Hello world" and prints the received string.

```
[root@myir:/]# /uart.out /dev/ttyS5
total_send is 11
my total len = 6
my data: ni hao
my total len = 12
my data: ni hao
my total len = 18
my data:ni hao
my total len = 24
my data:ni hao
```

In Windows, sSCOM.exe is displayed.

```
[22:46:53. 902] closed to derive hello world
[22:46:54. 117] - > filled ni hao
[22:46:54. 415] - > filled ni hao
[22:46:54. 482] - > filled ni hao
[22:46:54. 661] - > filled ni hao
[22:46:54. 850] - > filled ni hao
```

The data sent and received on the development board corresponds to the data sent and received by sSCOM.exe under Windows, that is, the data received by RS485 on the development board is normal.

4.2. MY - WF005S module

WIFI/BT module MY-WF005S is a Wi-Fi and Bluetooth two-in-one module introduced by MYIR. The chip scheme is AP6212. Users need to purchase this module according to their needs.For details about modules, see Table 1-1 Optional modules.Before testing, users need to connect the module to the development board J26 interface.

1) Wi-fi test

This topic describes how to configure and use Wi-Fi in Linux. Generally, Wi-Fi modules can work in STA mode and AP mode. Some devices can work in STA mode and AP mode simultaneously.The STA mode allows the device to connect to an external Wi-Fi hotspot. The AP mode turns the device into a Wi-Fi hotspot for other devices to connect to.

MYD-YT507H can connect to MYIR's AP6212 Wi-Fi and Bluetooth two-in-one module. Currently, it does not support STA and AP to work at the same time. The corresponding driver of AP6212 Wi-Fi module is:

```
[root@myir:/]# lsmod
Module                  Size  Used by    Tainted: G
bcmdhd                1269760  0
mali_kbase             520192  3
```

During driver loading, wi-fi firmware in /lib/firmware/brCM is loaded into the module.After the Wi-Fi module driver is loaded successfully, the wi-fi device network node Wwan0 is generated, as shown below:

```
[root@myir:/]# ifconfig wwan0
wwan0     Link encap:Ethernet  HWaddr 5A:38:06:A6:8C:60
          Inet addr: 192.168.30.1  Bcast: 192.168.30.255  Mask: 255.255.255.0
          inet6 addr: fe80::7573:713b:e50d:48e7/64 Scope:Link
          UP BROADCAST RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:61 errors:0 dropped:0 overruns:0 carrier:0
```

```
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0b) TX bytes:18920 (18.4kib)
```

2) STA mode script connects to WiFi hotspot

Let's try manually connecting to a nearby Wi-Fi hotspot "myir006", a WPA2-encrypted Wi-Fi hotspot with password 12345678.

Ensures that the WLAN0 network device is activated.

```
root@myir:~# ifconfig wlan0 up
```

- **Scan for nearby WiFi hotspots**

Scan nearby wifi hotspots and get the list of nearby Wi-Fi hotspots as follows:

```
[root@myir:/]#  iw dev wlan0 scan | grep SSID
[10173.935576] wl_RUN_escan: LEGACY_SCAN Sync ID: 1, bsSIDx: 0
          SSID: Myir-caigou_Wi-Fi5
          SSID: myir_hys_2. 4 g
          SSID: Myir-caigou
          SSID: test456
          SSID: DYX-01
          SSID: Myir-hys
          SSID: myir006
          SSID: DYX-02
          SSID: MERCURY_376F
          SSID: myir006
          SSID: MYIR_ROY
          SSID: DYX-03
          SSID: DYX-04
          SSID: ChinaNet-6kDy
```

- **Wpa_passphrase Sets the wifi name and password**

```
[root@myir:/usr/lib/ltp-testsuite/network]# wpa_passphrase SSID passwd >> /etc/wpa_supplicant. conf
[root@myir:/usr/lib/ltp-testsuite/network]# cat /etc/wpa_supplicant.conf
ctrl_interface=/var/run/wpa_supplicant
```

```
ctrl_interface_group=0
ap_scan=1
update_config=1

network={
        ssid="myir006"
   psk="12345678"
   key_mgmt=WPA-PSK
#  key_mgmt=NONE
}
```

Generate a WPA PSK from the ASCII password of one SSID for encryption operation.

- **Shut down the wpa_supplicant process**

Before connecting to wPA_supplicant and configuring WIFI, shut down the WPA_supplicant process:

```
Root @ myir: ~ # killall wpa_supplicant
```

- **Initialize wpa_supplican**

Wpa_supplicant is a tool for connecting and configuring WIFI. Its main job is to communicate with drives through sockets and report data to the user layer. The user layer can also send commands to the WPA_supplicant through the socket to activate the driver to operate on the WiFi chip.It usually runs in the background as follows:

```
root@myir:~# wpa_supplicant -B -Dnl80211 -c /etc/wpa_supplicant.conf -i wla
n0
```

- -b: runs the daemon process in the background
- -d: drives the name
- -c: indicates the configuration information path
- -i: indicates the listening wifi interface

- **Obtaining an IP Address**

After the configuration is complete, run the WiFi obtaining script /etc/test/wifi-on.sh to connect to WiFi, as shown in the following:

```
[root@myir:/]# ./etc/test/wifi-on.sh
[549.867485] DHd_module_init: In Dongle Host Driver, Version 1.579.77.41.11
(R)
[549.875661] = = = = = = = = dhd_wlan_init_plat_data = = = = = = = =
[549.881401] dhD_Wlan_INIT_GPIO: WL_HOST_WAKE=-1, oob_IRQ =127, oob_IR
q_FLAGS =0x4084
[549.889960] dhd_wlan_init_gpio: WL_REG_ON = 1
[549.894795] dhd_wifi_platform_load: Enter
[549.899366] Power-up Adapter 'DHD Generic Adapter '
.
  [555.783061] Connecting: E :ec: C3: D3: C2 SSID "myIR006 ", len (7) channel
=11
[555.783061]
[555.795625] dhd_dbg_start_pkt_monitor, 1724
[555.900283] wl_bSS_CONNECt_done Succeeded with 48:0 E: EC: C3 :d3: C2
[555.992310] wl_bSS_CONNECt_done Succeeded with 48:0 E: EC: C3: D3: C2 V
NDR_OUI: 00-90-4C 00-0C-43
[556.244504] IPv6: wlan0: IPv6 duplicate address fe80: : 36 e5:57 f0:5 ceb: 9 a
ee detected!
deleting routers
Adding DNS 192.168.1.1
Adding DNS 192.168.30.1
```

- **Ping Baidu to check whether the connection is normal**

```
[root@myir:/]# ping www.baidu.com
PING www.a.shifen.com (14.215.177.38) 56(84) bytes of data.
64 bytes from 14.215.177.38 (14.215.177.38): ICmp_seq =1 TTL =55 time= 10.
5ms
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=2 TTL =55 time= 14.6
ms
```

```
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=3 TTL =55 time=25.4
ms
64 bytes from 14.215.177.38 (14.215.177.38): ICmp_seq =4 TTL =55 time= 10.
5ms
^C
--- www.a.shifen.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 7ms
RTT is the min/avg/Max/mdev 25.357/6.085 = 10.469/15.227 / ms
```

4.3. EC20CEFDKG module

LINUX devices can also connect to an external 4G module for dial-up Internet access. The 4G module is mostly used by the remote EC25.

There are three dialing modes: PPPD, Gobinet and QMI_WWan. PPPD is more common. Here, PPP is used to explain dialing, using EC20 CE FDKG.

1) View the VID and PID

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

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

- 2C7C :0125: VID and PID of the EC25.

Which need to be in the ${KERNEL_DIR} / drivers/usb/serial/option. C arrays in the static const struct usb_device_id option_ids configured in the following:

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

These configurations need to be enabled in the kernel:

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

2) View the kernel identification module

If the kernel adds VID and PID configurations for this module, then the /dev/ttyusb * node is generated:

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

3) Perform preliminary tests using the AT instruction

The AT command can be used to conveniently query signal strength, whether SIM card is inserted, whether the SIM card is currently found by the operator, and also to test the current card function by calling AT. For AT communication, we also need to know which device is the communication port. Here, we need to query the module file. EC20 uses ttyUSB2 for AT communication.Use Microcom as an example, or minicom as an example.For example, if microcom /dev/ttyusb2 enters the mode, CTRL + X exits.

- **Query signal quality**

```
[root@myir:/]# microcom /dev/ttyUSB2
at+csq
31 + CSQ: almost
```

```
OK
```

- 31, 99:31 is the signal quality, the lower the number, the stronger the signal.

- **Check whether the operation can be performed.**

```
at+cpin?
+CPIN: READY

OK
```

- +CPIN:READY :READY means READY.

- **Viewing Carrier Information**

```
at+cops?
+ COPS: 0, 0, "CHN - CT", 100

OK
```

- Chn-ct,100: CHN-CT stands for telecom,100 stands for 2G, 3G, 4G, or 5G according to the module manual.

If all the above three steps work, you can use dial-up Internet access. Here is how to make a phone call and send a text message for one-step verification.

- **Make a phone call**

```
ATD177xxxx5673;
OK
```

- **texting**

```
at+cmgf=1
OK
at+cscs="GSM"
OK
at+cmgs="177xxxx5673"
> hello
+CMGS: 28

OK
```

- At+ CMGF =1: Sets the text message mode
- At+ CSCS = "GSM" : Set TE to use the GSM character
- At+ CMGS: CTRL+Z sends the message after "phone" is written. ECS exits sending

4) PPP Dialing Test

The PPPD dialing command of the development board is used here:

```
[root@myir:/]# pppd call quectel-dial
[root@myir:/]# [63.175640] [DHD] cfg80211-error) wl_cfg80211_netdev_notifier_
call: wdev null.do nothing
[63.184467] [DHD] cfg80211-error) wl_CFg80211_netdev_notifier_call: wdev null
[63.311254] [DHD] cfg80211-error) wl_CFg80211_netdev_notifier_call: wdev null
[63.319869] [DHD] cfg80211-error) wl_CFg80211_netdev_notifier_call: wdev null
Ca [63.640433] [DHD] cfg80211-error) wl_CFg80211_netdev_notifier_call: wdev
null. Do nothing
[DHD] cfg80211-error) wl_CFg80211_netdev_notifier_call: wdev null. Do nothing
```

This dial will take a while, the dial log has been hidden, the user can view the corresponding log:

```
[root@myir:/]# ./etc/test/4G_call.sh
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)
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)
expect (OK)
^M
ATE0^M^M
```

```
OK
 -- got it

send (ATI;+CSUB;+CSQ;+CPIN?;+COPS?;+CGREG?;&D2^M)
expect (OK)
^M
^M
Quectel^M
EC20F^M
Revision: EC20CEFDKGR06A03M2G^M
^M
SubEdition: V05^M
^M
+ CSQ: 3 13 ^ M
^M
+CPIN: READY^M
^M
+ COPS: 0, 0, CHN - CT, 100 ^ M
^M
+ CGREG: 0, 1 ^ M
^M
OK
 -- got it

Send (" IP "AT + CGDCONT = 1," 3 gnet, 0, 0 ^ M)
expect (OK)
^M
^M
OK
 -- got it

send (ATD*99#^M)
expect (CONNECT)
```

```
^M
^M
CONNECT
 -- got it

Script chat -s -v -f /etc/ppp/peers/quectel-chat-connect finished (pid 20232),
status = 0x0
Serial connection established.
using channel 1
Using interface ppp0
Connect: ppp0 <--> /dev/ttyUSB3
rcvd [LCP ConfReq id=0x1 <asyncmap 0x0> <auth chap MD5> <magic 0xad5
4169b> <pcomp> <accomp>]
sent [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0x40057e90> <pcomp>
<accomp>]
sent [LCP ConfAck id=0x1 <asyncmap 0x0> <auth chap MD5> <magic 0xad5
4169b> <pcomp> <accomp>]
rcvd [LCP ConfAck id=0x1 <asyncmap 0x0> <magic 0x40057e90> <pcomp>
<accomp>]
rcvd [CHAP Challenge id=0x1 <6903e995260057a277141ec3faf034e5da33f8b47
e512b54>, name = ""]
sent [CHAP Response id=0x1 <46f56ab573220f1951daa63fda8076d0>, name =
 "test"]
rcvd [CHAP Success id=0x1 ""]
CHAP authentication succeeded
CHAP authentication succeeded
Sent [IPCP ConfReq ID =0x1 <addr 0.0.0.0> < MS-DNS1 0.0.0.0> < Ms-DNS2
 0.0.0.0>]
RCVD [IPCP ConfReq id=0x1 < COMPRESS VJ 0f 00> <addr 192.168.1.1>]
sent [IPCP ConfRej id=0x1 <compress VJ 0f 00>]
RCVD [IPCP ConfNak ID =0x1 <addr 10.88.224.107> < MS-DNS1 219.128.134.
10> < Ms-DNS2 219.128.134.11>]
```

```
Sent [IPCP ConfReq ID =0x2 <addr 10.88.224.107> < MS-DNS1 219.128.134.1
0> < MS-DNS2 219.128.134.11>]
RCVD [IPCP ConfReq ID =0x2 <addr 192.168.1.1>]
Sent [IPCP ConfAck ID =0x2 <addr 192.168.1.1>]
RCVD [IPCP ConfAck id=0x2 <addr 10.88.224.107> < MS-dNS1 219.128.134.1
0> < Ms-dNS2 219.128.134.11>]
Not replacing existing default Route via 192.168.30.1
The local IP address 10.88.224.107
Remote IP address 192.168.1.1
Primary DNS address 219.128.134.10
Secondary DNS address 219.128.134.11
ppp connected!
```

You can see that the connection is normal and the IP address can be obtained.

Run the following command to view the PPP IP address:

```
[root@myir:/]# ifconfig
eth0      Link encap:Ethernet  HWaddr 36:C9:E3:F1:B8:05
          Inet addr: 169.254.127.228 Bcast: 169.254.255.255 Mask: 255.255.0.0
          inet6 addr: fe80::177a:be22:1be1:91e7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2584 errors:0 dropped:0 overruns:0 frame:0
          TX packets:170 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:167436 (163.5 KiB) TX bytes:49664 (48.5 KiB)
          Interrupt:65

eth1      Link encap:Ethernet  HWaddr 36:C9:E3:F1:B8:05
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0b) TX bytes:940 (940.0b)
```

```
          Interrupt:66

lo        Link encap:Local Loopback
          Inet addr: 127.0.0.1 Mask: 255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:0 (0.0b) TX bytes:0 (0.0b)

ppp0      Link encap:Point-to-Point Protocol
          Inet addr: 10.88.224.107 P - t - P: 192.168.1.1 Mask: 255.255.255.25
5
          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:70 (70.0 B) TX bytes:64 (64.0 B)
```

- ppp0: PPP0 is the dial-up network interface card device, and the IP address can be obtained properly.

5. Network applications

By default, the image of the development board contains some common network applications, which is convenient for users to develop or test.

5.1. PING

PING is used to test network connectivity, network latency, and packet loss rate.Once the Ethernet connection is configured, you can use PING for a simple test of the network connection.

1) Wiring and information output

Connect the development board to the switch or router through the CAT6 network cable, and the console will display the connection message output by the kernel, as follows:

```
[root@myir:/]# [9689.127024] sunxi-gmac gmac0 eth0: Link is up-100Mbps/full-flow control rx/tx
```

2) Test the extranet url

```
[root@myir:/]# ping www.baidu.com
PING www.a.shifen.com (14.215.177.39) 56(84) bytes of data.
64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=1 TTL =56 time= 7.09 ms
64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=2 TTL =56 time= 6.86 ms
64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=3 TTL =56 time= 6.34 ms
64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=4 TTL =56 time= 7.08 ms
--- www.a.shifen.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 7ms
RTT min/avg/ Max /mdev = 6.344/6.839/7.094/0.311 ms
```

Note: Ensure that the DNS works properly when ping the public network.

The result shows that the IP address of www.baidu.com after domain name resolution is 14.215.177.39. Icm p_seq indicates the ICMP packet number. If the number is consecutive, no packet is lost.Time represents the delay time for the response, but the shorter the time, the better.In addition to testing Ethernet, the ping command can also be used to test Wi-Fi.

5.2. SSH

SSH is short for Secure Shell. It is formulated by the Network Working Group of the Internet Tf.SSH is a reliable security protocol based on the application layer. It provides security for remote login sessions and other network services.Typically, Linux platforms use DropBear or OpenSSH to implement SSH server and client.Let's test the SSH client and server separately over an Ethernet connection.The client and service program provided by OpenSSH 7.6 P1 (http://www.openssh.com/) are included by default. Configure the connection between the Ethernet interface on the development board and the SSH server. The configured Ethernet card address is as follows:

```
[root@myir:/]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 36:C9:E3:F1:B8:05
          Inet addr: 192.168.1.195 Bcast: 192.168.1.255 Mask: 255.255.255.0
          inet6 addr: fe80::177a:be22:1be1:91e7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:30530 errors:0 dropped:0 overruns:0 frame:0
          TX packets:374 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2446136 (2.3 MiB) TX bytes:26826 (26.1kib)
          Interrupt:65
```

The CURRENT IP address of the SSH server is 192.168.1.195. You can run the ping command to test the connection between the development board and the SSH server.

- **SSH client test**

The development board serves as a client to connect to the SSH server. Run the SSH command on the development board to log in to the server. The command and result are as follows:

```
/ root @ myir: / # SSH zhaoy@192.168.1.13
The authenticity of host '192.168.1.13 (192.168.1.13)' can't be established.
```

```
ECDSA key fingerprint is SHA256:McWDUp/oT8q/smgP3Fo9/ZZBL1gRnjl691CgoqJwo9A.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.13' (ECDSA) to the list of known hosts.
zhaoy@192.168.1.13 's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-100-generic x86_64)

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

105 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

The New release '21.10' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Wed Apr 6 11:20:16 2022 from 192.168.1.229
zhaoy@myir-O-E-M:~$
```

“Zhaoy” is the user name on the server.

After successful login, the console console on the SSH server is automatically accessed, and the user can perform wujL user control on the remote server on the client.To exit, simply execute the "exit" command on the current console.

- **SSH server test**

The development board serves as the SSH server, and other external devices connect to this development board remotely. SSH service is enabled on the development board by default. Therefore, you can also log in to the current development board by using SSH command on other external devices

(development board or PC) with SSH client. The command and result are as follows:

```
Zhaoy @ myir -o - E - M: ~ $SSH to root@192.168.1.195
The authenticity of host '192.168.1.195 (192.168.1.195)' can't be established.
ECDSA key fingerprint is SHA256:jqqKf1IgHNlczHM4J2Cp9lJpGnUQTmBsXI4DIU
VCeII.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.195' (ECDSA) to the list of known hos
ts.
root@192.168.1.195 's password:
COLUMNS=122;LINES=50;export COLUMNS LINES;
[root@myir:~]#
```

In the example above, we remotely log in to the development board as root and access the Console console to perform root user control over the development board.To exit, run the "exit" command on the console. OpenSSH is the main connection tool for remote login using SSH.It encrypts all traffic to eliminate eavesdrop, connection hijacking, and other attacks.In addition, OpenSSH provides a large set of secure tunneling capabilities, multiple authentication methods, and complex and flexible configuration options.You can modify the configuration files ssh_config and sshd_config in the /etc/ssh/ directory on the PC as required. For example, if you want the SSH server to allow the root user to log in remotely without a password, add the following two lines to the /etc/ssh/sshd_config file on the SSH server. PermitRootLogin Yes PermitEmptyPasswords YES The preceding configuration has high security risks and is generally used for remote deployment during debugging.The actual product is generally turned off for safety reasons.

5.3. SCP

SCP is short for Secure Copy. It is a Secure remote file Copy command based on the SSH protocol in Linux. It is very useful in system debugging. We have already introduced the example of remote login using SSH and SSH client and server. Here is an example of remote file copy using SCP command:

1) Copy a file from a remote directory to a local directory

```
Zhaoy @ myir -o - E - M: ~ $SCP test root@192.168.1.195: / root /
root@192.168.1.195 's password:
Test 100% 0 0.0KB/s
```

Enter the development board home directory to see this file, as follows:

```
[root@myir:/root]# ls
test
```

2) Copy files from local to remote

```
Root @ myir: / root # SCP test zhaoy@192.168.1.13: ~ /
zhaoy@192.168.1.13 's password:
```

After verification, the file is copied from the development board to the $HOME directory of the specified account on the server

```
zhaoy@myir-O-E-M:~$ ls
test
```

You can copy a directory by adding the "-r" parameter. For details, see the help of the SCP command

5.4. TFTP

TFTP uses client and server software to connect and transfer files between different devices. TFTP uses UDP and does not have the login function. It is very simple and suitable for transferring and backing up firmware and configuration files on the device and server.For example, the COMMON U-boot supports TFTP negotiation. You can load the Linux system on the server over the network and start it over the network. The default image file contains the TFTP client provided by Busybox. The command syntax is as follows:

```
[root@myir:/]#  tftp --help
Usage: tftp [-4][-6][-v][-l][-m mode] [host [port]] [-c command]
```

is described as follows

- -g: retrieves a file
- -p: uploads a file
- -l: local file
- -r: remote file
- HOST: IP address of the remote HOST

The TFTP server can choose Linux platform - hpa, also can choose Windows platform TFTPD 32/64 (http://tftpd32.jounin.net/tftpd32_download.html).The following uses ubuntu as an example to describe how to configure the TFTP server.

1) Install the TFTP server

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

- **Configuring the TFTP Service**

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

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

Modify or add the following fields:

```
TFTP_DIRECTORY="/tftpboot"
TFTP_OPTIONS="-l -c -s"
```

- **Restarting the TFTP Service**

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

After configuring the TFTP server, place a test file zImage to the <WORKDIR>/tftpboot/ directory configured above, and you can use the TFTP client to download and upload the file on the development board.

```
[root@myir:/]# tftP-g -r zImage -l zImage 192.168.0.2
```

The above command will download zImage from TFTP server /tftpboot to the current directory on the development board.

```
[root@myir:/]# TFTP -p -l config -r config_01 192.168.0.2
```

The above command uploads the config file in the current directory on the development board to the <WORKDIR>/tftpboot directory previously configured on the TFTP server and renamed to config_01.

5.5. DHCP

DHCP (Dynamic Host Configuration Protocol) is a LAN network protocol.The IP address range is controlled by the server. When a client logs in to the server, it automatically obtains the IP address and subnet mask assigned by the server. DHCP also has both server and client roles. In 4.1.1, we have tested using DHCP client mode to automatically obtain IP addresses.When configuring WiFi AP mode in 4.1.2, we also tested DHCP server mode to assign IP addresses to connected WiFi devices.This section describes how to manually obtain an IP address using the udhcpc command for network debugging.

- **Run the udhcpc command to configure the IP address**

```
[root@myir:/]# udhcpc -i eth0
V1.29.3 udhcpc: started
Failed to kill daemon: No such file or directory
udhcpc: sending discover
Udhcpc: Sending select for 192.168.1.97
Udhcpc: lease of 192.168.1.97 obtained, lease time 1800
Failed to kill daemon: No such file or directory
deleting routers
Adding DNS 192.168.1.1
Adding DNS 192.168.1.1
```

Either way, you can configure the IP address, gateway, subnet mask, and DNS for eth0 as follows:

```
[root@myir:/]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 36:C9:E3:F1:B8:05
          Inet addr: 192.168.1.97 Bcast: 192.168.1.255 Mask: 255.255.255.0
          inet6 addr: fe80::177a:be22:1be1:91e7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:36798 errors:0 dropped:0 overruns:0 frame:0
          TX packets:606 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2887750 (2.7 MiB) TX bytes:56429 (55.1 KiB)
```

```
          Interrupt:65
[root@myir:/]# cat /etc/resolv.conf
# Generated by dhcpcd from eth0.dhcp
# /etc/resolv.conf.head can replace this line
Nameserver 192.168.1.1
# /etc/resolv.conf.tail can replace this line
```

5.6. Iptables

Iptables is a management tool for IPv4 packet filtering and NAT.It is used to set up, maintain, and examine IP packet filtering rule tables in the Linu X kernel.Several different tables can be defined.Each table contains many built-in chains and can also contain user-defined chains.Each chain is a list of rules that can match a set of packets.Each rule specifies how to process the matched packets. Development boards using Linux usually use the iptables tool to configure firewalls.Iptables processes packets according to packet filtering rules, such as accept, reject, and drop.

Iptables is used to test icmp packet interception to prevent PI ng detection by other external devices on the network.Specific commands to use see: https://linux.die.net/man/8/iptables.

1) Configure iptables for the development board

Run the following command on the development board to configure iptables to discard icmp packets and not respond to ping probes from other hosts:

```
[root@myir:/]#  iptables -A INPUT -p icmp --icmp-type 8 -j DROP
[root@myir:/]#
[root@myir:/]#  iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-A INPUT -p icmp -m icmp --icmp-type 8 -j DROP
```

2) Ping test

Ping the development board on the development host and set deadline to 10. The result is as follows:

```
C: \ Users \ Lenovo > ping 192.168.1.195-10 w
Ping 192.168.1.195 has 32 bytes of data:
The request timed out.
```

```
The request timed out.
The request timed out.
The request timed out.
Ping statistics for 192.168.1.195:
    Data packets: Sent = 4, received = 0, Lost = 4 (100% lost),
C:\Users\Lenovo>
C: \ Users \ Lenovo > ping 192.168.1.97-10 w
Ping 192.168.1.97 with 32 bytes of data:
The request timed out.
The request timed out.
The request timed out.
The request timed out.
Ping statistics for 192.168.1.97:
Data packets: Sent = 4, received = 0, Lost = 4 (100% lost),
```

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

- **Delete the corresponding firewall rules**

```
[root@myir:/]#  iptables -F
[root@myir:/]#  iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
```

- **Test the Ping development board again**

```
C: \ Users \ Lenovo > ping 192.168.1.97-10 w
Ping 192.168.1.97 with 32 bytes of data:
Reply from 192.168.1.97: bytes =32 time <1ms TTL=64
Reply from 192.168.1.97: bytes =32 time <1ms TTL=64
Reply from 192.168.1.97: bytes =32 time <1ms TTL=64
Reply from 192.168.1.97: bytes =32 time <1ms TTL=64
Ping statistics for 192.168.1.97:
    Data packets: Sent = 4, received = 4, Lost = 0 (0% lost),
Estimated round-trip time in milliseconds:
```

```
Minimum = 0ms, maximum = 0ms, average = 0ms
```

After the iptables rules are cleared, ping the development board from the development host again.The above example is just a simple demonstration, but iptables is actually quite powerful with a variety of rules that I won't go into here.

5.7. Ethtool

Ethtool is a tool for viewing and modifying Ethernet device parameters. It can be used during network debugging. The following command is used to view information about an Ethernet card and modify its parameters.

First, let's look at the help information for this command through ethtool -h:

```
[root@myir:/]# ethtool --help
Ethtool version 4.19
Usage:
        ethtool DEVNAME Display standard information about device
        ethtool -s|--change DEVNAME     Change generic options
                [ speed %d ]
                [ duplex half|full ]
                [ port tp|aui|bnc|mii|fibre ]
                [ mdix auto|on|off ]
                [ autoneg on|off ]
                [ advertise %x ]
                [ phyad %d ]
                [ xcvr internal|external ]
                [ wol p|u|m|b|a|g|s|f|d... ]
                [ sopass %x:%x:%x:%x:%x:%x ]
                [ msglvl %d | msglvl type on|off ... ]
        ethtool -a|--show-pause DEVNAME Show pause options
.
```

View basic information about Ethernet card of development board:

```
[root@myir:/]# ethtool eth0
Settings for eth0:
        Supported ports: [ TP MII ]
        Supported link modes:   10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
```

```
        Supported pause frame use: No
        Supports auto-negotiation: Yes
        Supported FEC modes: Not reported
        Advertised link modes:  10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Advertised pause frame use: No
        Advertised auto-negotiation: Yes
        Advertised FEC modes: Not reported
        Link partner advertised link modes:  10baseT/Half 10baseT/Full
                                             100baseT/Half 100baseT/Full
        Link partner advertised pause frame use: Symmetric Receive-only
        Link partner advertised auto-negotiation: Yes
        Link partner advertised FEC modes: Not reported
        Speed: 100Mb/s
        Duplex: Full
        Port: MII
        PHYAD: 0
        Transceiver: external
        Auto-negotiation: on
        Link detected: yes
```

Using the ethtool command, you can view that the Ethernet card supports six connection modes: 10 Mbit/s, 100 Mbit/s, and Gigabit half-duplex and full-duplex. The current connection status is negotiated gigabit, full-duplex mode, MII interface, PHY address 6, and so on.

We can also use ethtool to set Ethernet parameters, which play a certain role in Ethernet debugging and diagnosis. For example, we force Ethernet to be set to 100 MBIT/s full duplex and disable auto-negotiation. The command is as follows:

```
# ethtool -s eth0 speed 100 duplex full autoneg off
```

More instructions about ethtool refer to: http://man7.org/linux/man-pages /man8/etht ool. 8. HTML.

5.8. iperf3

Iperf3 is a tool for proactively measuring the maximum achievable bandwidth over IP networks. It supports adjusting various parameters such as test time, buffer size, and protocol (TCP, UDP, AND SCTP for IPV4 and IPV6). Iperf3 can be divided into server mode or client mode according to the role. We can use it to test and check the network bandwidth, TCP window value and retransmission probability in TCP mode, as well as test the packet loss rate, delay and jitter under the specified UDP bandwidth.

We opened Windows PowerShell on the development host, and the host with gigabit network card was used as the server of iperf3, while the development board under test was used as the client to test the performance of TCP and UDP of the development board network card respectively. First install iperf3 on the host as follows:

Connect the server and development board directly through CAT6 network cable, and configure their RESPECTIVE IP addresses.For example, let's set the server IP to 192.168.1.99 and the development board IP to 192.168.1.88, and use the ping command to test that they are connected.

Note: Try not to connect routers or switches, lest the test results be affected by the transmission and forwarding of intermediate devices.

1) Testing TCP Performance

- **Server (192.168.1.99)**

Iperf3 on the server uses the -s parameter to indicate that it works in server mode.

```
PS D:\iperf-3.1.3-win64.> \iperf3.exe -s
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

- **Client (192.168.1.88)**

The iperf3 program running on the development board works in client, TCP mode, where the parameter is described as follows:

- -c 192.168.40.143: works on the client and connects to the server 192.168.40.143
- -I 2: Test result is reported at an interval of 2 seconds
- -T 10: The test duration is 10 seconds

```
[root@myir:/]# iperf3  -c 192.168.1.99 -i 2 -t 10
Connecting to host 192.168.1.88, port 5201
[  5] local 192.168.1.99 port 49692 connected to 192.168.1.99 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-2.00   sec   218 MBytes   914 Mbits/sec  242    625 KBytes

[  5]   2.00-4.00   sec   219 MBytes   918 Mbits/sec   14    551 KBytes

[  5]   4.00-6.00   sec   221 MBytes   927 Mbits/sec    0    609 KBytes

[  5]   6.00-8.00   sec   220 MBytes   923 Mbits/sec    0    631 KBytes

[  5]   8.00-10.00  sec   220 MBytes   923 Mbits/sec  369    554 KBytes

- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec  1.07 GBytes   921 Mbits/sec  625             sender
[  5]   0.00-10.00  sec  1.07 GBytes   918 Mbits/sec                  receiver

iperf Done.
```

The TCP bandwidth is 921 Mbits and no retransmission is performed. The TCP window value is 624KBytes. The server also displays the following test result and continues to listen on the port waiting for the client to connect:

```
PS D:\iperf-3.1.3-win64.>\iperf3.exe -s
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
Accepted Connection from 192.168.1.88, port 35716
```

```
[5] Local 192.168.1.99 port 5201 Connected to 192.168.1.88 port 35718
[ ID] Interval           Transfer     Bandwidth
[5] 0.00-2.00 SEC 220 MBytes 921 Mbits/ SEC
[5] 2.00-4.00 SEC 219 MBytes 917 Mbits/ SEC
[5] 4.00-6.00 SEC 221 MBytes 925 Mbits/ SEC
[5] 6.00-8.00 SEC 219 MBytes 920 Mbits/ SEC
[5] 8.00-10.00 SEC 219 MBytes 917 Mbits/ SEC
[5] 10.00-10.02 SEC 1.38 MBytes 530 Mbits/ SEC
 - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth
[5] 0.00-10.02 SEC 1.07 GBytes 921 Mbits/ SEC 78 Sender
[5] 0.00-10.02 SEC 1.07 GBytes 919 Mbits/ SEC receiver
-----------------------------------------------------------
Server listening on 5201
 -----------------------------------------------------------
```

2) Testing UDP Performance

- **Server (192.168.1.99)**

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

```
PS D: \ iperf 3.1.3 - win64. > \ iperf3 exe - s
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

- **Client (192.168.1.88)**

Iperf3 on the device works in client, UDP mode, where the parameters are described as follows:

- -u: works in UDP mode
- -c 192.168.40.88: works on the client and connects to the server 192.168.40.99
- -I 2: The interval for reporting test results is 2 seconds
- -T 10: The test duration is 10 seconds
- -b 100M: Sets the UDP transmission bandwidth to 100 Mbit/s.

```
[root@myir:/]# iperf3-c 192.168.1.99 -u -i 2-t 10-b 100M
Connecting to host 192.168.1.99, port 5201
[5] Local 192.168.1.88 port 36915 Connected to 192.168.1.99 port 5201
[ ID] Interval           Transfer     Bitrate         Total Datagrams
[5] 0.00-2.00 SEC 23.8 MBytes 100 Mbits/ SEC 17259
[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 100 MBytes 100 Mbits/ SEC 0.000 ms 0/86319 (0%) sender
[5] 0.00-10.00 SEC 119 MBytes 99.4 Mbits/ SEC 0.186 ms 466/86313 (0.54%) r
eceiver

iperf Done.
```

The client finishes the test after 10 seconds and displays the above test result, indicating that UDP does not lose packets when the specified bandwidth is 100Mbps.

At the same time, the server also displays the following test result, and continues to listen on port 5201 waiting for the client to connect:

```
$ $ iperf3 -s
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
Server listening on 5201
Accepted Connection from 192.168.1.99, port 49694
[5] Local 192.168.1.88 port 5201 Connected to 192.168.40.99 port 40126
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total Datag
rams
[5] 0.00-2.00 SEC 23.8 Mbits/ SEC 0.230 ms 0/17240 (0%)
```

```
[5] 2.00 4.00 SEC 23.8 MBytes 99.6 Mbits/SEC 0.161 ms 67/17267 (0.39%)
[5] 4.00-6.01 SEC 23.6 MBytes 98.7 Mbits/ SEC 0.926 ms 124/17208 (0.72%)
[5] 6.01-8.00 SEC 23.7mbits/SEC 0.171ms 136/17330 (0.78%)
[5] 8.00-10.00 SEC 23.6 MBytes 99.2 Mbits/ SEC 0.186 ms 139/17258 (0.81%)
[5] 10.00-10.00 SEC 14.1 KBytes 61.1 Mbits/ SEC 0.186 ms 0/10 (0%)
- - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval Transfer Bandwidth Jitter Lost/Total Datag
rams
[5] 0.00-10.00 SEC 120 MBytes 100 Mbits/ SEC 0.186 ms 466/86313 (0.54%)
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

The client changes the -b parameter and increases the specified UDP bandwidth. The maximum UDP packet loss rate is the maximum bandwidth. The packet loss rate depends on the CPU performance of the server and the buffer size of the nic.

```
[root@myir:/]# iperf3 -u-c 192.168.1.99 -I 2-t 10-b 1000M
Connecting to host 192.168.1.99, port 5201
[5] Local 192.168.1.99 port 45388 Connected to 192.168.1.88 port 5201
[ ID] Interval Transfer Bitrate Total Datagrams
[5] 0.00-2.00 SEC 238 MBytes 1000 Mbits/ SEC 172623
[5] 2.00-4.00 SEC 238 MBytes 1000 Mbits/ SEC 172643
[5] 4.00-6.00 SEC 238 MBytes 1.00 Gbits/ SEC 172673
[5] 6.00-8.00 SEC 238 MBytes 1000 Mbits/ SEC 172602
[5] 8.00-10.00 SEC 238 MBytes 1.00 Gbits/ SEC 172707
- - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval Transfer Bitrate Jitter Lost/Total Datagra
ms
[5] 0.00-10.00 SEC 1.16 GBytes 1000 Mbits/ SEC 0.000 ms 0/863248 (0%) sen
der
```

```
[5] 0.00-10.00 SEC 979 MBytes 821 Mbits/ SEC 0.014 ms 153823/862946 (18%)
 receiver

iperf Done.
```

Iperf3 also has many parameters that can be configured during the test, so users can adjust the test according to actual application needs.For example, you can increase the value of the -t parameter for a long time stress test, or specify the -p parameter for multiple connections and concurrent stress test.More information about iperf3 testing can be found at 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 programs 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 true component programming.

The development board will burn systems out of the factory with Qt runtime libraries and provides a rich HMI demo system, which can be found in the MEasy HMI2.0 Development Manual.

1) Get information about QT

The QT versions supported by the current system are as follows:

```
[root@myir:/]# ls usr/local/
Qt_5.12.5
```

2) QT runtime environment

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

- **qtenv.sh scripts**

On embedded Linux systems, you can use multiple platform plug-ins: 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.

In MYD-YT507H, we use the qtenv.sh script to load the environment variables needed to run the QT program. The script content is as follows:

```
[root@myir:/]# cat /etc/qtenv.sh
```

```
export   QTDIR=/usr/local/Qt_5.12.5
if [ -d $QTDIR ];then

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

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

        export QML_IMPORT_PATH=$QTDIR/qml
        export QML2_IMPORT_PATH=$QTDIR/qml

        TouchDevice=ft5x_ts

        for InputDevices in /sys/class/input/input*
        do
                DeviceName=`cat $InputDevices/name`
                if [ $DeviceName == $TouchDevice ];then
                    TouchDeviceNum=${InputDevices##*input}
                    export QT_QPA_EVDEV_TOUCHSCREEN_PARAMETERS=/dev/i
nput/event$TouchDeviceNum
                    echo "add "/dev/input/event$TouchDeviceNum "to Qt Appli
cation."
                    break
                fi
        done
        if [ ! -n "$TouchDeviceNum" ]; then
         echo "Error:Input device $TouchDevice can not be found,plz check it!
"
```

```
        fi

        export QT_QPA_PLATFORM=eglfs
        export QT_QPA_GENERIC_PLUGINS=evdevtouch
        export QT_QPA_EGLFS_INTEGRATION=eglfs_mali
        #export QT_QPA_FB_HIDECURSOR=1
        #export QT_QPA_EGLFS_HIDECURSOR=1
        #export QT_QPA_EGLFS_ROTATION=90

        export QWS_MOUSE_PROTO=
        export DBUS_SESSION_BUS_ADDRESS=`cat /tmp/dbusaddr`
        mkdir -p /dev/shm
        ulimit -c unlimited
        #debug Launcher &
        mxapp2 &
        echo "find qt5 installed done"
fi
```

- **Display Parameter Configuration**

QT applications can use the QScreen class or the QDesktopWidget to get screen display-related parameters to write applications that match the screen.Getting screen resolution and color depth via QScreen or QDesktopWidget is generally fine, but sometimes the physical dimensions are not necessarily correct due to display drivers.In this case, you can configure and adjust the following parameters to adjust the size of the elements displayed on the actual screen.

Table 6-1. Environment variables related to the QT EGLS plug-in

The environment variable	describe
QT_QPA_EGLFS_INTEGRATION	This environment variable forces the execution of a particular plug-in.For example, setting it to eglfs_kms will use the KMS/DRM back end. Note: On some devices, the special value None is used instead of the actual plug-in.This indicates that using EGL with the frame buffer does not require any special integration.No plug-ins need to be loaded.

QT_QPA_EGLFS_ROTATION	Specifies the rotation to be applied to software rendering content in qWidget-based applications.The supported values are 180, 90, and -90.This variable is not available for OpenGL based Windows, including Qt Quick.Qt Quick applications can instead apply transformations in their QML scenarios.Eglfs Regardless of the type of application, the standard mouse cursor always takes values into account and has a pointer image that is properly placed and rotated.However, special cursor implementations, such as hardware cursors on the KMS/DRM back end, may not support rotation.
QT_QPA_EGLFS_FORCEVSYNC	After this setting, eglFS requests FBIO_WAITFORVSYNC on the frame buffer device after each call to eglSwapBuffers ().This variable is only relevant to the backend that relies on the traditional Linux fbdev subsystem.In general, Qt assumes that a call to eglSwapBuffers () will process vsync, in the case of a default swap interval of 1.If not (for example, due to a driver error), try setting it to QT_QPA_EGLFS_FORCEVSYNC to a non-zero value.
QT_QPA_EGLFS_FORCE888	When set, red, green and blue channel sizes are ignored when eGLFS creates a new context, window or outer surface of the screen.Instead, the plug-in requests an 8-bit configuration per channel.This is helpful for devices that by default choose configurations with less than 32 or 24 bits per pixel (e.g. 5-6-5 or 4-4-4), despite knowing that they are not ideal, for example due to striping effects.Instead of changing the application code, this variable provides a shortcut to enforce the 24 or 32 BPP configuration.
QT_QPA_EGLFS_FB	Overwrite the frame buffer device.The default value is /dev/fb0.On most embedded platforms, this variable is not very relevant because the frame buffer is only used to query Settings such as display size.However, on some devices, this variable provides the ability to specify which monitor to use in multiple display Settings, similar to the fb parameter in LinuxFB.
QT_QPA_EGLFS_PHYSICAL_WIDTH QT_QPA_EGLFS_PHYSICAL_HEIGHT	Specifies the width and height, in millimeters, of the physical screen.On platforms where the value cannot be queried from the frame buffer device /dev/fb0 or otherwise, the default DPI is 100.Use this variable to override any such default values.Setting this variable is important because applications

	based on QWidget or Qt Quick Controls depend on these values.Running these applications with hard-coded Settings can result in user interface elements not being sized for the display being used.
QT_QPA_EGLFS_WIDTH QT_QPA_EGLFS_HEIGHT	Contains the width and height of the screen in pixels.Although eGLFS tries to determine the size from the frame buffer device /dev/fb0, this does not always work.You may need to specify dimensions manually.
QT_QPA_EGLFS_DEPTH	Overlays the color depth of the screen.On platforms where the frame buffer device /dev/fb0 is unavailable or the query is unsuccessful, use the default value 32.Use this variable to override any such default values. Note: This variable affects only the depth value reported by QScreen.It is independent of the EGL configuration and the color depth used for OpenGL rendering.
QT_QPA_EGLFS_SWAPINTERVAL	By default, the exchange interval for 1 requests is.Use this variable to synchronize the vertical refresh of the monitor.Use this variable to override the value of the swap interval.For example, passing 0 disables swap blocking, resulting in running as fast as possible without any synchronization.
QT_QPA_EGLFS_DEBUG	After the setting, some debugging information is printed on the debugging output.For example, when a new context is created, the properties of the input QSurfaceFormat and selected EGL configuration are printed.When used with Qt Quick's QSG_INFO variable, you can get useful information that can be used to solve problems related to EGL configuration.

In most cases, the default Settings are ok, but if the display elements do not match the actual screen, you can adjust the parameters according to the above description.

- **Enter the peripheral configuration**

When there is no windowed system (such as XWindow or Weston) on an embedded Linux device, the mouse, keystroke, or touch device obtains input device information by reading evdev directly or using other intermediate libraries, such as Libinput or tslib.The eGLFS and LinuxFb platform plug-ins include both input methods.About Qt5 input device configuration mode is the most direct view

Qt5 platform used by the plugin code, for example eglfs plug-in input device configuration part code: qtbase - 5. 7.0.x.x/SRC/plugins/platforms/eglfs qeglfsintegration. CPP.

```
void QEglFSIntegration::createInputHandlers()
{
#ifndef QT_NO_LIBINPUT
    if (!qEnvironmentVariableIntValue("QT_QPA_EGLFS_NO_LIBINPUT")) {
        new QLibInputHandler(QLatin1String("libinput"), QString());
        return;
    }
#endif

#if !defined(QT_NO_EVDEV) && (!defined(Q_OS_ANDROID) || defined(Q_OS_AN
DROID_NO_SDK))
    m_kbdMgr = new QEvdevKeyboardManager(QLatin1String("EvdevKeyboard
"), QString() /* spec */, this);
    new QEvdevMouseManager(QLatin1String("EvdevMouse"), QString() /* spec
 */, this);
#ifndef QT_NO_TSLIB
    const bool useTslib = qEnvironmentVariableIntValue("QT_QPA_EGLFS_TSLIB
");
    if (useTslib)
        new QTsLibMouseHandler(QLatin1String("TsLib"), QString() /* spec */);
    else
#endif // QT_NO_TSLIB
        new QEvdevTouchManager(QLatin1String("EvdevTouch"), QString() /* s
pec */, this);
#endif
}
```

Can be seen from the above code, eglfs platform plug-in is used by default EvdevTouch input handler, this approach is often used to deal with capacitive touch, capacitive touch drive event coordinates of the report and the actual

screen area coordinates completely corresponding words, there is no need to do additional processing, if there is a reverse, will be environment variables to make some adjustment, The EvdevTouch input handler supports the following additional parameters:

Table 6-2. Environment variable parameters related to the QT EGLS plug-in EVDEV touch handler

parameter	describe
/dev/input/...	Specifies the name of the input device.If not specified, Qt will either libudev or traverse the available nodes to find the appropriate device.
rotate	On some touch screens, you have to rotate the coordinates by setting the rotate coordinates to 90, 180, or 270.
invertx/inverty	Specifies the parameter used to invert the X or Y coordinate in the input event.

For example, if QT_QPA_EVDEV_TOUCHSCREEN_PARAMETERS passes the following values to the platform plug-in before starting the application, the touch device is explicitly specified as /dev/input.event5, whose coordinates are flipped 180 degrees.This is useful when the orientation of the actual screen and the touch screen don't match.

```
export QT_QPA_EVDEV_TOUCHSCREEN_PARAMETERS=/dev/input/event5:rotate=180
```

To enable tSLIb support, set the QT_QPA_EGLFS_TSLIB (for eglfs) or QT_QPA_FB_TSLIB (for LinuxFb) environment variable to 1.About the specific way of using the tslib reference https://github.com/libts/tslib/blob/master/README.md.

Note: The TSLIb input handler is commonly used for resistive touches, generates mouse events and supports only single touches. Initial use requires screen calibration.

3) Start the Qt program

When we need to run our own QT program on myD-YT 507 platform, we need to modify the Qtenv script file and add the executable QT program path to the script file, so that after running the script, we can configure the correct environment

variables. To run the QT program qt_test, make the following changes at the end of the qtenv.sh script.

```
export QWS_MOUSE_PROTO=
        export DBUS_SESSION_BUS_ADDRESS=`cat /tmp/dbusaddr`
        mkdir -p /dev/shm
        ulimit -c unlimited
        #debug Launcher &
        #mxapp2 &
        /etc/qt_test &
        echo "find qt5 installed done"
fi
```

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

Mxapp2 is already started by default. If you want to run your own Qt application, you need to terminate MXapp2 before starting other applications.

You can kill a process to exit it.

```
[root@myir:/]# killall mxapp2
```

7. Multimedia application

7.1. Camera

This section uses video2lcd and CSI_test_mplane tools to test myD-T507 development board.The main test CSI camera preview, capture frame (take pictures).

1) View basic device information

Access miPI and parallel camera through J2 and J3 respectively, and the following character devices will be generated:

```
[root@myir:/]# ls /dev/video2
/dev/video2      /dev/video0
```

2) Camera Window Preview

- **Use the video2lcd tool to preview**

To use video2lcd to preview the camera, add the following command at the end of the startup script /etc/qtenv.sh to start the preview tool:

```
#debug Launcher &
        #mxapp2 &
       /etc/video2lcd &
        echo "find qt5 installed done"
```

After the screen is added, run the screen initialization command

```
[root@myir:/]# fbinit 0
cleanning /dev/fb0 ...
clean /dev/fb0 finish
```

Then run the startup script /etc/qtenv.sh again and a preview window will be generated on the screen. Click camera video to select mipi camera or parallel port camera to open preview. Three resolutions can be selected during preview: 640*

480,1280 * 720,1920 *1080. Select video0 MIPI camera as shown below and preview with 640*480 resolution.

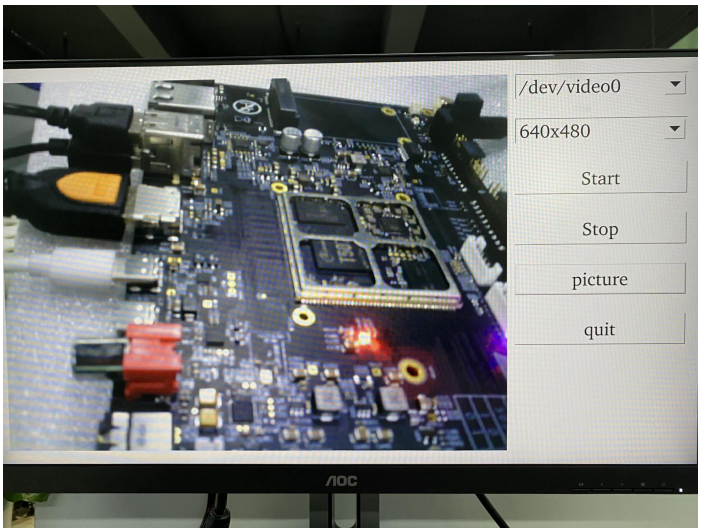

Figure 6-1. MIPI camera preview example

3) Camera taking

Use the csi_test_mplane command to take a picture of the camera and generate a picture file.For example, run the following command to capture a 1920 x 1080 image from a MIPI camera and save the image to the root directory:

```
[root@myir:/]# ./csi_test_mplane -a
please select the video device: 0-video0 1-video1 ......
0
please select the camera: 0-dev0 1-dev1 ......
0
please input the resolution: width height.....
1920
1080
please input the frame saving path.....
/
please input the test mode: 0~3.....
3
please input the test_cnt: >=1.....
1
```

After inputing the corresponding data, the corresponding resolution of the picture can be generated, and the picture can be copied to Windows using the corresponding image decoding tool to view.

7.2. VPU

MYD-YT507H has a VPU, which can be used for hardware encoding/decoding of video.

VPU supports the following decoding formats:

- 1080P HEVC
- H.265
- VP9
- H.264
- VP8

The encoding format is as follows:

- H.264
- H.265

1) Xplayerdemo tools

T5 system has its own player XplayerDemo can achieve video decoding playback function.

```
[root@myir:/mnt/u/video]# xplayerdemo
WARNING: awplayer <log_set_level:30>: Set log level to 3
DEBUG  : awplayer <ReadPluginEntry:194>: read plugin entry adecoder-0 fail!
DEBUG  : awplayer <CdxPluginLoadList:221>: have config 0 entry
DEBUG  : awplayer <CdxPluginLoadList:222>: start to open adecoder lib
DEBUG  : awplayer <CdxPluginLoadList:202>: Load Plugin list vdecoder
DEBUG  : awplayer <ReadPluginEntry:178>: read plugin entry vdecoder-0 ok.
DEBUG  : awplayer <ReadPluginEntry:178>: read plugin entry vdecoder-1 ok.
DEBUG  : awplayer <ReadPluginEntry:178>: read plugin entry vdecoder-2 ok.
DEBUG  : awplayer <ReadPluginEntry:178>: read plugin entry vdecoder-3 ok.
DEBUG  : awplayer <ReadPluginEntry:178>: read plugin entry vdecoder-4 ok.
DEBUG  : awplayer <ReadPluginEntry:178>: read plugin entry vdecoder-5 ok.
DEBUG  : awplayer <ReadPluginEntry:178>: read plugin entry vdecoder-6 ok.
DEBUG  : awplayer <ReadPluginEntry:178>: read plugin entry vdecoder-7 ok.
```

```
DEBUG  : awplayer <ReadPluginEntry:178>: read plugin entry vdecoder-8 ok.
DEBUG  : awplayer <ReadPluginEntry:178>: read plugin entry vdecoder-9 ok.
DEBUG  : awplayer <ReadPluginEntry:178>: read plugin entry vdecoder-10 ok.
DEBUG  : awplayer <ReadPluginEntry:178>: read plugin entry vdecoder-11 ok.
DEBUG  : awplayer <ReadPluginEntry:178>: read plugin entry vdecoder-12 ok.
DEBUG  : awplayer <ReadPluginEntry:194>: read plugin entry vdecoder-13 fail!
DEBUG  : awplayer <CdxPluginLoadList:221>: have config 13 entry
DEBUG  : awplayer <CdxPluginLoadList:222>: start to open vdecoder lib
DEBUG  : awplayer <DlOpenPlugin:96>: plugin vdecoder.avs comment is "avs_
vdecoder"
DEBUG  : awplayer <DlOpenPlugin:97>: plugin open lib: libawavs.so
DEBUG  : awplayer <DlOpenPlugin:116>: plugin init : CedarPluginVDInit
DEBUG  : awplayer <DlOpenPlugin:96>: plugin vdecoder.h264 comment is "h2
64_vdecoder"
DEBUG  : awplayer <DlOpenPlugin:97>: plugin open lib: libawh264.so
DEBUG  : awplayer <DlOpenPlugin:116>: plugin init : CedarPluginVDInit
DEBUG  : awplayer <DlOpenPlugin:96>: plugin vdecoder.h265 comment is "h2
65_vdecoder"
DEBUG  : awplayer <DlOpenPlugin:97>: plugin open lib: libawh265.so
DEBUG  : awplayer <DlOpenPlugin:116>: plugin init : CedarPluginVDInit
DEBUG  : awplayer <DlOpenPlugin:96>: plugin vdecoder.mjpeg comment is "
mjpeg_vdecoder"
DEBUG  : awplayer <DlOpenPlugin:97>: plugin open lib: libawmjpeg.so
DEBUG  : awplayer <DlOpenPlugin:116>: plugin init : CedarPluginVDInit
DEBUG  : awplayer <DlOpenPlugin:96>: plugin vdecoder.mjpegplus comment i
s "mjpegplus_vdecoder"
DEBUG  : awplayer <DlOpenPlugin:97>: plugin open lib: libawmjpegplus.so
DEBUG  : awplayer <DlOpenPlugin:116>: plugin init : CedarPluginVDInit
DEBUG  : awplayer <DlOpenPlugin:96>: plugin vdecoder.mpeg2 comment is "
mpeg2_vdecoder"
DEBUG  : awplayer <DlOpenPlugin:97>: plugin open lib: libawmpeg2.so
DEBUG  : awplayer <DlOpenPlugin:116>: plugin init : CedarPluginVDInit
```

```
DEBUG  : awplayer <DlOpenPlugin:96>: plugin vdecoder.mpeg4base comment
 is "mpeg4base_vdecoder"
DEBUG  : awplayer <DlOpenPlugin:97>: plugin open lib: libawmpeg4base.so
WARNING: awplayer <DlOpenPlugin:112>: Invalid plugin,function CedarPluginV
DInit not found.
DEBUG  : awplayer <DlOpenPlugin:96>: plugin vdecoder.mpeg4dx comment is
 "mpeg4dx_vdecoder"
DEBUG  : awplayer <DlOpenPlugin:97>: plugin open lib: libawmpeg4dx.so
DEBUG  : awplayer <DlOpenPlugin:116>: plugin init : CedarPluginVDInit
DEBUG  : awplayer <DlOpenPlugin:96>: plugin vdecoder.mpeg4h263 comment
 is "mpeg4h263_vdecoder"
DEBUG  : awplayer <DlOpenPlugin:97>: plugin open lib: libawmpeg4h263.so
DEBUG  : awplayer <DlOpenPlugin:116>: plugin init : CedarPluginVDInit
DEBUG  : awplayer <DlOpenPlugin:96>: plugin vdecoder.mpeg4normal comme
nt is "mpeg4normal_vdecoder"
DEBUG  : awplayer <DlOpenPlugin:97>: plugin open lib: libawmpeg4normal.so
DEBUG  : awplayer <DlOpenPlugin:116>: plugin init : CedarPluginVDInit
DEBUG  : awplayer <DlOpenPlugin:96>: plugin vdecoder.mpeg4vp6 comment
is "mpeg4vp6_vdecoder"
DEBUG  : awplayer <DlOpenPlugin:97>: plugin open lib: libawmpeg4vp6.so
DEBUG  : awplayer <DlOpenPlugin:116>: plugin init : CedarPluginVDInit
DEBUG  : awplayer <DlOpenPlugin:96>: plugin vdecoder.vp8 comment is "vp8
_vdecoder"
DEBUG  : awplayer <DlOpenPlugin:97>: plugin open lib: libawvp8.so
DEBUG  : awplayer <DlOpenPlugin:116>: plugin init : CedarPluginVDInit
DEBUG  : awplayer <DlOpenPlugin:96>: plugin vdecoder.wmv3 comment is "w
mv3_vdecoder"
DEBUG  : awplayer <DlOpenPlugin:97>: plugin open lib: libawwmv3.so
DEBUG  : awplayer <DlOpenPlugin:116>: plugin init : CedarPluginVDInit
DEBUG  : awplayer <CdxPluginLoadList:202>: Load Plugin list plugin
DEBUG  : awplayer <ReadPluginEntry:194>: read plugin entry plugin-0 fail!
DEBUG  : awplayer <CdxPluginLoadList:221>: have config 0 entry
DEBUG  : awplayer <CdxPluginLoadList:222>: start to open plugin lib
```

```
DEBUG   : awplayer <AwStreamInit:92>: aw stream init...
DEBUG   : awplayer <AwStreamInit:124>: stream list size:6
INFO    : awplayer <AwParserInit:397>: aw parser init...
DEBUG   : awplayer <AwParserInit:467>: aw parser size:18
DEBUG   : awplayer <AwMuxerInit:53>: aw muxer init ..
DEBUG   : awplayer <AwMuxerInit:62>: aw muxer size:4

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* 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.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
DEBUG   : awplayer <XPlayerCreate:214>: XPlayerCreate.
DEBUG   : awplayer <LogVersionInfo:34>:
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> CedarX <<<<<<<<<<<<<<<<
<<<<<<<<<<<<<<<<
Tag: CedarX - 2.8.0
branch: master
commit: 967535b8ff6a073cb4f38e85a4ae5fa6008014d8
date  : Mon, 15 May 2017 01:30:22 +0000 (09:30 +0800)
author:
------------------------------------------------------------------------

DEBUG   : awplayer <LayerCreate_DE:1497>: LayerCreate.

(getInstance 91)
init chwd sucess !
(HwDisplay 284)
(hwd_init 1259)
the output type: 0(4)
```

```
DEBUG   : awplayer <LayerCreate_DE:1549>: screen:w 1920, screen:h 1080 disp
xfang tinyalsa SoundDeviceCreate
DEBUG   : awplayer <SubtitleCreate:88>: ==== pCallback: 0x401d7c, pUser: 0x
7fc0ea32b0
DEBUG   : awplayer <XPlayerSetVideoSurfaceTexture:591>: setVideoSurfaceText
ure, surface = 0x1ca21410
DEBUG   : awplayer <XPlayerThread:1870>: process message XPLAYER_COMMA
ND_SET_SURFACE.
DEBUG   : awplayer <XPlayerThread:1931>: ==== process message XPLAYER_C
OMMAND_SET_SUBCTRL.
DEBUG   : awplayer <PlayerSetSubCtrl:682>: === PlayerSetSubCtrl
DEBUG   : awplayer <XPlayerSetDeinterlace:692>: set deinterlace
DEBUG   : awplayer <XPlayerThread:1946>: ==== process message XPLAYER_C
OMMAND_SET_SUBCTRL.

demoPlayer#
```

View player supported 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 belo
w:
*       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:
*                 stop the playback.
*     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.
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * *
```

2) Play the video

Take playing mounted USB disk 4kcanadA_h264.mp4 as an example:

```
demoPlayer# set url:/mnt/u/video/4KCanada_h264.mp4
DEBUG   : awplayer <XPlayerSetDataSourceUrl:456>: setDataSource(url), url='/mnt/u/video/4KCanada_h264.mp4'
INFO    : awplayer <XPlayerThread:1707>: process message XPLAYER_COMMAND_SET_SOURCE.
DEBUG   : awplayer <XPlayerPrepare:741>: prepare
DEBUG   : awplayer <XPlayerThread:1960>: process message XPLAYER_COMMAND_PREPARE. mPriData->mStatus: 1
DEBUG   : demuxComponent <DemuxThread:1783>: process message DEMUX_COMMAND_PREPARE.
DEBUG   : demuxComponent <DemuxThread:1850>: === prepare msg
DEBUG   : awplayer <CdxParserPrepare:757>: source uri 'file:///mnt/u/video/4KCanada_h264.mp4'
.
INFO    : awplayer <XPlayerThread:1996>: xxxxxxxxxx video size: width = 1920, height = 960
++++ video width: 1920, height: 960
DEBUG   : awplayer <CallbackForAwPlayer:440>: info : preared
info: prepare ok.
preparing...
```

It indicates that the play is ready. Run the play command to start the play.

```
demoPlayer# play
DEBUG   : awplayer <XPlayerStart:771>: start
DEBUG   : awplayer <XPlayerThread:2140>: process message XPLAYER_COMMAND_START.
DEBUG   : awplayer <PlayerStart:730>: player start
DEBUG   : awplayer <BaseCompPostAndWait:61>: video decoder receive cmd: start
debug   : cedarc <SbmFrameReset:613>: ** wait for reset sem
debug   : cedarc <ProcessThread:1653>: *** post reset sem
debug   : cedarc <SbmFrameReset:615>: ** wait for reset sem ok
debug   : cedarc <SbmFrameReset:620>: SbmFrameReset finish
```

```
DEBUG   : awplayer <BaseCompPostAndWait:61>: audio decoder receive cmd:
start
debug   : cedarc <H264ProcessExtraData2:543>: H264ProcessNaluUnit, bNeedFi
ndSPS = 0, bNeedFindPPS = 0
(Allwinner Audio Middle Layer),line(958) : Create Decoder!!= = = = =
DEBUG   : audioDecItf <handleStart:1064>: Create libadecoder success...
(Allwinner Audio Middle Layer),line(592) : AudioDec_Installaudiolib ok
(Allwinner Audio Middle Layer),line(595) : audio decoder init start ...
(AllwinnerAlibs),line(626) : libaw_aacdec.so open, use dlopen!
(AllwinnerAlibs),line(660) : Khan----Loading 'libaw_aacdec.so' success!
.
```

7.3. Audio

This chapter is to test playing audio.There are three ways, namely audio HeadPhone, audio LineinOut and audio SPDIF.

1) A debugging tool

- **Tinycap recording test tool**

Used to operate the audio recording device node of the audio card.

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

For example, the following command will use the 0th device of sound card 3 to record a 10s-long audio file named file.wav and save it in the current path.

```
tinycap_ahub file.wav  -aD 2 -ad 2 -D 3 -d 0 -t 10
```

- **alsamixer**

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

```
[root@myir:/]# 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
```

As shown in the following figure, press ← or → to control the cursor and press ↑ and ↓ to adjust the parameter size.

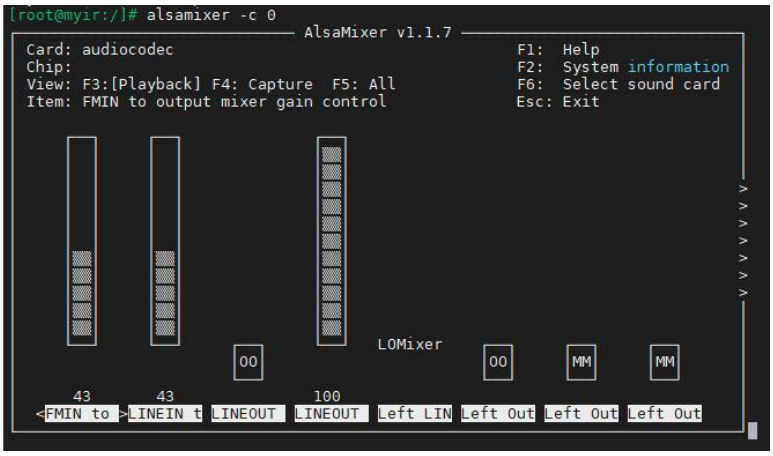

Figure 6.2.1 UI for adjusting player parameters

2) Playback of audio

- **Audio LineinOut**

Use tinyPlay to play the test tool.Used to operate audio player device nodes of Audiocedec, SPDIF, USB Audio.

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

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

```
[root@myir:/mnt/u/audio]# tinyplay file.wav
Playing sample: 2 ch, 44100 hz, 16 bit
[14896.076203] #### sunxi_start_desc chan start
```

```
[14896.081022] # # # # Common register:
[14896.081022] mask0 (0000) : 0 x00000002
[14896.081022] mask1 (0004) : 0 x00000000
[14896.081022] pend0 (0010) : 0 x00000000
[14896.081022] pend1 (0014) : 0 x00000000
[14896.081022] secur (0020) : 0 x000000ff
[14896.081022] _gate (0028) : 0 x00000007
[14896.081022] stats (0030) : 0 x00000001
[14896.112812] #### Chan 0 reg:
[14896.112812] ___en (0100) : 0 x00000001
[14896.112812] pause (0104) : 0 x00000000
[14896.112812] start (0108) : 0 x7f403020
[14896.112812] __cfg (010 c) : 0 x03460240
[14896.112812] __src (0110) : 0 x7f481680
[14896.112812] __dst (0114) : 0 x05096020
[14896.112812] count (0118) : 0 x00000a00
[14896.112812] _para (011 c) : 0 x00000008
[14896.112812]
```

- **Audio HeadPhone**

Use tinyplay_ahub to play the test tool.Used to operate HDMI (I2S1), I2S2, I2S3 audio player device nodes.

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

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

```
[root@myir:/mnt/u/audio]# tinyplay_ahub file.wav -aD 2 -ad 2 -D 3 -d 0
Playing : file.wav
[14834.777101] raw_flag value is 0
Playing sample: 2 ch, 44100 hz, 16 bit
[14834.783845] #### sunxi_start_desc chan start
[14834.789179] # # # # Common register:
```

```
[14834.789179] mask0 (0000) : 0 x00000002
[14834.789179] mask1 (0004) : 0 x00000000
[14834.789179] pend0 (0010) : 0 x00000000
[14834.789179] pend1 (0014) : 0 x00000000
[14834.789179] secur (0020) : 0 x000000ff
[14834.789179] _gate (0028) : 0 x00000007
[14834.789179] stats (0030) : 0 x00000001
[14834.820963] #### Chan 0 reg:
[14834.820963] ___en (0100) : 0 x00000001
[14834.820963] pause (0104) : 0 x00000000
[14834.820963] start (0108) : 0 x7f403020
[14834.820963] __cfg (010 c) : 0 x03450240
[14834.820963] __src (0110) : 0 x7f680000
[14834.820963] __dst (0114) : 0 x05097090
[14834.820963] count (0118) : 0 x00001000
[14834.820963] _para (011 c) : 0 x00000008
[14834.820963]
```

- **SPDIF**

When playing SPDIF audio, also use tinyplay_ahub to play the test tool.We insert the playback device into the J11 dock and run the following command to play the audio.

```
[root@myir:/mnt/u/audio]# tinyplay_ahub caihong.wav -aD 1 -ad 0
Playing : caihong.wav
Playing sample: 2 ch, 44100 hz, 16 bit
[14963.280791] #### sunxi_start_desc chan start
[14963.288294] # # # # Common register:
[14963.288294] mask0 (0000) : 0 x00000022
[14963.288294] mask1 (0004) : 0 x00000000
[14963.288294] pend0 (0010) : 0 x00000000
[14963.288294] pend1 (0014) : 0 x00000000
[14963.288294] secur (0020) : 0 x000000ff
[14963.288294] _gate (0028) : 0 x00000007
```

```
[14963.288294] stats (0030) : 0 x00000002
[14963.320079] #### Chan 1 reg:
[14963.320079] ___en (0140) : 0 x00000001
[14963.320079] pause (0144) : 0 x00000000
[14963.320079] start (0148) : 0 x7f403020
[14963.320079] __cfg (014 c) : 0 x03820280
[14963.320079] __src (0150) : 0 x7f500000
[14963.320079] __dst (0154) : 0 x05093020
[14963.320079] count (0158) : 0 x00001000
[14963.320079] _para (015 c) : 0 x00000008
[14963.320079]
```

8. System tools

The default image contains some common system tools for users to view and manage system resources in system debugging or actual deployed products, and can be called in SHELL scripts or other applications.These tools may not fully meet users' system customization requirements. In this case, system developers need to make appropriate adjustments according to the actual situation.

8.1. Decompression tool

This section tests the decompression tool of the system.You can compress multiple files into a compressed package to facilitate file transfer.Decompression can restore the compressed files to the original size for easy use.This section uses file systems such as tar, gzip, and gunzip as examples.

1) tar tool

The tar tool we now use in Linux can not only package files, but also compress, view, add, and unpack them.Here is the package operation.Enter the following command to view the tar syntax format:

```
[root@myir:/]# tar --help
BusyBox V1.29.3 (2022-01-28 15:52:25 CST) Multi-call Binary.

Usage: tar c|x|t [-hvokO] [-f TARFILE] [-C DIR] [-T FILE] [-X FILE] [--exclude PA
TTERN]... [FILE]...
Create, extract, or list files from a tar file
        c       Create
        x       Extract
        t       List
        -f FILE Name of TARFILE ('-' for stdin/out)
        -C DIR  Change to DIR before operation
        -v      Verbose
        -O      Extract to stdout
```

```
        -o      Don't restore user:group
        -k      Don't replace existing files
        -h      Follow symlinks
        -T FILE File with names to include
        -X FILE File with glob patterns to exclude
        --exclude PATTERN       Glob pattern to exclude
```

- **Using tar compression**

Create a test.txt file and type the following command to package the file in xxx.tar.gz format:

```
[root@myir:/]# tar -cf test.tar.gz test.txt
[root@myir:/]# ls
test.tar.gz
```

- **Decompress using tar**

Unpack the tar.gz file

```
[root@myir:/]# tar -xvf test.tar.gz
test.txt
[root@myir:/]# ls
test.txt
```

2) Gzip compression tool

- **Syntax format**

Gzip is a command used to compress and decompress files in Linux. It is convenient and easy to use.Enter the following command on the development board terminal to view the GZIP syntax:

```
[root@myir:/]# gzip --hple
gzip: unrecognized option '--hple'
BusyBox V1.29.3 (2022-01-28 15:52:25 CST) Multi-call Binary.
Usage: gzip [-cfkdt] [FILE]...
Compress FILEs (or stdin)
```

```
        -d          Decompress
        -t          Test file integrity
        -c          Write to stdout
        -f          Force
        -k          Keep input files
```

- **Compress the file with gzip**

```
[root@myir:/]# gzip test.txt
[root@myir:/]# ls
test.tar.gz
```

- **Unzip the file with gunzip**

```
[root@myir:/]# gunzip test.txt.gz
[root@myir:/]# ls
test.txt
```

8.2. File system tools

This section describes several common file system management tools used to test the system.The built-in file system tools mount, mkfs, fsck, and dumpe2fs.

1) Mount Mount tool

Mount is a Linux command that attaches a partition to a folder in Linux, thus associating the partition with the directory. Therefore, if we access the folder, we can access the partition.

```
[root@myir:/]# mount -h
Usage:
 mount [-lhV]
 mount -a [options]
 mount [options] [--source] <source> | [--target] <directory>
 mount [options] <source> <directory>
 mount <operation> <mountpoint> [<target>]

Mount a filesystem.
```

- **Mount the U disk**

```
[root@myir:/]# mount /dev/sda1 /mnt/
```

2) mkfs format tool

After partitioning the hard disk, the next step is to set up the Linux file system.This is similar to formatting hard disks in Windows.Creating a file system on a disk partition will flush out data on the partition and cannot be restored. Therefore, ensure that data on the partition is no longer used before creating a file system.To create a file system, run the MKFS command in the following syntax:

```
[root@myir:/]# mkfs -h
Usage:
 mkfs [options] [-t <type>] [fs-options] <device> [<size>]
```

```
Make a Linux filesystem.
Options:
 -t, --type=<type>   filesystem type; when unspecified, ext2 is used
      fs-options       parameters for the real filesystem builder
      <device>          path to the device to be used
      <size>            number of blocks to be used on the device
 -V, --verbose        explain what is being done;
                        specifying -V more than once will cause a dry-run
 -h, --help           display this help
 -V, --version        display version

For more details see mkfs(8).
```

- **Formatting a USB Flash drive**

```
[root@myir:/]# umount /mnt
[root@myir:/]# umount /run/media/sda1/

[root@myir:/]# mkfs -t ext3 -V -c /dev/sda1
MKFS from util - Linux 2.34
mkfs.ext3 -c /dev/sda1
Mke2fs 1.45.3 (14 - Jul - 2019)
/dev/sda1 contains a vfat file system
Proceed anyway? (y,N) y
Creating filesystem with 3890688 4k blocks and 972944 inodes
Filesystem UUID: 97810d2b-76aa-44a4-9409-2c70de71eca0
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 265
4208

Checking for bad blocks (read-only test):
[6873.703223] Audit: Type =1006 Audit (1599269401.268:4): pid=947 uid=0 ol
d-auid=4294967295 auid=0 tty=(none) old-ses=4294967295 ses=3 res=1
done
Allocating group tables: done
```

```
Writing inode tables:
done
Creating journal (16384 blocks):
done
Writing superblocks and filesystem accounting information: done
```

3) fsck file repair tool

The fsck command is used to check the correctness of a file system. If a file system error occurs, the fsck command can be used to rectify the error.And fix the Linux disk.Such as:

```
[root@myir:/]#  fsck -a /dev/mmcblk0p1
FSCK from util - Linux 2.33
```

4) dumpe2fs

Prints information about super blocks and blocks groups of existing file systems on a specific device.To view the application syntax, enter the following command:

```
[root@myir:/]# dumpe2fs -h
Dumpe2fs 1.44.5 (15 - Dec - 2018)
Usage: dumpe2fs [-bfghimxV] [-o superblock=<num>] [-o blocksize=<num>]
device
```

View the detailed properties of a formatted file system. For example, run the following command to view details about a disk:

```
[root@myir:/]# dumpe2fs /dev/sda1
Dumpe2fs 1.45.3 (14 - Jul - 2019)
Filesystem volume name:   <none>
Last mounted on:          <not available>
Filesystem UUID:          97810d2b-76aa-44a4-9409-2c70de71eca0
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal ext_attr resize_inode dir_index filetype sp
arse_super large_file
```

```
Filesystem flags:         unsigned_directory_hash
```

Check the number of inodes on a disk. Inodes also consume disk space. Therefore, the operating system automatically divides the disk into two areas when formatting the disk.One is the data area, storing file data;The other is the inode table, which stores the information contained in the inode.

```
[root@myir:/]# dumpe2fs /dev/sda1 | grep -i "inode size"
Dumpe2fs 1.45.3 (14 - Jul - 2019)
Inode size:               256
```

When viewing the number of blocks on a disk, the operating system reads disks in consecutive sectors at a time, that is, one block at a time, rather than one sector at a time.This "block", composed of multiple sectors, is the smallest unit of file access.

```
[root@myir:/]# dumpe2fs /dev/sda1 | grep -i "block size"
Dumpe2fs 1.45.3 (14 - Jul - 2019)
Block size:               4096
```

8.3. Disk Management tool

This section mainly tests the disk management tools of the system. This section describes several common disk management tools.The system provides the disk management tools fdisk, dd, mkfs, du, df, cfdisk, and fsck.You can use these commands to monitor the disk usage.

1) fdisk Disk partition tool

The fdisk disk partitioning tool has applications in DOS, Windows, and Linux.In Linux, fdisk is a menu-based command.To partition a hard disk using fdisk, you can directly add the hard disk to be partitioned as a parameter after the fdisk command. The syntax is as follows:

```
[root@myir:/]# fdisk -h
BusyBox V1.29.3 (2022-01-28 15:52:25 CST) Multi-call Binary.
Usage: fdisk [-ul] [-C CYLINDERS] [-H HEADS] [-S SECTORS] [-b SSZ] DISK
Change partition table

        -u              Start and End are in sectors (instead of cylinders)
        -l              Show partition table for each DISK, then exit
        -b 2048         (for certain MO disks) use 2048-byte sectors
        -C CYLINDERS    Set number of cylinders/heads/sectors
        -H HEADS        Typically 255
        -S SECTORS      Typically 63
```

Partition eMMC:

```
[root@myir:/]# fdisk /dev/mmcblk0
Found valid GPT with protective MBR; using GPT

Command (m for help): m
Command Action
o       create a new empty DOS partition table
p       print the partition table
```

```
q          quit without saving changes
s          create a new empty Sun disklabel

Command (m for help):
```

2) dd Copy Commands

The dd command is used to copy the input file to the output file.And format conversion can be carried out in the process of replication.The dd command is different from the cp command in that the dd command can be executed on a floppy disk without creating a file system. The data copied to the floppy disk is actually an image file.Similar to the diskcopy command in DOS.The format of the dd command is dd [<if= Input file name/device name >] [<of= Output file name/device name >] [bs= block size] [count = number of blocks].

Create a file with a size of 2M.

```
[root@myir:/]# time dd  if=/dev/zero of=ffmpeg1 bs=2M count=1 conv=fsync
1+0 records in
1+0 records out

The real 0 m0. 094 s
The user 0 m0. The 000 s
Sys 0 m0. The 027 s
```

3) du Disk usage statistics tool

The du command is used to display the disk space usage.This command displays data blocks occupied by subdirectories of a specified directory level by level.The syntax of du is as follows:

```
[root@myir:/]# du --help
BusyBox V1.29.3 (2022-01-28 15:52:25 CST) Multi-call Binary.
Usage: du [-aHLdclsxhmk] [FILE]...
Summarize disk space used for each FILE and/or directory
```

```
        -a      Show file sizes too
        -L      Follow all symlinks
        -H      Follow symlinks on command line
        -d N    Limit output to directories (and files with -a) of depth < N
        -c      Show grand total
        -l      Count sizes many times if hard linked
        -s      Display only a total for each argument
        -x      Skip directories on different filesystems
        -h      Sizes in human readable format (e.g., 1K 243M 2G)
        -m      Sizes in megabytes
        -k      Sizes in kilobytes (default)
```

Some parameters:

- -a: Displays the sizes of all directories or files
- -h: the unit is K,M, or G to improve the readability of information
- -k: The output is in KB
- -m: The output is in MB

Count the size of the file generated by the dd command:

```
[root@myir:/]# du ffmpeg1
2048    ffmpeg1
[root@myir:/]# du -h ffmpeg1
2.0 M ffmpeg1
```

4) df Disk statistics tool

This command is used to display disk usage statistics of the file system running on Linux. The general usage is as follows:

```
[root@myir:/]# df -help
df: invalid option -- 'e'
BusyBox V1.29.3 (2022-01-28 15:52:25 CST) Multi-call Binary.
Usage: df [-PkmhT] [FILESYSTEM]...
Print filesystem usage statistics
```

```
-P      POSIX output format
-k      1024-byte blocks (default)
-m      1M-byte blocks
-h      Human readable (e.g. 1K 243M 2G)
-T      Print filesystem type
```

Some parameters:

- -h: displays in appropriate units according to the size used
- -i: Displays the number of inodes in a partition and inode usage
- -t: Displays the type of the file system

Run the following command to view the number of inodes in a partition and inode usage:

```
[root@myir:/]# df -h
Filesystem              Size      Used      Available  Use%      Mo
unted on
/dev/mmcblk0p4 1.9g 805.6m 1.1g 41% /
TMPFS 490.1m 64.0k 490.0m 0% / TMP
TMPFS 490.1m 32.0K 490.1m 0% /run
Devtmpfs 480.7m 0 480.7m 0% /dev
/dev/mmcblk0p8 2.9g 9.0m 2.7g 0% /media
/dev/shm /dev/shm /dev/shm /dev/shm
```

Inodes are partitioned by the system when we format them. Inodes are related to disk partition size.When our inode usage reaches 100%, we can't write data to disk even if we still have free disk space.

8.4. Process management tool

Process is also an important concept in the operating system, it is a process of execution, the program is a static description of the process, the system runs every program is running in its process.All processes in Linux are related to each other and have a parent except for the initializer process.A new process is not created, but is copied or copied from a previous process.All processes in Linux are derived from a single init process with process number . The Linux system includes three different types of processes, each with its own characteristics and attributes:

- Interactive process: A process started by a Shell that can run either in the foreground or in the background.
- Batch process: This process has no connection to the terminal and is a sequence of processes.Such processes are submitted to processes that wait for the queue to execute sequentially.
- Monitor processes (daemons) : Daemons are always active and usually run in the background. Daemons are usually started by the system with automatic script activation or by root.

In Linux, process management is an important step. Process management is usually implemented by using the process management tool. The common process management commands in Linux are ps, top, and vmstat kill.

1) ps Displays the current process tool

- **Syntax format**

The running status of the current system process is displayed in the following syntax:

```
[root@myir:/]# ps --help
BusyBox V1.29.3 (2022-01-28 15:52:25 CST) Multi-call Binary.

Usage: ps [-o COL1,COL2=HEADER]
```

```
Show list of processes

        -o COL1,COL2=HEADER        Select columns for display
```

Description of some parameters:

- -u: displays the user-centered process status.
- -a: indicates processes unrelated to terminals.
- -x: processes related to terminals. (threads are lightweight processes;)
- Usually the above commands are combined: aux.
- --e: displays all processes.That's the same thing as ax;
- -f: Displays program information in complete format.
- Usually, the preceding commands are combined: ef
- -h: displays the number of processes at the process level
- -f: Displays more program information

The command is usually combined: eHF.

- **Displays information about all processes**

```
[root@myir:/]# ps
PID   USER     COMMAND
    1 root     init
    2 root     [kthreadd]
    4 root     [kworker/0:0H]
    6 root     [ksoftirqd/0]
    7 root     [rcu_preempt]
    8 root     [rcu_sched]
    9 root     [rcu_bh]
   10 root     [migration/0]
   11 root     [lru-add-drain]
   12 root     [cpuhp/0]
   13 root     [cpuhp/1]
   14 root     [migration/1]
   15 root     [ksoftirqd/1]
```

```
17 root      [kworker/1:0H]
18 root      [cpuhp/2]
19 root      [migration/2]
20 root      [ksoftirqd/2]
22 root      [kworker/2:0H]
23 root      [cpuhp/3]
24 root      [migration/3]
.
```

2) Top Displays Linux processes

- **Syntax format**

The top command puts quite a bit of overall system performance information on one screen.Display content can also be changed interactively.Dynamically monitors the running status of a process. The syntax of top is as follows:

```
[root@myir:/]# top --help
BusyBox V1.29.3 (2022-01-28 15:52:25 CST) Multi-call Binary.

Usage: top [-b] [-nCOUNT] [-dSECONDS]

Provide a view of process activity in real time.
Read the status of all processes from /proc each SECONDS
and display a screenful of them.
Keys:
        N/M/P/T: sort by pid/mem/cpu/time
        R: reverse sort
        Q,^C: exit

Options:
        -b      Batch mode
        -n N    Exit after N iterations
        -d N    Delay between updates
```

- **Dynamically view system processes**

```
[root@myir:/]# top
Mem: 234716K used, 769024K free, 88K shrd, 10208K buff, 62924K cached
CPU:   0% usr   2% sys   0% nic  97% idle   0% io   0% irq   0% sirq
Load Average: 0.00 0.00 0.00 1/132 32743
  PID  PPID USER     STAT   VSZ %VSZ %CPU COMMAND
32736  1903 root     R     2580   0%   2% top
 1705     1 root     S     294m  30%   0% /etc/video2lcd
 1532     1 root     S     156m  16%   0% adbd
 1903     1 root     S     3624   0%   0% -/bin/sh
 1345     1 root     S     2716   0%   0% dbus-daemon --system
    1     0 root     S     2580   0%   0% init
 1316     1 root     S     2580   0%   0% /sbin/syslogd -n
 1321     1 root     S     2580   0%   0% /sbin/klogd -n
 1417     1 root     S     2328   0%   0% /usr/sbin/dropbear -R
 1398     1 root     S     2192   0%   0% /sbin/dhcpcd -f /etc/dhcpcd.conf
 1455     1 root     S     2160   0%   0% /usr/sbin/tftpd -c -l -s /var/lib/tf
tpboot
  589     2 root     SW       0   0%   0% [vsync proc 0]
    7     2 root     SW       0   0%   0% [rcu_preempt]
 1226     2 root     SW       0   0%   0% [cec thread]
 6256     2 root     SW       0   0%   0% [kworker/u8:0]
28128     2 root     SW       0   0%   0% [kworker/u8:1]
  921     2 root     SW       0   0%   0% [kworker/0:1]
 1225     2 root     SW       0   0%   0% [hdmi proc]
 1232     2 root     SW       0   0%   0% [tve detect]
  551     2 root     SW       0   0%   0% [kworker/2:1]
 1504     2 root     SW       0   0%   0% [mali-simple-pow]
 8359     2 root     SW       0   0%   0% [kworker/2:0]
  548     2 root     SW       0   0%   0% [kworker/1:1]
  554     2 root     SW       0   0%   0% [kworker/3:1]
```

3) Kill Process termination tool

- **Syntax format**

Sends the specified signal to the corresponding process.Not specifying a model will send SIGTERM (15) to terminate the specified process.If the program cannot be terminated, the "-kill" parameter can be used, and the signal it sends is SIGKILL(9), which will force the process to end. You can use the ps command or the jobs command to view the process number.The root user can affect user processes. Non-root users can only affect their own processes.The syntax of the kill command is as follows:

```
[root@myir:/]# kill --help
kill: kill [-s sigspec | -n signum | -sigspec] pid | jobspec ... or kill -l [sigspec]
    Send a signal to a job.

    Send the processes identified by PID or JOBSPEC the signal named by
    SIGSPEC or SIGNUM.  If neither SIGSPEC nor SIGNUM is present, then
    SIGTERM is assumed.

    Options:
      -s sig     SIG is a signal name
      -n sig     SIG is a signal number
      -l         list the signal names; if arguments follow `-l' they are
                 assumed to be signal numbers for which names should be lis
ted
      -L         synonym for -l

    Kill is a shell builtin for two reasons: it allows job IDs to be used
    instead of process IDs, and allows processes to be killed if the limit
    on processes that you can create is reached.
```

Description of some parameters:

- -s: indicates the signal to be sent
- -p: simulates sending signals
- -l: Specifies the name list of signals

- Pid: INDICATES the ID of the process to abort
- Signal: indicates the Signal

First use ps -ef and pipeline command to determine the PID of the process to kill,

```
[root@myir:/]# ps -ef | grep /etc/video2lcd
 1705 root       /etc/video2lcd
 6254 root       grep /etc/video2lcd
```

Then type the following command to terminate the process:

```
[root@myir:/]# kill 1705
```

Killall terminates all processes in the same process group, allowing you to specify the name of the process to be terminated instead of the PID process number:

```
[root@myir:/]# killall /etc/video2lcd
```

9. Development support

This chapter mainly introduces some basic information about secondary development for the current SDK. The current SDK provides two types of reference images. One is myir-image-core, which is mainly for guI-free applications.The other is myir-image-full, which adds some GUI applications on the basis of myir-image-core.Please refer to the SDK Release Notes for information on these two images.

9.1. Development of language

1) SHELL

Shell is a program written in C language, which is a bridge for users to use Linux.Shell is both a command language and a programming language.There are many types of common Linux shells, including:

Bourne Shell (/usr/bin/sh or /bin/sh)

- Bourne Again Shell (/bin/bash)
- C Shell (/usr/bin/csh)
- K Shell (/usr/bin/ksh)
- Shell for Root (/sbin/sh)

MYD-YT507H supports bourne Shell and Bourne Again Shell:

```
[root@myir:/]# echo "echo 'myir test'" > shell_demo.sh
[root@myir:/]# sh shell_demo.sh
myir test
[root@myir:/]# bash shell_demo.sh
myir test
```

2) C/C++

C/C++ is the most commonly used programming language for low-level application development under Linux platform, and the most efficient language after assembly.Development with C/C++ is usually done in a cross-development

manner, where development is done on the development host, binary executables are compiled and generated to run on the target machine, and then deployed to run on the target machine.

In this way, you need to install sdK-based software first. Please refer to the "MYD-YT507H_Linux Software Development Guide" for installation steps. After installation, you need to configure the SDK environment as follows:

First add the build tool chain to the environment variable:

```
zhaoy@myir-O-E-M:~/t507$ export PATH=$PATH://<WORKDIR>/build/toolchain/gcc-linaro-7.4.1-2019.02-x86_64_aarch64-linux-gnu/bin
zhaoy@myir-O-E-M:~/t507$ export CROSS_COMPILE=/<WORKDIR>/build/toolchain/gcc-linaro-7.4.1-2019.02-x86_64_aarch64-linux-gnu/bin/aarch64-linux-gnu-

gec@ubuntu:~/t507$aarch64-linux-gnu-gcc -v
使用内建 specs。
COLLECT_GCC=aarch64-linux-gnu-gcc
COLLECT_LTO_WRAPPER=/home/zhaoy/t507/build/toolchain/gcc-linaro-7.4.1-2019.02-x86_64_aarch64-linux-gnu/bin/../libexec/gcc/aarch64-linux-gnu/7.4.1/lto-wrapper
目标：aarch64-linux-gnu
配置为：'/home/tcwg-buildslave/workspace/tcwg-make-release_1/snapshots/gcc.git~linaro-7.4-2019.02/configure' SHELL=/bin/bash --with-mpc=/home/tcwg-buildslave/workspace/tcwg-make-release_1/_build/builds/destdir/x86_64-unknown-linux-gnu --with-mpfr=/home/tcwg-buildslave/workspace/tcwg-make-release_1/_build/builds/destdir/x86_64-unknown-linux-gnu --with-gmp=/home/tcwg-buildslave/workspace/tcwg-make-release_1/_build/builds/destdir/x86_64-unknown-linux-gnu --with-gnu-as --with-gnu-ld --disable-libmudflap --enable-lto --enable-shared --without-included-gettext --enable-nls --with-system-zlib --disable-sjlj-exceptions --enable-gnu-unique-object --enable-linker-build-id --disable-libstdcxx-pch --enable-c99 --enable-clocale=gnu --enable-libstdcxx-debug --enable-long-long --with-cloog=no --with-ppl=no --with-isl=no --disable-multilib --enable-fix-cortex-a53-835769 --enable-fix-cortex-a53-843419 --with-arch=armv8-a --enable-threads=posix --enable-multiarch --enable-libstdcxx-time=yes --enable-g
```

```
nu-indirect-function --with-build-sysroot=/home/tcwg-buildslave/workspace/tcwg-make-release_1/_build/sysroots/aarch64-linux-gnu --with-sysroot=/home/tcwg-buildslave/workspace/tcwg-make-release_1/_build/builds/destdir/x86_64-unknown-linux-gnu/aarch64-linux-gnu/libc --enable-checking=release --disable-bootstrap --enable-languages=c,c++,fortran,lto --build=x86_64-unknown-linux-gnu --host=x86_64-unknown-linux-gnu --target=aarch64-linux-gnu --prefix=/home/tcwg-buildslave/workspace/tcwg-make-release_1/_build/builds/destdir/x86_64-unknown-linux-gnu
线程模型：posix
gcc 版本 7.4.1 20181213 [linaro-7.4-2019.02 revision 56ec6f6b99cc167ff0c2f8e1a2eed33b1edc85d4] (Linaro GCC 7.4-2019.02)
```

This section demonstrates application development by writing a simple Example of Hello World. The following is a demo program hello.c written on the development host:

```
#include<stdio.h>
int main(int argc,char *argv[])
{
    printf("hello world!\n");
     return 0;
}
```

C++ prepared demo hello-cxx.cpp

```
//file: hello-CXX.cpp
#include <iostream>
using namespace std;
int main(int argc,char *argv[])
{
        cout << "hello world!";
        return 0;
 }
```

Next, compile the application. The hello.c file uses the compilation tool chain imported above aarch64-linux-gnu-gcc , C++file hello-CXX.cpp file use aarch64-linux-gnu-cpptool chain.

```
$aarch64-linux-gnu-gcc hello.c -o hello
or
$aarch64-linux-gnu-cpp hello-CXX.cpp -o hello-CXX
.....
```

Then run the scp command to copy the generated execution file to the target machine. The result is as follows:

```
[root@myir:/]# ./hello-CC
hello world!
or
[root@myir:/]# ./hello-CXX
hello world!
```

For more complex examples and development methods, please refer to the application migration section of the "MYD-YT507H_Linux Software Development Guide".

3) python

python is an interpreted, object-oriented, dynamic data typed high-level programming language.python was invented by Guido van Rossum in late 1989, with the first public release in 1991.Like Perl, python source code is governed by the GNU General Public License (GPL).This section tests the use of python from the python command line and scripts.

- **View the supported versions of Phthon**

```
[root@myir:/]# python
python python2 python2.7
```

- **python command line tests**

Start python, type the following text at the python prompt, and press Enter to see what works:

```
[root@myir:/]# python2
python 2.7.15 (Default, Jan 28 2022, 15:38:16)
20181213 [[GCC 7.4.1 linaro - 7.4-2019.02 56 ec6f6b99cc167ff0c2f8e1a2eed3 r
evision on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> print("myir test")
myir test
```

Exit the python command line and exit() to exit python:

```
>>> exit()
[root@myir:/]#
```

- **Write scripts to test python**

Write a simple python script with all python files with the.py extension:

```
[root@myir:/]# vi test.py
[root@myir:/]# cat test.py
#!/usr/bin/env python3
print("myir test")
```

To execute the script file, the interpreter executes the script in the python2 run directory in /usr/bin/env using the following command:

```
[root@myir:/]# chmod a+x test.py
[root@myir:/]# ./test.py
myir test
```

The python2 interpreter is called with script arguments to start executing the script until it finishes executing.When the script is finished, the interpreter is no longer valid.

9.2. The database

A Database is a warehouse that organizes, stores, and manages data according to data structures.There are many types of database, commonly used database Access, Oracle, Mysql, SQL Server, SQLite and so on.

1) System SQlite

SQLite is an embedded SQL database engine.Unlike most other SQL databases, SQLite does not have a separate server process.SQLite reads and writes common disk files directly.A complete SQL database with multiple tables, indexes, triggers, and views is contained in a single disk file.This lightweight database is an ACID compliant relational database management system. It is designed to be embedded, and it is already used in many embedded products. It has a very low resource footprint, in embedded devices, may need only a few hundred K of memory.The database runs faster than Mysql and PostgreSQL.

- **SQlite creates the database**

Start sqlite3 and create a new database < testdb. db>. Enter the following command on the terminal interface to enter the operation interface.

```
[root@myir:/]# sqlite3 testDB.db
SQLite Version 3.25.3 2018-11-05 20:37:38
Enter ".help" for usage hints.
sqlite>
```

The command above will create a file testdb.db in the current directory.This file will be used as a database by the SQLite engine.Notice that the sqlite3 command provides an SQLite > prompt after successfully creating the database file.

When the database is created, you can use SQLite's.databases command to check whether it is in the database list, as shown below:

```
sqlite> .databases
main: /home/root/testDB.db
sqlite>
```

Use the.quit command to exit the sqlite prompt as follows:

```
sqlite> .quit
[root@myir:/]
```

If you want to learn more about SQlite related information, please refer to th e website: https://www.sqlite.org/docs.html.

9.3. Qt application localization

This section discusses localization related setup and testing.Localization means that a program or software, on the basis of supporting internationalization, is given the language information of a specific region to adapt to the use of people in a specific region in information input and output processing.This allows some of the locale variables used by the program to be dynamically configured at program execution time.This chapter focuses on the localization of QT applications, with MYIR demonstrating mirror MEasy HMI 2.0 as an example.

1) multilingual

This section mainly uses MYIR to demonstrate mirror MEasy HMI 2.0 as an example to illustrate the practical application of multiple languages in QT projects.

Please refer to The MEasy HMI 2.0 Development Manual, Chapter 3.1 setting up the Environment for MEasy HMI compilation.

- **Open the Mxapp2 project**

MEasy HMI 2.0 (mxapp2.tar.gz), copy it to the environment and use QT Creator to open it.

- **Generating TS files**

Enter the source directory of mxapp2 project through the terminal, and execute the following command to generate the translation file.

```
qinlh@qinlh-VirtualBox:~/download/mxapp2$ lupdate mxapp2.pro
Info: creating stash file /home/qinlh/download/MXAPP/.qmake.stash
Updating 'languages/language_zh.ts'...
    Found 202 source text(s) (0 new and 202 already existing)
Updating 'languages/language_en.ts'...
Found 202 source text(s) (0 new and 202 already existing)
```

Ts and LANGUAGE_en. ts files are not generated again. Qm files generated by LANGUAGE_en. ts are used in Chinese, and QM files generated by language_en.ts are used in English.

- **Translated text**

Open languages/language_en.ts file. The source string for translation is the content inside the <source> node, and the target string for translation is the content inside the < Translation > node. Users can modify according to their requirements.

```
<message>
        <location filename="../Album.qml" line="50"/>
        < source > back < / source >
        <translation type="unfinished">Return</translation>
</message>
```

- **Generate qml files**

After modifying the Ts file, you need to manually generate the translation template file for use. Go to the languages directory and run the following command to generate the translation template file.

```
qinlh@qinlh-VirtualBox:~/download/mxapp2/languages$ lrelease language_en.ts
 -qm language_en.qm
Updating 'language_en.qm'...
    Generated 183 translation(s) (2 finished and 181 unfinished)
Ignored 21 untranslated source text(s)
qinlh@qinlh-VirtualBox:~/download/mxapp2/languages$ lrelease language_zh.ts
 -qm language_zh.qm
Updating 'language_en.qm'...
    Generated 183 translation(s) (2 finished and 181 unfinished)
    Ignored 21 untranslated source text(s)
```

- **Apply translation files**

To use the translation file, refer to the LoadLanguage member function in the translator. CPP file in the source code.

2) The font

This section mainly illustrates the practical application of fonts in the QT project with The example of MYIR demonstrating mirror MEasy HMI 2.0.Please refer to

The MEasy HMI 2.0 Development Manual, Chapter 3.1 setting up the Environment for MEasy HMI compilation.

- **Install font files**

Font files can be placed directly into the /usr/lib/fonts/ directory on the development board file system.

```
[root@myir:/]# ls /usr/lib/fonts/msyh.ttc
/usr/lib/fonts/msyh.ttc
```

Or directly add QT project inside.

```
qinlh@qinlh-VirtualBox:~/download/mxapp2/fonts$ tree
├ — — DIGITAL
│  ├ — — DIGITAL. TXT
│  └ — — DS - DIGIB. The vera.ttf
└ — — fontawesome - webfont. The vera.ttf
```

- **Using font files**

The use of the font file is called by the application code, refer to main.cpp. The iconFontInit() function in.

```
void iconFontInit()
{
    int fontId_digital = QFontDatabase::addApplicationFont(":/fonts/DIGITAL/DS-DIGIB.TTF");
    int fontId_fws = QFontDatabase::addApplicationFont(":/fonts/fontawesome-webfont.ttf");
    QString fontName_fws = QFontDatabase::applicationFontFamilies(fontId_fws).at(0);
    QFont iconFont_fws;
    iconFont_fws.setFamily(fontName_fws);
    QApplication::setFont(iconFont_fws);
    iconFont_fws.setPixelSize(20);
}
```

3) Soft keyboard

This chapter mainly illustrates the practical application of the soft keyboard in QT project with The example of Mir demonstrating mirror MEasy HMI 2.0.

Please refer to The MEasy HMI 2.0 Development Manual, Chapter 3.1 setting up the Environment for MEasy HMI compilation.

- **The soft keyboard is embedded in QML code**

Qt soft keyboard can only be called in QML code. Before calling, you need to define the position of the soft keyboard and the size of the soft keyboard, as shown in the following code.

```
InputPanel {
id: inputPanel
x: adaptive_width/8
Y: adaptive_height / 1.06
z:99
anchors.left: parent.left
anchors.right: parent.right

states: State {
name: "visible"
when: inputPanel.active
PropertyChanges {
target: inputPanel
        Y: 1.06 inputPanel adaptive_height/height
}
}
```

- **Trigger soft keyboard**

The soft keyboard is triggered by QML's TextField and TextEdit components. Users only need to add this component in the UI component to call up the soft keyboard on the UI and use it.For a QT operating system, you can use the Qt VirtualKeyboard provided with QT.

```
TextField{
id: netmask_input
InputMethodHints: Qt.ImhFormattedNumbersOnly
onAccepted: digitsField.focus = true
font.family: "Microsoft YaHei"
color: "white"
}
```

- **Use a soft keyboard**

Please refer to the SYSTEM Setting UI in Chapter 2.6 of MEasy HMI 2.0 Development Manual for the usage of the soft keyboard.A soft keyboard pops up when the user clicks on an editable interface component.

10. The resources

- **Linux Kernel open source community**

https://www.kernel.org/

- **Full vision development community**

https://www.aw-ol.com

- **Linux kernel watchdog**

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

- **Qt under embedded Linux**

https://doc.qt.io/qt-5/embedded-linux.html

- **Systemd Network configuration**

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

- **Official website of allwinner**

https://www.allwinnertech.com/

Contact us at Appendix I

Shenzhen headquarters

Responsible region: Guangdong/Sichuan/Chongqing/Hunan/Guangxi/Yunnan/Guizhou/Hainan/Hong Kong/Macao
Tel: 0755-25622735 0755-22929657
Fax: 0755-25532724
Zip code: 518020
Address: Room 04, 6 / F, Building 2, Yunli Intelligent Park, Kaifa Road, Bantian Street, Longgang District, Shenzhen

Production base

Address: 2nd Floor, Factory building C, Shengjianli Industrial Park, Dafu Industrial Zone, Guanlan Street, Longhua District, Shenzhen
Telephone: 0755-21015844

Shanghai Office

Responsible region: Shanghai/Hubei/Jiangsu/Zhejiang/Anhui/Fujian/Jiangxi
Tel: 021-60317628 15901764611
Fax: 021-60317630
Zip code: 200062
Address: Room 805, Building 1, Pudong Development Jiangcheng Plaza, 778 Jinji Road, Pudong New Area, Shanghai

Beijing Office

Responsible Area:
Beijing/Tianjin/Shaanxi/Liaoning/Shandong/Henan/Hebei/Heilongjiang/Jilin/Shanxi/Gansu/Inner Mongolia/Ningxia
Tel: 010-84675491 13269791724
Fax: 010-84675491
Zip code: 102218
Address: Room 901, Building 10, Libao Square, Yard 8, Ronghua Middle Road, Daxing District, Beijing

Sales Contact

Website: www.myir-tech.com
E-mail: sales.cn@myirtech.com

Contact technical support personnel

Telephone: 0755-25622735
E-mail: support.cn@myirtech.com

If you receive help via email, please use the following format for the subject of the email:

[Company name/person -- Development board model] Problem Overview

This will allow us to follow up on your problem more quickly so that the appropriate development team can address your problem.

Appendix II After-sales Service and Technical support

All the full range of products purchased by Mir Electronics directly or through the regular agents authorized by Mir Electronics can enjoy the following rights and interests:

1. 6 months free warranty service period
2. Lifelong free technical support service
3. Lifelong maintenance service
4. Enjoy the software upgrade service of the purchased product for free
5. Enjoy the software source code supporting the purchased products for free, as well as part of the software source code developed by MYiR Technology
6. Main chip samples can be purchased directly from Miltech, which is simple, convenient and fast;It eliminates the long waiting period when buying from agents
7. From the date of purchase, you will become a permanent customer of Miltech and enjoy preferential policies to purchase any software and hardware products of MilTech again

OEM/ODM service

Free warranty service is not provided in any of the following cases:

1. Beyond the free warranty service period
2. No product serial number or valid purchase receipt
3, into the liquid, damp, mildew or corrosion
4, by impact, extrusion, fall, scratch and other non-product quality problems caused by the fault and damage
5. Failure and damage caused by unauthorized hardware modification, incorrect power-on and incorrect operation
6. Failure and damage caused by irresistible natural factors

Product repair:
In case of product failure, damage or other abnormal phenomena in the process of use, users should call the Customer service Department of Miltech and communicate with engineers to confirm the problem before sending the product back for repair, so as to avoid unnecessary freight loss and cycle delay caused by fault judgment errors.

Maintenance cycle:
Upon receipt of the repaired products, we will arrange engineers to test them immediately, and we will repair or replace them and send them back in the shortest possible time.Generally, the maintenance period for faults is 3 working days (from the date when we receive the goods, excluding the transportation time). For products that cannot be repaired in a short time due to special faults, we will communicate with the user separately and confirm the maintenance period.

Maintenance cost:

In the free warranty period of the product, due to product quality problems caused by the fault, do not charge any maintenance fees;If the fault or damage is not within the scope of free warranty, we will communicate with the customer and confirm the maintenance fee after the detection and confirmation of the problem. We will only charge the cost of components and materials, not the maintenance service

fee.For products beyond the warranty period, the charge for component materials and maintenance service fee will be determined according to the actual damage degree.

Transportation costs:

When the product is under normal warranty, the freight sent back by the user shall be borne by the user, and the cost sent back to the user after repair shall be borne by our company.Abnormal warranty products back and forth freight are borne by the user.